



# **Universidad de las Ciencias Informáticas**

## **Facultad 2**

**Título: “Herramienta informática para la extracción masiva de metadatos en documentos científicos digitales.”**

**Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas.**

**Autores:** Arianna Páez Valdés.

Daima de la Caridad LLanes Knight.

**Tutores:** Ing. Heliodoro Rodríguez Milian.

Ing. Liusmila Nieto Cervantes.

**Co-tutor:** Ing. Carlos Rafael Labrada Arce.

Ciudad de La Habana, Cuba, 2013

“Año 54 de la Revolución”



*"La magnitud de lo que logramos no depende de lo que tenemos  
sino de lo que seamos capaces de hacer"*  
*Ernesto Rafael Guevara de la Serna*

**Declaración de Auditoría**

Declaramos ser los autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_

Daima de la Caridad Llanes Knight

\_\_\_\_\_

Arianna Páez Valdés

.

## **Arianna**

Al líder histórico de la revolución cubana Fidel Castro Ruz, principal artífice de un sueño querido la UCI. Gracias a la revolución y su desarrollo económico social por propiciar mis estudios como una verdadera profesional en ciencias informáticas en una institución tan prestigiosa de la enseñanza universitaria en Cuba. Gracias a mis profesores que durante 5 años han dado lo mejor de sí para forjarme hoy como una profesional de excelencia. A mis tutores Liusmila y Heliodoro, Al tribunal por su sabiduría y su labor educativa en este ejercicio. Al oponente por ser tan estricto exigiéndome una mayor preparación en mi formación profesional. Al profesor Jorge Amado por llegar en el momento preciso y brindarnos su apoyo y amistad. A mis amistades Sulema, Lizzy, Marvis, Anailis, Reyalex que aunque son pocas pero verdaderas y auténticas gracias por acompañarme durante estos años de lindos y gratos momentos. A mi compañera de tesis por sobrellevarme todo este tiempo.

A toda mi familia por aportar un granito de arena en mi formación profesional y espiritual. A mi mamá Erenia a mi abuela Antonia, por ser las mujeres que más amo en este mundo, por depositar toda su confianza en mí y guiarme por buen camino. Por estar siempre dispuestas a escucharme y ayudarme incondicionalmente, por ser mis cómplices y guía. A mi hermana Olivia por ser tan cariñosa y especial conmigo, por decirme siempre “te quiero tata”. A mi tía Miriam, a mi abuelo Regino que aunque casca rabia es el mejor abuelo del mundo, a mis primos Yordis, Lisandra y Alain, a mi hermanastra Yaimarys y mi padrastro Olivio.

A la familia de mi esposo por acogerme en su seno familiar y abrir su corazón para quererme como una hija más , a mi suegra por toda la ayuda que me ha brindado para crecerme profesionalmente siempre dándome luz y guía acorde a estos tiempos. A mi suegro y tío manolito por apoyarme en la decisión más importante de mi vida, gracias por estar siempre dispuestos a ayudarme en lo que sea.

A mi esposo por estar siempre a mi lado dándome apoyo, fortaleza, alegrías y espíritus necesarios en los momentos más importantes y difíciles de mi vida, por tu amor incondicional cariño y compañía. Gracias por existir y estar ahí para mí.

A mi padre que donde quiera que este se sienta orgulloso de la hija que no pudo criar ni ver crecer pues así lo quiso el destino, pero que dejaste impregnada tu fortaleza e inteligencia, gracias a tu legado soy una mujer de bien muy luchadora y revolucionaria.

A todos los que de una forma u otra han contribuido con la realización de este trabajo, pues sin todos ustedes que son mi fuente de inspiración no hubiera sido posible realizar uno de mis grandes sueños.

**Daima**

A mis padres Ileana y Reinier, por acompañarme en todo momento, por comprenderme, aconsejarme, apoyarme y por todo su amor incondicional.

A mi hermano por guiarme y ayudarme en la trayectoria de la carrera y estar a mi lado cuando más lo necesitaba.

A mi abuela Iris que para mí es como mi madre, por estar siempre a mi lado, apoyándome y ayudándome en todo lo que me haga falta.

A mi tía Iyolexis por estar pendiente a todas mi cosas, a lo que necesitaba y brindarme su apoyo desinteresado e incondicional.

A mis tíos Isora y Gregorio que me ayudaron mucho dándome consejos y compartiendo parte de la recta final de mi carrera su ayuda me hizo mucha falta gracias por existir.

A mi novio Wilmer por quererme mucho y tener esa paciencia conmigo cuando más lo necesitaba, a ti mi amor te quiero mucho.

A mi tía Martha María que no se encuentra hoy aquí pero sé que su sueño era verme graduada y preparada para la vida, a ti tía no te olvidare nunca.

A todos mis tíos, por su constante preocupación, apoyo y darme ánimo en todo momento.

A mis abuelos fallecidos Cini y Martha que con su amor y cariño me educaron y formaron esta familia que tanto quiero nunca los olvidaré.

A mis primos por todo su amor y cariño.

A toda mi familia por apoyarme cuando lo he necesitado.

A mi compañera de tesis por todo el trabajo realizado.

A mis tutores Heliodoro y Liusmila por brindarnos todo su apoyo en la realización de este trabajo.

A todos mis profesores especialmente a Jorge Amado Soria Ramírez y Luis Carlos Rodríguez que me transmitieron conocimientos y valores que me permitieron formarme como profesional.

A todos mis compañeros de aula por estar siempre ahí cuando más los necesitaba.

A todos en general por desearme lo mejor.

**Dedicatoria:**

**Arianna**

A mi abuela Antonia por ser mi todo confidente y amiga, a mi mamá por exigirme tanto que me preparase en la vida, por su dedicación y entrega.  
A mi esposo por ayudarme en todo momento y estar siempre para mí cumpliendo mis caprichos. Gracias a todos querermelo como soy.

**Daima**

Dedico esta investigación a mis abuelos Iris y Cini por haberme dado todo lo que soy como persona, por su apoyo y dedicación, los quiero mucho.  
A ti abuelita linda Iris, tú has hecho en esta vida mucho por toda la familia a ti te debo este trabajo por tu apoyo, educación y sacrificio ante todo, como tu ninguna por eso te quiero mucho.  
A mis padres por haberme dado la vida, por hacer de mí la persona que hoy soy.  
A mi tía Iyolexis porque sin su apoyo incondicional todas las cosas no se hubieran logrado, nunca te voy a olvidar.

**Resumen**

La gestión de la información científico-técnica en las instituciones académicas es uno de los desafíos que enfrenta la sociedad actual. La Universidad de las Ciencias Informáticas cuenta con un Sistema Integrado de Gestión Bibliotecaria, en el cual se manejan de forma automatizada los procesos que se desarrollan en la misma adquisición, catalogación, circulación y control de los usuarios. La institución cuenta además con un Repositorio Institucional para la conservación de la información con que cuenta.

El presente trabajo de investigación tiene como objetivo desarrollar una herramienta informática para la extracción y exportación automática de metadatos en documentos científicos digitales a los formatos bibliográficos Marc 21 y Dublin Core. Esta exportación se realiza en un XML<sup>1</sup> utilizando los campos de datos necesarios de los formatos antes mencionados, para ser insertado posteriormente en el Sistema Integrado de Gestión Bibliotecaria y en el Repositorio Institucional DSpace.

**Palabras Claves:**

metadatos, extracción, sistema integrado de gestión bibliotecaria, repositorio institucional, marc21, dublin core.

---

<sup>1</sup> Extensible Markup Language (Lenguaje de Marcas Extensible)

**Índice de Contenidos**

<b>Introducción</b> -----	<b>1</b>
<b>Capítulo 1: Fundamentación Teórica</b> -----	<b>5</b>
<b>1.1 Introducción</b> -----	<b>5</b>
<b>1.2 ¿Qué es extracción de metadatos en documentos digitales?</b> -----	<b>5</b>
<b>1.3 Estándares para representar Metadatos</b> -----	<b>6</b>
<b>1.3.1 MARC 21</b> -----	<b>6</b>
<b>1.3.2 Dublin Core</b> -----	<b>7</b>
<b>1.4 Exportar Metadatos</b> -----	<b>8</b>
<b>1.5 Análisis de herramientas de extracción de metadatos</b> -----	<b>8</b>
<b>1.5.1 Metadata Extraction Tool</b> -----	<b>9</b>
<b>1.5.2 Foca (Fingerprinting Organizations with Collected Archives)</b> -----	<b>10</b>
<b>1.5.3 Apache Tika</b> -----	<b>10</b>
<b>1.5.4 Google Desktop</b> -----	<b>11</b>
<b>1.5.5 Windows Search</b> -----	<b>12</b>
<b>1.5.6 Finder</b> -----	<b>12</b>
<b>1.5.7 Selección de la herramienta a utilizar</b> -----	<b>13</b>
<b>1.6 Metodologías de Desarrollo de Software.</b> -----	<b>13</b>
<b>1.6.1 Proceso Unificado de Desarrollo de Software</b> -----	<b>13</b>
<b>1.7 Lenguajes usados para el desarrollo del sistema.</b> -----	<b>14</b>
<b>1.7.1 Lenguaje Unificado de Modelado</b> -----	<b>14</b>
<b>1.7.2 Lenguaje de Programación</b> -----	<b>14</b>
<b>1.7.2.1 Java</b> -----	<b>15</b>
<b>1.7.3 Documento de Modelado de Objetos en Java</b> -----	<b>15</b>
<b>1.8 Gestor de Base Datos SQLite3</b> -----	<b>16</b>
<b>1.9 Herramienta CASE</b> -----	<b>17</b>
<b>1.9.1 Visual Paradigm</b> -----	<b>17</b>
<b>1.10 IDE Desarrollo</b> -----	<b>17</b>
<b>1.10.1 NetBeans</b> -----	<b>18</b>

1.10.2 Eclipse .....	18
1.10.3 Selección del IDE Desarrollo a utilizar .....	18
1.11 Conclusiones .....	19
Capítulo 2: Diseño .....	20
2.1 Introducción .....	20
2.2 Propuesta del Sistema .....	20
2.3 Modelo de Dominio .....	20
2.4 Requisitos del Sistema .....	22
2.4.1 Requerimientos funcionales .....	22
2.4.1.1 Validación de los requisitos funcionales .....	23
2.4.2 Requerimientos no funcionales .....	24
2.5 Modelo de casos de uso del sistema .....	24
2.5.1 Descripción del actor .....	24
2.5.2 Definición de los casos de uso del sistema .....	25
2.5.2.1 Especificación de Casos de Uso .....	26
2.6 Descripción de la Arquitectura .....	32
2.7 Patrones de Diseño .....	33
2.8 Modelos de Diseño .....	36
2.8.1 Diagrama de Clases del Diseño .....	36
2.8.2 Diagrama de Secuencia .....	38
2.9 Modelo Entidad-Relación .....	42
2.10 Conclusiones .....	43
Capítulo 3: Implementación y Prueba .....	44
3.1 Introducción .....	44
3.2 Diagrama de Componentes .....	44
3.3 Diagrama de Despliegue .....	45
3.4 Prueba .....	45
3.4.1 Pruebas Unitarias .....	45
3.4.2 Prueba de Aceptación .....	47
3.4.3 Diseño de Casos de Prueba .....	48

<b>3.4.4 Resultado de Pruebas</b> -----	<b>51</b>
<b>3.5 Conclusiones</b> -----	<b>52</b>
<b>Conclusiones Generales</b> -----	<b>53</b>
<b>Recomendaciones</b> -----	<b>54</b>
<b>Referencia Bibliográfica</b> -----	<b>55</b>
<b>Bibliografía</b> -----	<b>56</b>
<b>Anexos</b> -----	¡Error! Marcador no definido.
<b>Glosario de Términos</b> -----	<b>59</b>

**Índice de Figuras**

Figura 2. 1 Proceso de Extracción de Metadatos .....21

Figura 2. 2 Diagrama de Casos de Uso .....26

Figura 2. 3 Representación de la arquitectura.....33

Figura 2. 4 Diagrama de Clases del Diseño .....37

Figura 2. 5 Diagrama de Secuencia del Caso de Uso Extraer Metadatos .....39

Figura 2. 6 Diagrama de Secuencia del Caso de Uso Extendido Mostrar Metadatos .....40

Figura 2. 7 Diagrama de Secuencia Sección 1 “Seleccionar todos los ficheros “ .....41

Figura 2. 8 Diagrama de Secuencia Sección 2 “Visualizar metadatos” .....41

Figura 2. 9 Diagrama de Secuencia Sección 3 “Eliminar Ficheros” .....42

Figura 2. 10 Modelo de Datos .....43

Figura 3. 1 Diagrama de Componente .....44

Figura 3. 2 Diagrama de Despliegue.....45

Figura 3. 3 Pruebas Unitarias .....47

## **Índice de Tablas**

Tabla 2. 1 Descripción de los actores del sistema.....	25
Tabla 3. 1 Diseño de Casos de Prueba CU Gestionar Listado de Documentos .....	49
Tabla 3. 2 Descripción de Variables CU Gestionar Listado de Documentos .....	49
Tabla 3. 3 Diseño de Casos de Prueba CU Crear Usuario.....	50
Tabla 3. 4 Descripción de Variables CU Crear Usuario.....	50
Tabla A1. 1 Campos de datos agrupados en bloques por el primer carácter de la etiqueta. ....	<b>¡Error!</b>
<b>Marcador no definido.</b>	
Tabla A1. 2 Campos de datos agrupados en bloques por el primer carácter de la etiqueta (Continuación). .....	<b>¡Error! Marcador no definido.</b>
Tabla A1. 3 Apartados de contenido, propiedad intelectual.....	<b>¡Error! Marcador no definido.</b>
Tabla A1. 4 Apartados de aplicación.....	<b>¡Error! Marcador no definido.</b>
Tabla A1. 5 Descripción del Caso de Uso Autenticar Usuario .....	<b>¡Error! Marcador no definido.</b>
Tabla A1. 6 Descripción del Caso de Uso Modificar Contraseña .....	<b>¡Error! Marcador no definido.</b>
Tabla A1. 7 Descripción del Caso de Uso Buscar Documentos .....	<b>¡Error! Marcador no definido.</b>
Tabla A1. 8 Descripción del Gestionar Listado de Documentos .....	<b>¡Error! Marcador no definido.</b>
Tabla A1. 9 Descripción del Caso de Uso Generar MarcXML .....	<b>¡Error! Marcador no definido.</b>
Tabla A1. 10 Descripción del Caso de Uso Generar Dublin Core .....	<b>¡Error! Marcador no definido.</b>
Tabla A1. 11 Descripción del Caso de Uso Crear Usuario .....	<b>¡Error! Marcador no definido.</b>
Tabla A1. 12 Descripción del Caso de Uso Gestionar Listado de Usuario .....	<b>¡Error! Marcador no definido.</b>

## Introducción

Las instituciones académicas, a lo largo de su historia han generado y siguen generando, gran cantidad de documentos, revistas, proyectos, informes de investigaciones, tesis, libros, conferencias y clases, anteriormente difundir esta información resultaba difícil. Las nuevas Tecnologías de la Información y las Comunicaciones (TIC) ofrecen la oportunidad de crear bibliotecas digitales y repositorios institucionales para la organización, preservación, centralización y difusión de la información en las instituciones.

Para asegurar el acceso a dicha información la misma puede gestionarse a través de un software que permita ubicarla de manera coherente dentro de la Web, facilitando su recuperación a través de herramientas de búsqueda por medio de metadatos asociados a cada objeto digital.

*“Un metadato describe los atributos de un recurso, teniendo en cuenta que el recurso puede consistir en un objeto bibliográfico, registros e inventarios archivísticos, objetos geoespaciales, recursos visuales y de museos o implementaciones de software. Aunque puedan presentar diferentes niveles de especificidad o estructura, el objetivo principal es el mismo: describir, identificar y definir un recurso para recuperar, filtrar, informar sobre condiciones de uso, autenticación y evaluación, preservación e interoperatividad.”* (José A. Senso)(1)

En la Universidad de las Ciencias Informáticas (UCI), parte de la documentación científica que se gestiona en formato digital está en manos del Centro de Información Científico Técnico (CICT).

El CICT cuenta con un Sistema Integrado de Gestión Bibliotecaria (SIGB), donde son automatizados los procesos que se desarrollan en la misma: adquisición, catalogación, circulación, control de los usuarios y cuenta con un catálogo en línea (OPAC) por sus siglas en Inglés (On line Public Access Catalog) en el cual se pueden realizar búsquedas sobre el material bibliográfico con que cuenta la institución, tanto en formato duro como digital.

Una vez que los documentos digitales son adquiridos por el Departamento de Proceso Técnico del CICT de la universidad se les realiza el procesamiento analítico sintético de la documentación (PAC), este se refiere a la descripción física y del contenido, lo cual crea una relación dicotómica<sup>2</sup> entre la forma y contenido. Los metadatos del documento se extraen de forma manual por el personal de

---

<sup>2</sup> Pertenece o relativo a la división entre dos partes.

dicho departamento y se almacenan en el formato bibliográfico Marc 21<sup>3</sup>, y pasan a estar disponibles en el catálogo en línea (OPAC).

Debido al cúmulo de documentos científicos en formato digital obtenidos a raíz de las publicaciones realizadas por los estudiantes y profesores de la universidad, producto a los trabajos de diplomas, tesis de maestría y doctorado, libros, artículos y participación en eventos, se decidió implantar un repositorio institucional para la Universidad de las Ciencias Informáticas, con el objetivo de almacenar la información científico-técnica generados por los investigadores(estudiantes y profesores) de la misma.

*“Un Repositorio Institucional es un conjunto de servicios de almacenamiento, gestión y diseminación<sup>4</sup> de materiales digitales disponibles a los miembros de una determinada comunidad académica.”* (Luis Carlos Álvarez Fernández, 2012)(2). Es un sistema que ofrece un conjunto de servicios soportados en red, para la sociedad científica de la institución, que permite a los usuarios el acceso a trabajos realizados y a la organización, el almacenamiento, preservación y visibilidad de la producción científica en formato digital de sus investigadores.

El repositorio institucional y el SIGB se enriquecen continuamente de nuevos materiales. Actualmente ambos permiten incorporar y catalogar nuevos documentos, pero este proceso es realizado hoy de forma manual, lo que provoca un gran atraso en la publicación de dichos documentos y genera gran cantidad de trabajo al personal encargado de la catalogación.

Por lo planteado anteriormente surge el siguiente **problema a resolver**: ¿Cómo contribuir a la catalogación masiva de documentos científicos digitales en la Universidad de las Ciencias Informáticas?

Del problema expuesto se identifica como **objeto de estudio**: Proceso de extracción de metadatos en documentos digitales.

El **campo de acción** se enmarca en la automatización del proceso de extracción de metadatos en documentos científicos digitales de la Universidad de las Ciencias Informáticas.

---

<sup>3</sup> MARC21,” Format for Bibliographic Data.” “Machine Readable Cataloging (MARC)” Escrito por Betty Furrie en conjunto con el Departamento de Desarrollo de Bases de Datos de la Follet Software Company Séptima edición revisada y editada por la Oficina de Desarrollo de Redes y Normas MARC, Biblioteca del Congreso, 2003 con permiso de la Biblioteca del Congreso.

<sup>4</sup> Acción y efecto de esparcir.

Se define como **objetivo general**: Desarrollar una herramienta informática para la extracción y exportación automática de metadatos en documentos científicos digitales a los formatos bibliográficos Marc 21 y Dublin Core.

Del objetivo general se derivan los siguientes **objetivos específicos**:

1. Caracterizar el marco teórico-conceptual de los métodos y herramientas para la extracción masiva de metadatos en documentos digitales.
2. Investigar los estándares de metadatos para la descripción bibliográfica Marc 21 y Dublin Core.
3. Implementar una herramienta de extracción masiva de metadatos en documentos científicos digitales.

Para dar cumplimiento a los objetivos específicos de esta investigación se definen las siguientes **tareas**:

1. Estudio de las herramientas informáticas existentes en el mundo para extracción automática de metadatos en documentos digitales.
2. Selección de una librería para la extracción automática de metadatos en documentos digitales.
3. Estudio sobre la exportación de metadatos en el formato XML con los campos de datos de Marc 21 y Dublin Core.
4. Estudio y selección de la metodología para el desarrollo de la herramienta informática.
5. Análisis, diseño e implementación de la herramienta para la extracción automática de metadatos.
6. Realización de pruebas funcionales a la solución.

**Los métodos utilizados en la investigación:**

**Métodos Teóricos:**

**Analítico-Sintético:** Con el objetivo de analizar documentos, trabajos de diplomas, sitios web donde se expongan información referente a la extracción de metadatos en documentos digitales. Permitiendo la recopilación de los elementos más importantes para el desarrollo del presente trabajo de investigación.

## **Métodos Empíricos:**

**Observación:** Con el desarrollo de la herramienta de extracción masiva de metadatos en documentos digitales y la exportación de estos en los formatos bibliográficos Marc 21 y Dublin Core, se evalúa los resultados obtenidos contra los esperados.

Se plantea como **resultado:**

Herramienta para la extracción masiva de metadatos en documentos científicos de la Universidad de las Ciencias Informática.

## **Estructura Capítular:**

**Capítulo 1: Fundamentación teórica:** Se realiza un estudio de las principales herramientas de extracción de metadatos en documentos digitales y las diferentes clasificaciones de metadatos que existen. Se describen los lenguajes, tecnologías y metodología a utilizar para el desarrollo del sistema.

**Capítulo 2: Diseño:** Se realiza la propuesta del sistema a desarrollar, describiendo los elementos relacionados con el diseño de la solución planteada; creándose los artefactos, diagrama de modelo de dominio, el diagrama de caso de uso del sistema y sus descripciones. Se define la arquitectura y los patrones de diseño que se van a utilizar, conformándose finalmente el modelo de diseño, el cual constituye el punto de partida para la implementación del sistema.

**Capítulo 3: Implementación y Prueba:** Se implementan todos los elementos descritos en el diseño para lograr el objetivo general, realizándose las pruebas que permitan validar la funcionalidad del sistema.

## Capítulo 1: Fundamentación Teórica

### 1.1 Introducción

En el presente capítulo se realiza un estudio del estado del arte de las distintas herramientas de extracción de metadatos existentes en el mundo. Se definen los conceptos básicos asociados al dominio del problema, se estudian las tecnologías, metodologías y herramientas usadas para el desarrollo del sistema, seleccionando así la herramienta de extracción de metadatos que más se adapta a la solución del problema de la investigación científica.

### 1.2 ¿Qué es extracción de metadatos en documentos digitales?

Las bibliotecas que contengan documentos digitales requieren de la gestión de metadatos sobre dichos objetos.

Los metadatos pueden ser clasificados en tres tipos:

**Metadatos Descriptivos:** Se utilizan para la descripción e identificación de la información contenida en el recurso. Contienen atributos físicos (medios, condición de las dimensiones) y atributos bibliográficos (título, autor/ creador, idioma, palabras claves). (Patricia Testa, 2008)

**Metadatos Administrativos:** Se refieren a las características y propiedades del recurso, facilitando la gestión y procesamiento tecnológico y físico de las colecciones digitales tanto a corto como a largo plazo. Incluyen información sobre la creación y el control de calidad, gestión de derechos, control de acceso y utilización y condiciones de preservación. (Patricia Testa, 2008)

**Metadatos Estructurales:** Proporcionan información sobre la estructura interna de los recursos electrónicos, como página, sección, capítulo, partes, índices, tabla de contenido, etc. y describen la relación entre los materiales. Facilitan la navegación y presentación de los recursos y relacionan las diferentes partes que lo componen. Ver la descripción en el *Anexo #1*. (Patricia Testa, 2008)

La extracción automática de metadatos consiste en obtener un conjunto de atributos o elementos necesarios que describan documentos digitales. Los metadatos extraídos se utilizan para la descripción e identificación de los materiales digitales, una vez que este proceso se realice estos elementos pueden ser depositados en una base de datos, repositorio o en un lugar donde se almacena información digital con el objetivo preservar dicha información.

### 1.3 Estándares para representar Metadatos

Un estándar es algo “que sirve como tipo, modelo, norma, patrón o referencia.” (Real Academia Española, 2001)(3).

Existen diferentes estándares para representar metadatos como son los de archivos geográficos, los de transferencia de datos y los estándares de metadatos digitales geoespaciales y educativos, entre otros; los mismos poseen características particulares. Para resolver el problema expuesto se realizó un estudio de los estándares más reconocido para la descripción bibliográfica Marc 21 y Dublin Core, estos son utilizados por la biblioteca de la universidad y el repositorio institucional, los mismos son explicados a continuación:

#### 1.3.1 MARC 21

*“El formato Marc 21 para registros bibliográficos ha sido diseñado para servir como portador de la información bibliográfica relativa a: materiales textuales impresos y manuscritos, archivos de computador, mapas, música, recursos continuos, materiales visuales y materiales mixtos. La información bibliográfica comúnmente incluye: títulos, nombres, tópicos, notas, datos de publicación e información sobre la descripción física de un ítem. El formato bibliográfico contiene elementos de información para los siguientes tipos de materiales:”* (Congreso, 2008) (4)

##### ❖ Libros

Se utiliza para materiales textuales impresos, electrónicos, manuscritos y en microformatos, cuya naturaleza es monográfica.

##### ❖ Recursos Continuos

Se utiliza para materiales textuales impresos, electrónicos y en microformatos, que se emiten en partes con un patrón de publicación recurrente, por ejemplo publicaciones periódicas, diarios, anuarios.

##### ❖ Mapas

Se utiliza para todo tipo de materiales cartográficos impresos, manuscritos, electrónicos y en microformatos, incluyendo atlas, mapas planos y globos. Los materiales pueden ser de naturaleza monográfica o seriada.

### ❖ Música

Se utiliza para música impresa, electrónica y en microformato; así como para grabaciones sonoras musicales y grabaciones sonoras no musicales. Los materiales pueden ser de naturaleza monográfica o seriada.

*“Los campos variables de datos se agrupan en bloques de acuerdo con el primer carácter de la etiqueta, la cual es identificadora con algunas excepciones, la función de los datos dentro del registro. El tipo de información del campo se identifica por el resto de la etiqueta.”* (Congreso, 2008)(4)

Ver todos los campos de datos en el *Anexo #2*.

Los campos de datos que se utilizaron de Marc 21 para el presente trabajo de investigación son centro, facultad, autor, tutor, título, resumen, palabras claves estos son los que serán generados por la herramienta implementada.

### 1.3.2 Dublin Core

Dublin Core es un modelo de metadatos elaborado 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. (Universidad, 2010)

Las implementaciones de Dublin Core usan generalmente XML, se define por la norma ISO 15836 del año 2003, y la norma NISO Z39.85-2007. (Universidad, 2010)

*“El modelo Dublin Core es, juntamente con el formato MARC, el formato de metadatos más ampliamente utilizado en el desarrollo de repositorios de objetos digitales.”* (REBIUM, 1994)(5)

El sistema Dublin Core está constituido por quince metadatos básicos, los cuales se dividen en tres apartados: contenido, propiedad intelectual y aplicación. (Universidad, 2010). Ver los metadatos básicos de cada apartado en el *Anexo #3*.

Los campos de datos que se utilizaron de Dublin Core para el presente trabajo de investigación son centro, facultad, autor, tutor, título, resumen, palabras claves estos son los que serán generados por la herramienta implementada.

### **1.4 Exportar Metadatos**

Para el desarrollo de la herramienta la misma debe extraer de forma masiva los metadatos de los documentos científicos digitales dígase, trabajo de diploma, tesis de maestría y doctorado de la universidad y después exportar los metadatos en un XML, para que estos sean importados al SIGB, y al Repositorio Institucional DSpace.

DSpace es un software de código abierto que provee herramientas para la administración de colecciones digitales, y comúnmente es usada como solución de repositorio institucional. Es utilizado actualmente en el repositorio institucional de la universidad. DSpace está instalado y configurado para utilizar por defecto el formato de metadatos Dublin Core. (Luis Carlos, 2010)

Para importar un conjunto de metadatos al Repositorio Institucional DSpace es necesario un directorio, y dentro de este varios sub-directorios (uno por cada fichero en formato .pdf a subir). Dentro de cada subdirectorio va incluido el fichero a subir, un archivo dublin\_core.xml, que contiene los metadatos del fichero, y un fichero de texto contents que contiene la dirección del fichero y una breve descripción del fichero. (Luis Carlos, 2010)

El Centro de Información Científico Técnico, cuenta con un SIGB el cual contiene los campos de datos del registro bibliográfico Marc 21. Una vez que la herramienta exporte los metadatos en el formato XML, este contendrá los campos de datos necesarios del formato bibliográfico Marc 21 para ser registrados en el SIGB, lo mismo sucede para repositorio institucional, después que se exporten los metadatos con la estructura explicada anteriormente podrá ser insertada en dicho sistema.

### **1.5 Análisis de herramientas de extracción de metadatos**

Para facilitar el acceso, almacenamiento, clasificación y búsqueda de los recursos es necesario describirlos según sus características o propiedades como son: autor/es, título, año publicación, fecha, resumen entre otros.

Existen varias librerías de indexación y búsqueda, que se pueden embeber en aplicaciones informáticas como son Egothor, Xapian, entre otras. Una de las más potentes es Lucene que es una librería para la recuperación de información apoyado por el Apache Software Foundation y se distribuye bajo la Apache Software License. Es útil para cualquier aplicación que requiera indexado y búsqueda a texto completo. Es desarrollada en java pero tiene versiones para otros lenguajes como Delphi, C#, C++, Ruby, Perl, Python y PHP. Lucene puede indexar y realizar búsquedas sobre cualquier dato que pueda ser convertido a texto plano, por lo que el archivo de entrada debe ser un .txt. (Erik Hatcher, 2005)

Con el objetivo de obtener descripciones de los recursos, se analizaron a continuación diferentes herramientas de extracción de metadatos:

### **1.5.1 Metadata Extraction Tool**

Desarrollada por la Biblioteca Nacional de Nueva Zelanda inicialmente en 2003 y lanzado como software de código abierto en 2007, la versión actual se puede descargar desde la página de descarga de SourceForge o directamente desde la página de la Biblioteca Nacional de Nueva Zelanda. (Zealand, 2003)

Los objetivos del software consisten en (Zealand, 2003):

- ❖ Extraer automáticamente metadatos para la preservación de archivos digitales.
- ❖ Dar salida a los metadatos en un formato estándar XML o (Lenguaje de Marcas Extensible) para su uso en actividades de conservación y de recuperación de información.

La herramienta extraer metadatos en los siguientes formatos:

- ❖ Imágenes: BMP, GIF, JPEG y TIFF.
- ❖ Documentos de Office: Microsoft Word (versión 2, 6), Word Perfect, Open Office (versión 1), MS Works, MS Excel, MS PowerPoint y PDF.
- ❖ Audio y vídeo: WAV, MP3 (normal y con ID3Tags), BFW, FLAC.
- ❖ Lenguajes de marcado: HTML y XML.

Requerimiento de hardware y software:

**Hardware:** A nivel de procesador, memoria RAM, espacio en disco duro no hay limitaciones conocidas; no requiere un hardware con condiciones específicas de capacidad.

**Software:** La herramienta tiene tanto una interfaz de Microsoft Windows como una interfaz de línea de comandos de UNIX que permite trabajar para ser automatizado a través del procesamiento por lotes o procesado de forma individual según las necesidades. La herramienta de extracción de metadatos utiliza una combinación de Java y XML.

**Licencia:**

Se distribuye como software libre bajo la Licencia Pública Apache (versión 2.0). (Zealand, 2003)

### 1.5.2 Foca (Fingerprinting Organizations with Collected Archives)

Es una herramienta para la recolección de archivos publicados en los sitios web; extrae los metadatos y los analiza. Permite trabajar con una base de datos SQL server. Ayuda en las labores de Footprinting<sup>5</sup> y Fingerprinting<sup>6</sup> a los auditores de seguridad, además elimina y modifica los metadatos. (Informática 64)

La herramienta extraer metadatos e información oculta en los siguientes formatos:

- ❖ Microsoft Office de la versión 97 a la 2007.
- ❖ Open Office.
- ❖ Portable Document Format (pdf).
- ❖ Corel Word Perfect document (wpd).
- ❖ Joint Photographic Experts Group (jpg).

### 1.5.3 Apache Tika

Apache Tika es un conjunto de herramientas para detectar y extraer metadatos y texto estructurado del contenido de varios tipos de documentos usando librerías de parseo existentes. Tika es un proyecto de la Apache Software Foundation escrito en Java. Tika está escrito en Java, que es donde se obtiene la mayor parte de su flexibilidad y expresividad. (Maven, 2012)

---

<sup>5</sup> obtención de la huella

<sup>6</sup> obtención de la huella genética

## Licencia:

Se distribuye bajo la Licencia Pública Apache (versión 2.0) (License, 2004).

La herramienta puede extraer metadatos en los siguientes formatos:

- ❖ HTML.
- ❖ XML y derivados.
- ❖ doc, .xls, .ppt (formatos de documentos de Microsoft Office).
- ❖ .odt (Formato OpenDocument de OpenOffice).
- ❖ .pdf (Portable Document Format).
- ❖ .epub (formato para libros electrónicos).
- ❖ .rtf (Rich Text Format).
- ❖ formatos comprimidos (ar, cpio, tar, zip, gzip, bzip2 y zip).
- ❖ .txt (detectando el juego de caracteres).
- ❖ Audio (.mp3, .mid, .wav).
- ❖ Imagen (.jpeg).
- ❖ Video (.flv).
- ❖ ficheros Java.

Las empresas más importantes en el mundo del software que poseen sus propias aplicaciones:

### 1.5.4 Google Desktop

Aplicación de búsqueda de escritorio que permite encontrar texto en mensajes de correo electrónico, archivos del equipo, chats y páginas web que se han visitado. Tras su descarga e instalación, la aplicación crea un índice de todo el contenido que puede buscarse y lo almacena en el ordenador para que encuentre la información con la misma facilidad con que realizarías una búsqueda por Internet mediante Google. A diferencia del software de búsqueda tradicional que se actualiza una vez al día, esta aplicación actualiza continuamente la mayor parte de archivos. De este modo, por

ejemplo, se puede buscar un nuevo mensaje en la aplicación de correo a los pocos segundos de haberlo recibido. Google Desktop también presenta nuevas formas de acceder a la información organiza los resultados de las búsquedas de mensajes en conversaciones de modo que todos los mensajes del mismo subproceso se agrupen en un solo resultado, y guarda copias "en caché" de toda la información para que pueda visualizar antiguas versiones de documentos y páginas web aunque no esté conectado a Internet. (Compendio)

### **1.5.5 Windows Search**

Windows Search (anteriormente conocido como Windows Desktop Search o WDS en Windows XP y Windows Server 2003) es una plataforma de búsquedas indexadas de escritorio publicado por Microsoft para el sistema operativo Windows. Windows Search para Windows Vista y Windows Server 2008 (también mencionada como Instant Search) es un sucesor de Windows Indexing Service.

Emplea una arquitectura diferente y un nuevo indizador en comparación con el servicio de indización de Windows XP. Windows Search colectivamente se refiere a ambos: búsqueda indexada en Windows Vista y WDS en Windows XP. No sólo comparten una arquitectura común y tecnología de indización, sino que son compatibles entre sí. Windows Desktop Search crea y mantiene un índice de escritorio de su equipo personal. En el índice se incluye la metadata disponible sobre cada archivo. Windows Desktop Search actualiza el índice cada vez que añade o cambia un archivo o un mensaje. (Corporation, 2013)

### **1.5.6 Finder**

Finder es la aplicación ejecutada en el sistema operativo Mac OS X responsable de la gestión total de los archivos de usuario, discos, red y el lanzamiento de otras aplicaciones. Es el primer programa con el que un usuario interactúa después de arrancar un Mac, y por ello es el responsable de la apariencia general de la máquina. Se debe tener cuidado de distinguir esto de la interfaz gráfica real del equipo, que es proporcionada por servicios particulares dentro del Mac OS.

Mantiene una vista del sistema de archivos que es representada usando la apariencia del escritorio, es decir los archivos y los directorios son representados como sus iconos apropiados, los volúmenes se muestran en el escritorio, y hay una papelerera a la cual los archivos pueden ser arrastrados para marcarlos como borrados. Además Finder presenta características de búsqueda muy avanzadas

como: búsqueda por criterios que se pueden guardar, búsqueda inmediata o el uso de etiquetas. (Finder, 2010)

### 1.5.7 Selección de la herramienta a utilizar

Se llega a la conclusión de que existen disímiles herramientas que son muy potentes en cuanto a la indexación y búsqueda de metadatos en diferentes formatos. Se utilizó como base para el desarrollo la librería Apache Tika por sus características.

### 1.6 Metodologías de Desarrollo de Software.

Las metodologías de desarrollo de software abarcan todo el ciclo de vida del software, y se definen como *“conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software.”* (Patón, 2006-2007)(6)

A continuación se explica la metodología que facilita la respuesta al problema científico:

#### 1.6.1 Proceso Unificado de Desarrollo de Software

El Proceso Unificado de Desarrollo de Software (RUP) es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.

El proceso unificado está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectado a través de interfaces. El proceso unificado utiliza el lenguaje unificado de modelado (UML) para preparar todos los esquemas de un sistema software. (Jacobson, y otros, 2000)

Ventajas que ofrece trabajar con RUP (KRUP, 2000):

- ❖ Permite crear software que cumplan las expectativas de los usuarios a través de las especificaciones de requisitos.
- ❖ Permite llevar un seguimiento detallado en cada una de las fases de desarrollo.

Se escoge RUP como la metodología de desarrollo software puesto que es una metodología completa y bien documentada. Además el sistema que se desarrollará formará parte del proyecto SIGB de la Facultad 2, por lo que se debe ajustar a la metodología de desarrollo que utiliza el mismo y los

artefactos generados deben corresponderse con los definidos en las plantillas de su Expediente de Proyecto.

## 1.7 Lenguajes usados para el desarrollo del sistema.

### 1.7.1 Lenguaje Unificado de Modelado

El lenguaje de modelado es la notación que utilizan los métodos para expresar un diseño. El proceso indica los pasos que se deben seguir para llegar a un diseño. El más conocido internacionalmente es el Lenguaje de Modelado Unificado (UML) que es el que se utilizará para el modelado del sistema en su versión UML v1.x, este permite modelar, construir y documentar los elementos que forman un producto de software que responde a un enfoque orientado a objetos<sup>7</sup>.

Será utilizado por las siguientes ventajas:

- ❖ Posibilita la descripción de sistemas, simplificando la complejidad de estos y sin pérdida de información, haciendo posible la comprensión del sistema tanto para usuarios como desarrolladores.
- ❖ Brinda facilidades para el diseño, documentación, reutilización de código y detecciones de fallas.
- ❖ Facilita la comunicación entre desarrolladores, permite ahorrar tiempo en el desarrollo del software y hace más sencillas las modificaciones que se vayan a realizar.

### 1.7.2 Lenguaje de Programación

*“Un lenguaje de programación es un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. Los lenguajes de programación son idiomas artificiales diseñados para expresar cálculos y procesos que serán llevados a cabo por ordenadores. Dicho lenguaje está formado por un conjunto de palabras reservadas, símbolos, reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones.” (Guevara)(7)*

---

<sup>7</sup> BOOCH Grady et al. *El lenguaje Unificado de Modelado*, Primera Edición, Editorial Addison Wesley, 1999.

En la actualidad existen disímiles lenguajes de programación cada uno de ellos con características que lo distinguen, a continuación se expone el que se utilizó:

### 1.7.2.1 Java

El lenguaje de programación Java, fue diseñado por la compañía Sun Microsystems, que describen el lenguaje de java como “*simple, orientado a objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutra, portable, de altas prestaciones, multitarea y dinámico*”. (Jalón, y otros, 2000)(8)

A continuación se explican algunas características: (Aguilar, y otros, 2001)

- ❖ **Sencillo:** Los lenguajes de programación orientados a objetos no son tan sencillos ni fáciles de utilizar, pero Java es un poco más fácil que el popular C++, lenguaje de desarrollo de software más popular hasta la implementación de Java. Java elimina los punteros de C++ y reemplaza la herencia múltiple de C++ en una estructura única denominada interfaz (interface<sup>8</sup>).
- ❖ **Orientado a Objetos:** Se denomina debido a que la programación en Java se centra en la creación, manipulación y construcción de objetos. Un objeto tiene propiedades (un estado) y un comportamiento.
- ❖ **Interpretado:** Es interpretado y necesita un intérprete para ejecutar programas de Java. Los programas se compilan en la Máquina Virtual de Java (JVM, Java Virtual Machine) generándose un código interpretado denominado bytecode<sup>9</sup>. El bytecode es independiente de la máquina y se puede ejecutar en cualquier máquina que tenga un intérprete de Java.
- ❖ **Multihilo:** Es la capacidad de un programa ejecutar varias tareas simultáneamente.

Se seleccionó el lenguaje java porque la librería utilizada como base para el desarrollo esta implementada en java.

### 1.7.3 Documento de Modelado de Objetos en Java

---

<sup>8</sup> Una interface es un conjunto de declaraciones de funciones.

<sup>9</sup> Es un conjunto de instrucciones altamente optimizadas que son interpretados por el intérprete de Java (Libro: Fundamentos de programación en Java 2 por Mc Herbert Schildt).

JDOM<sup>10</sup> o Documento de Modelado de Objetos en Java, es un API para leer, crear y manipular documentos XML en Java. JDOM, sólo aporta una capa de abstracción en el tratamiento de documentos XML, facilitando la tarea de interpretar DOM y SAX<sup>11</sup> para poder trabajar con XML desde Java. JDOM es específico de la plataforma Java y es una alternativa para DOM y SAX (JDOM, 2012)

Ventajas de JDOM:

- ❖ Utilizada para el lenguaje Java.
- ❖ Facilidad de uso ya que permite la sobrecarga de métodos y colecciones.
- ❖ Representación de documentos como árboles, lo que posibilita el acceso directo a cualquier parte del documento (ventaja sobre SAX).

### 1.8 Gestor de Base Datos SQLite3

SQLite3 es un sistema gestor de base de datos relacional creado por D. Richard Hipp de dominio público escrito en C en una biblioteca aproximadamente de 500Kb, y muy ligero. El formato del archivo que conforma la base de datos que maneja SQLite es multiplataforma, este archivo puede ser escrito en una computadora, copiado a otra y leído en esta sin problemas, incluso con una arquitectura de 32-bit o 64-bit. (Soto, 2009)

Principales Ventajas:

- ❖ No requiere configuración.
- ❖ No se requiere uso de servidor (proceso activo para atender las peticiones).
- ❖ Fácilmente portable (multiplataforma Windows, Linux, Mac, dispositivos móviles, tablets, etc.)
- ❖ Acceso mucho más rápido.
- ❖ Registros de longitud variable.
- ❖ Único archivo de Base de Datos.

---

<sup>10</sup> Acrónimo de **J**ava **D**ocument **O**bject **M**odel (Documento de Modelado de Objetos en Java).

<sup>11</sup> Simple API for XML

La herramienta para administrar los datos que se utilizó fue sqllitestudio versión 2.1.2, este es utilizado para modificar todos los datos de las tablas creadas utilizando la versión de base datos SQLite3.

Se seleccionó SQLite3 como gestor de base de datos, porque la base de datos de la aplicación no constará de numerosas tablas por lo que no se requiere de un gestor más potente, en cuanto al tamaño de almacenamiento.

### 1.9 Herramienta CASE

Para el modelado de los artefactos y diagramas generados a lo largo del ciclo de vida del proyecto se hace necesario utilizar una herramienta CASE. Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, implementación de parte del código automáticamente con el diseño dado, documentación o detección de errores entre otras.

#### 1.9.1 Visual Paradigm

Visual Paradigm en su versión 8.0 es una herramienta profesional que se utiliza para el modelado, soporta el ciclo de vida completo del desarrollo de software: análisis y diseño, construcción, pruebas y despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones. Permite dibujar todos los tipos de diagramas de clases, ingeniería inversa, generar código desde diagramas y generar documentación. (Visual Paradigm, 2011)

Se utiliza esta herramienta debido a sus ventajas:

- ❖ Disponibilidad en múltiples plataformas.
- ❖ Capacidades de ingeniería directa (versión profesional) e inversa.
- ❖ Las imágenes y reportes generados son de buena calidad.

### 1.10 IDE Desarrollo

Un entorno de desarrollo integrado, llamado también IDE (sigla en inglés de Integrated Development Environment), es un programa informático compuesto por un conjunto de herramientas de

programación, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. (Godoy, 2012)

### **1.10.1 NetBeans**

EL IDE NetBeans en su versión 7.3, es un producto libre y gratuito sin restricciones de uso, es un entorno de desarrollo para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación como C/C++, Ruby on Rails, PHP, Groovy, Python, JavaScript entre otros. Permite desarrollar aplicaciones de escritorio, Web, Mobile, Enterprise.

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio de 2000 continúa siendo el patrocinador principal de los proyectos. (NetBeans, 2010)

### **1.10.2 Eclipse**

El Eclipse es un entorno de desarrollo integrado (IDE, Integrated Development Environment) que facilita enormemente las tareas de edición, compilación y ejecución de programas durante su fase de desarrollo. Aunque Eclipse pretende ser un entorno versátil soportando varios lenguajes de programación, es con el lenguaje Java con el que mejor se integra y con el que ha ganado su popularidad. (Martínez, 2007)

Eclipse es una aplicación gratuita y de código abierto, disponible en la red para su descarga e incluida ya en muchas distribuciones de Linux. Eclipse proporciona el entorno de desarrollo solamente, siendo necesario además, para el caso del lenguaje Java, disponer del Java Development Kit o JDK para poder compilar y ejecutar las aplicaciones desarrolladas. (Martínez, 2007)

### **1.10.3 Selección del IDE Desarrollo a utilizar**

Para las funcionalidades previstas en el desarrollo de la solución, las funcionalidades de ambos IDEs mencionados son muy parejas, determinándose que ninguna posee una diferencia apreciable que pueda convertirse en ventaja decisiva. Bajo esta situación se decide utilizar NetBeans por cuanto es ligeramente más cómodo de utilizar para la implementación de aplicaciones visuales de escritorio en Java, y se posee mayor conocimiento de su manejo por parte del equipo de desarrollo.

### **1.11 Conclusiones**

En el presente capítulo se realizó un estudio del estado del arte de las herramientas de extracción de metadatos en documentos digitales, y las diferentes clasificaciones de metadatos que existen, así como las dos representaciones que se utilizan para la descripción bibliográfica en la universidad Marc 21 y Dublin Core. El estudio arrojó que aunque las herramientas estudiadas son potentes para la recuperación de información e indexación, es necesario realizar un sistema de extracción de metadatos que satisfaga las necesidades del repositorio institucional y la biblioteca de la universidad. Se acordó utilizar la librería Apache Tika como base para el desarrollo del sistema por sus principales características mencionadas anteriormente. Se definió como metodología de desarrollo a utilizar RUP con el lenguaje de modelado (UML 1.x), apoyándose en la herramienta CASE Visual Paradigm v8.0. Además se decidió que el lenguaje de programación será Java con el IDE de desarrollo NetBeans v7.3.

### Capítulo 2: Diseño

#### 2.1 Introducción

En el presente capítulo se hace referencia a la presentación de la propuesta de solución, con el objetivo de entender el problema de la investigación. Se representa el modelo de dominio con todas las clases conceptuales con que se trabajan, además se enumeran los requerimientos funcionales y no funcionales. Posteriormente se realiza la descripción de los actores y los casos de uso del sistema para el desarrollo del modelado del diseño.

#### 2.2 Propuesta del Sistema

El desarrollo de la herramienta para la extracción de metadatos será de gran utilidad para los administradores del SIGB de la Universidad de las Ciencias Informáticas, ya que los librará de todo trabajo manual a la hora de realizar la catalogación de documentos digitales. La herramienta cuenta con 2 funcionalidades principales. La primera funcionalidad es que permite realizar la extracción masiva de metadatos al conjunto de documentos en formato .pdf que el usuario cargue, y validar los metadatos que son extraídos, mostrándolos en una lista. La segunda funcionalidad es exportar los metadatos en un formato estándar XML con los campos de datos del registro bibliográfico Marc 21 y Dublin Core estos son utilizados por la biblioteca y el repositorio institucional de la universidad. Además brinda la opción de generar una estructura de carpeta en el caso de Dublin Core para almacenar los metadatos de cada documento y que estos sirvan para ser insertados en el Repositorio Institucional DSpace.

#### 2.3 Modelo de Dominio

El modelo del dominio muestra a los modeladores clases conceptuales significativas en un dominio del problema; es el artefacto más importante que se crea durante el análisis orientado a objetos. *“Un modelo del dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés.”* (Larman, 1999)(9)

A continuación se muestra el diagrama del modelo de dominio:

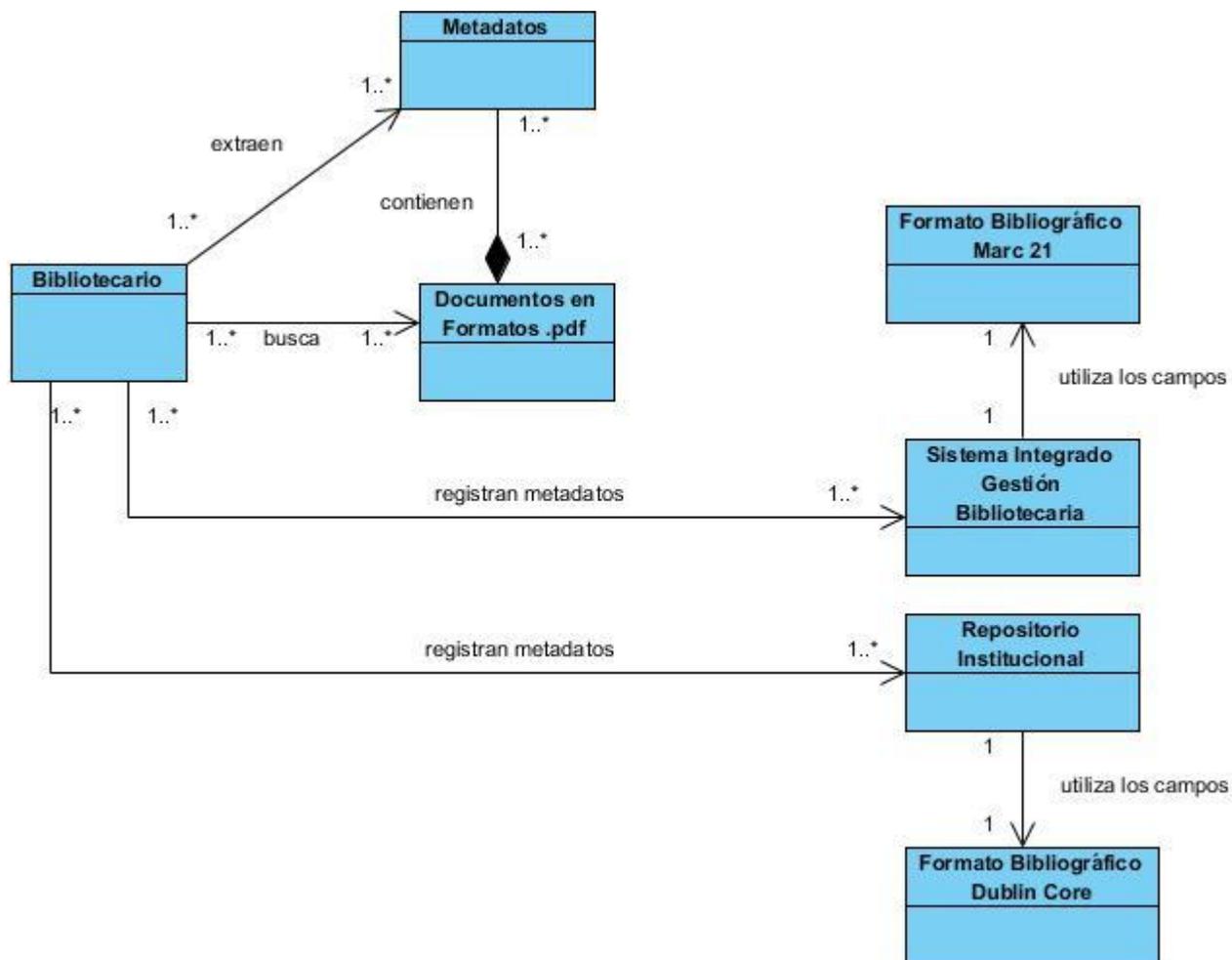


Figura 2. 1 Proceso de Extracción de Metadatos

Descripción de las clases del dominio:

Bibliotecario: Usuario que interactúa con el sistema.

Documentos en Formatos pdf: Documentos en formatos .pdf que carga el bibliotecario para realizar el proceso de extracción.

Metadatos: Son un conjunto de atributos o elementos necesarios que describen un documentos determinado, estos metadatos funcionan como identificador de los materiales digitales.

Formato Bibliográfico Marc21: Norma para la representación de la información bibliográfica.

Formato Bibliográfico Dublin Core: Formato para la representación de metadatos en el Repositorio de objetos digitales.

SIGB: Sistema Integrado de Gestión Bibliotecaria, el cual será utilizado para realizar el registro de los formatos bibliográficos.

Repositorio Institucional: Servicio de almacenamiento de objetos digitales.

### **2.4 Requisitos del Sistema**

Los requerimientos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. (Sommerville, 2005)

Un requerimiento funcional no es más que condiciones que el sistema debe cumplir. A continuación se muestran los requisitos funcionales:

#### **2.4.1 Requerimientos funcionales**

RF1. Autenticar usuario.

RF2. Seleccionar el idioma inglés o español.

RF3. Modificar contraseña del usuario autenticado.

RF4. Buscar documentos a revisar para la extracción de metadatos.

RF5. Listar documentos buscados.

RF6. Seleccionar todos los documentos.

RF7. Modificar selección de documentos.

RF8. Eliminar documento del listado.

RF9. Extraer metadatos de los documentos.

RF10. Listar los ficheros con errores de formato.

RF11. Listar los ficheros correspondientes a cada documento con los metadatos extraídos.

RF12. Visualizar metadatos extraídos de cada fichero.

RF13. Eliminar fichero que contiene metadatos.

RF14. Generar fichero Marc XML con los metadatos utilizando los campos de Marc 21.

RF15. Generar fichero XML con los metadatos utilizando los campos de Dublin Core.

RF16. Generar estructura de carpeta para almacenar los metadatos en el formato Dublin Core

RF17. Crear usuario.

RF18. Buscar usuario.

RF19. Listar usuarios.

RF20. Modificar usuario.

RF21. Eliminar usuario.

RF22. Ver historial de extracciones de metadatos de un usuario.

RF23. Eliminar historial de extracciones de metadatos de un usuario.

### 2.4.1.1 Validación de los requisitos funcionales

El proceso de validación de requisitos comprende actividades que generalmente se realizan una vez obtenida una primera versión de la documentación de requisitos. Este proceso tiene por finalidad comprobar que los requisitos del software poseen todos los atributos de calidad si son consistentes, completos, precisos, realistas, verificables y definen lo que el usuario desea del producto final. (Compendio, 2004)

Los métodos que se utilizaron para la validación de los requisitos son: (Compendio, 2004)

- ❖ Revisión de requisitos.
- ❖ Prototipado.
- ❖ Generación de casos de prueba (test de requisitos).

Se realizó el método de revisión de requisitos que consiste en una o varias reuniones planificadas por el equipo de analistas del proyecto SIGB y las autoras del presente trabajo (desarrolladores de la herramienta) donde se confirmó que los requisitos poseen los atributos de calidad deseados. Los

requisitos fueron evaluados a través de los criterios que presenta el Expediente de Proyecto. El método prototipo se utilizó para la creación de una maqueta o versión del producto final, teniendo a través de este una visión sobre cómo quedaría el sistema, el que se utilizó en específico fue Mock-ups que consiste en representar dibujando a mano en un papel el sistema pensado. También se realizaron casos de prueba como se muestra en el capítulo 3 epígrafe 3.4.3 para verificar que los requisitos estén bien enunciados, sean consistentes y estén razonablemente completos.

### **2.4.2 Requerimientos no funcionales**

En este epígrafe se hará referencia a los requisitos no funcionales. Los requisitos no funcionales no van asociados a casos de uso concreto y consisten en restricciones impuestas por el entorno y tecnologías, especificaciones sobre tiempo de respuesta o volumen de información tratado por una unidad de tiempo, requisitos en cuanto a interfaces, extensibilidad y facilidad de mantenimiento. (Falgueras, 2003)

#### **Apariencia o interfaz externa**

La interfaz posee una tonalidad de colores azul y blanco.

#### **Usabilidad**

Presenta un menú compuesto por elementos que representan de forma gráfica y textual, cada opción que permite el sistema.

#### **Seguridad**

El sistema garantizará la autenticación como primera acción. Esta consistirá en suministrar un nombre de usuario único y una contraseña.

Los usuarios estarán autorizados a realizar las acciones que se encuentran definidas para el rol al cual pertenece.

## **2.5 Modelo de casos de uso del sistema**

### **2.5.1 Descripción del actor**

Un actor es una entidad externa que interacciona con el sistema participando en un caso de uso. Los actores pueden ser gente real, otros ordenadores o eventos externos.

Los actores no representan a personas físicas o a sistemas, sino su rol. Esto significa que cuando una persona interactúa con el sistema de diferentes maneras (asumiendo diferentes papeles), estará representado por varios actores. (Documentación)

<b>Actor</b>	<b>Descripción</b>
Usuario	Tiene privilegios, una vez autenticado en el sistema, para cargar los documentos y realizar la extracción de metadatos, mostrar los metadatos extraídos, cambiar contraseña, además de generar y guardar los metadatos extraídos en el formato MarcXML y Dublin Core.
Administrador	Tiene privilegios, una vez autenticado en el sistema, de hacer todas las operaciones que hace el usuario, pero además puede crear usuario, modificar usuario y ver el historial de extracciones de metadatos de un usuario.

**Tabla 2. 1 Descripción de los actores del sistema**

### **2.5.2 Definición de los casos de uso del sistema**

Un caso de uso es una descripción narrativa de un proceso de dominio. (Larman, 1999)

Una vez identificados los actores del sistema se pasa a la elaboración del diagrama de casos de uso del sistema. A continuación se muestra el diagrama de casos de uso del sistema:

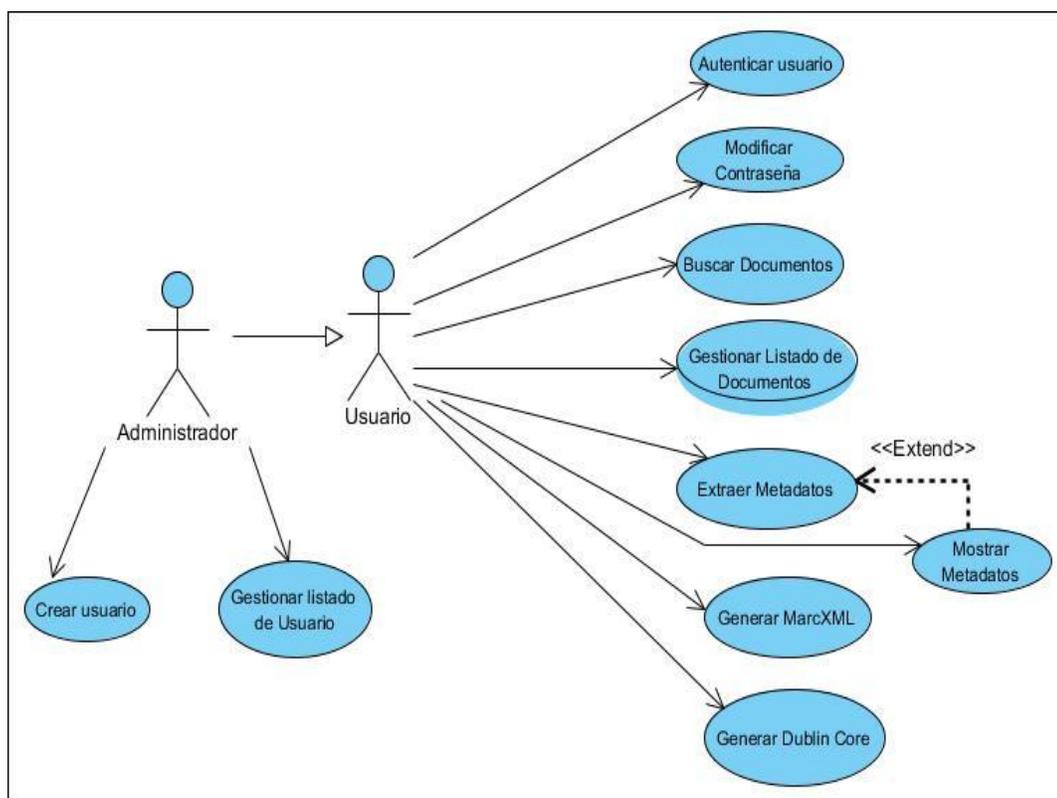


Figura 2. 2 Diagrama de Casos de Uso

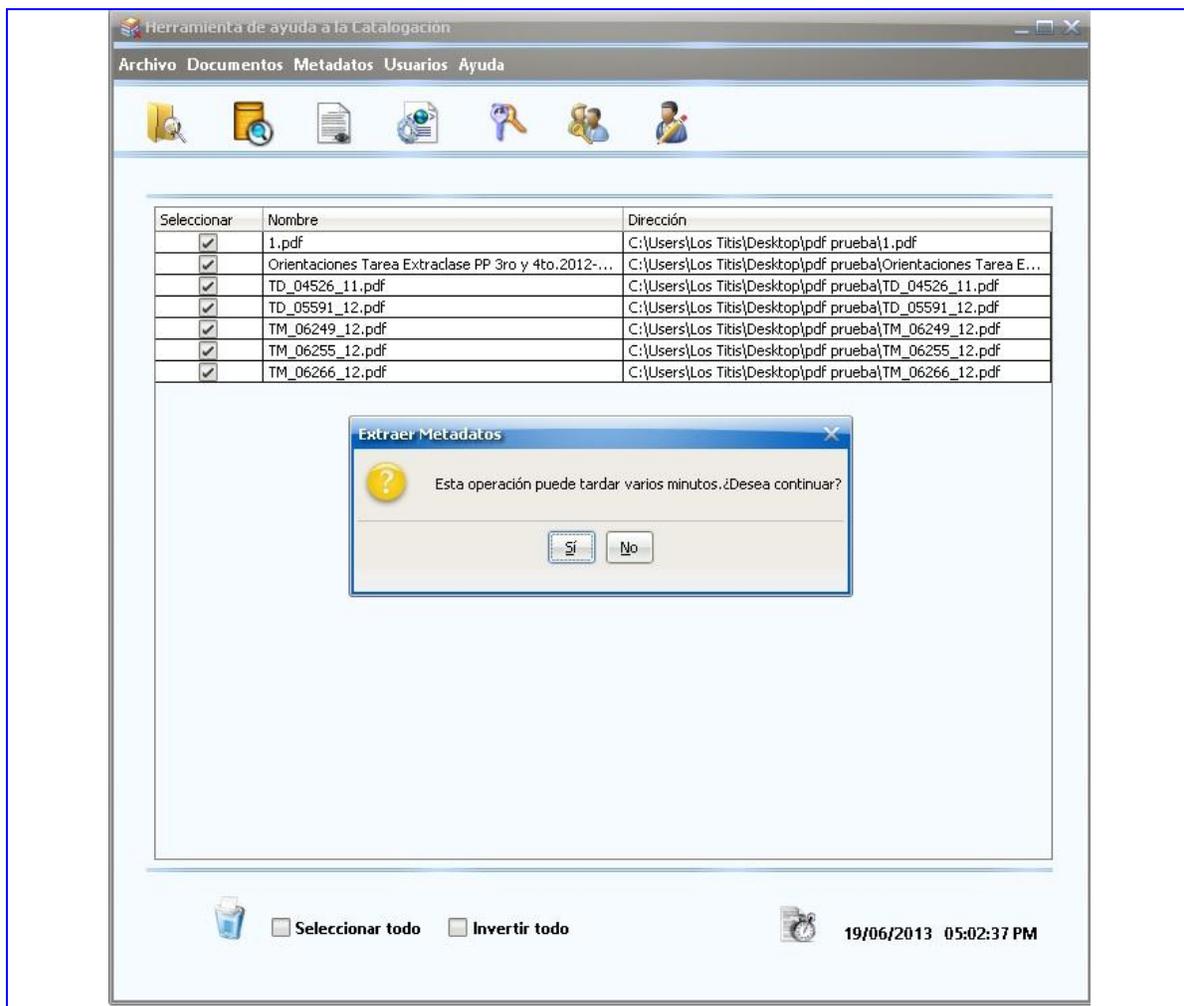
### 2.5.2.1 Especificación de Casos de Uso

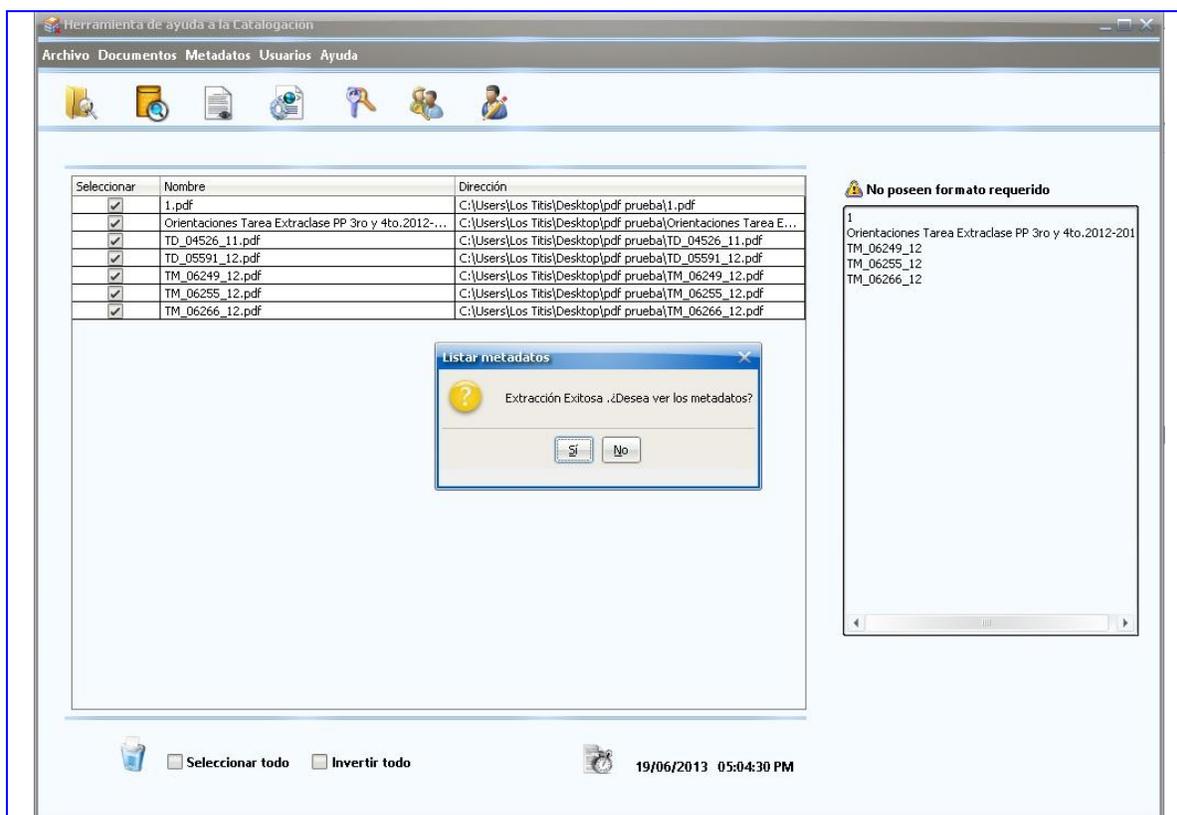
A continuación se muestran las descripciones de los casos de usos más importantes referente al tema de la investigación realizada, las restantes descripciones se encuentran a partir del Anexo 4:

#### CU1.Extraer Metadatos:

<b>Nombre</b>	Extraer Metadatos
<b>Actores</b>	Usuario
<b>Resumen</b>	El caso de uso inicia cuando el usuario selecciona los documentos que se le quiere extraer los metadatos y ejecuta la opción extraer metadatos, mostrando después un listado con los documentos que tienen errores de formatos, y un listado de ficheros con los metadatos extraídos de cada uno, finalizando así el caso de uso.
<b>Complejidad</b>	Baja
<b>Prioridad</b>	Baja

<b>Referencias</b>	RF9,RF10	
<b>Precondiciones</b>	Deben estar seleccionados los documentos a los que desea extraerle los metadatos.	
<b>Postcondiciones</b>	El sistema muestra el listado de ficheros con los metadatos extraídos.	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt;Extraer Metadatos&gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción extraer metadatos.	
2.		Muestra un mensaje: "Esta operación puede tardar varios minutos". Desea continuar.
3.	Selecciona la opción si para continuar el proceso de extracción.	
4.		Extrae los metadatos de los documentos especificados.
5.		Muestra un listado con los ficheros que no poseen el formato requerido: <ul style="list-style-type: none"> <li>- Trabajo de diploma para optar...</li> <li>- Tesis para optar por el título académico...</li> </ul>
6.		Muestra un mensaje indicando si desea ver los metadatos extraídos.
7.	Indica que si desea ver los metadatos.	
8.		Desmarca los documentos que inicialmente fueron seleccionados para extraerle los metadatos.
9.		Se ejecuta el caso de uso "Mostrar Metadatos", acción 3 del flujo básico.
10.		Termina el caso de uso.
<b>Prototipo de Interfaz</b>		





**Flujo alternativo del flujo básico < Extraer Metadatos >Cancelar**

	Actor	Sistema
3.a	Indica que no desea continuar con la extracción de metadatos y termina el caso de uso.	

**Flujo alternativo del flujo básico < Extraer Metadatos >Cancelar Proceso de Extracción**

	Actor	Sistema
7.a	Indica que no desea ver los metadatos y termina el caso de uso.	

Tabla 2. 2 Descripción del Caso de Uso Extraer Metadatos

**CU7.Mostra Metadatos:**

<b>Nombre</b>	Mostrar metadatos
<b>Actores</b>	Usuario
<b>Resumen</b>	El caso de uso se inicia cuando el usuario selecciona que desea ver los metadatos de los documentos y finaliza cuando se

	muestra en un listado los ficheros y sus metadatos asociados.	
<b>Complejidad</b>	Baja	
<b>Prioridad</b>	Baja	
<b>Referencias</b>	RF11,RF12,RF13	
<b>Precondiciones</b>	Se debe haber realizado la extracción de metadatos a los documentos seleccionados.	
<b>Postcondiciones</b>	Se muestra el listado de documentos y sus metadatos asociados.	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt; Mostrar metadatos &gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona mostrar metadatos extraídos.	
2.		Verifica que existan metadatos extraídos.
3.		Muestra un listado de ficheros con los metadatos extraídos de cada documento permitiendo realizar varias acciones: <ul style="list-style-type: none"> <li>- Seleccionar todos los ficheros. Ver Sección 1: Seleccionar todos los ficheros.</li> <li>- Visualizar metadatos. Ver Sección 2: Visualizar metadatos.</li> <li>- Eliminar ficheros. Ver Sección 3: Eliminar ficheros.</li> </ul>
4.		Termina el caso de uso.
<b>Flujos alternos del flujo básico &lt; Mostrar metadatos &gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
2.a		Si no se ha realizado la extracción de los metadatos muestra un mensaje indicando que no existen metadatos y termina el caso de uso.
<b>Sección 1: “Seleccionar todos los ficheros”</b>		
<b>Flujo básico &lt; Mostrar Metadatos&gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Indica seleccionar todos los ficheros.	
		Marca todos los ficheros en el listado.
2.		Se ejecuta la acción 3 del Flujo básico.
<b>Sección 2: “Visualizar metadatos”</b>		
<b>Flujo básico &lt; Mostrar Metadatos&gt;</b>		

	Actor	Sistema
1.	Selecciona el fichero que desea ver sus metadatos.	
2.		Muestra los metadatos del fichero seleccionado.
3.		Se ejecuta la acción 3 del Flujo básico.
<b>Flujos alternos de la Sección 2: “Visualizar metadatos”</b>		
	Actor	Sistema
1.a		Si no existen documentos seleccionados muestra un mensaje indicando que se deben seleccionar los ficheros para ver sus metadatos y se ejecuta la acción 3 del Flujo básico.
<b>Sección 3: “Eliminar ficheros”</b>		
<b>Flujo básico &lt; Mostrar Metadatos&gt;</b>		
	Actor	Sistema
1.	Selecciona los ficheros que desea eliminar.	
2.		Verifica que existan ficheros seleccionados.
3.		Muestra mensaje de confirmación para eliminar.
4.	Indica que desea eliminar.	
5.		Elimina los ficheros seleccionados.
6.		Se ejecuta la acción 3 del flujo básico.
<b>Flujos alternos de la Sección 3: “Eliminar ficheros”</b>		
	Actor	Sistema
2.a		Si no existen ficheros seleccionados muestra un mensaje indicando que se deben seleccionar los ficheros a eliminar y se ejecuta la acción 3 del flujo básico.
4.a	Indica que no desea eliminar.	
4.b		Ejecuta la acción 3 del flujo básico.
<b>Prototipo de Interfaz</b>		

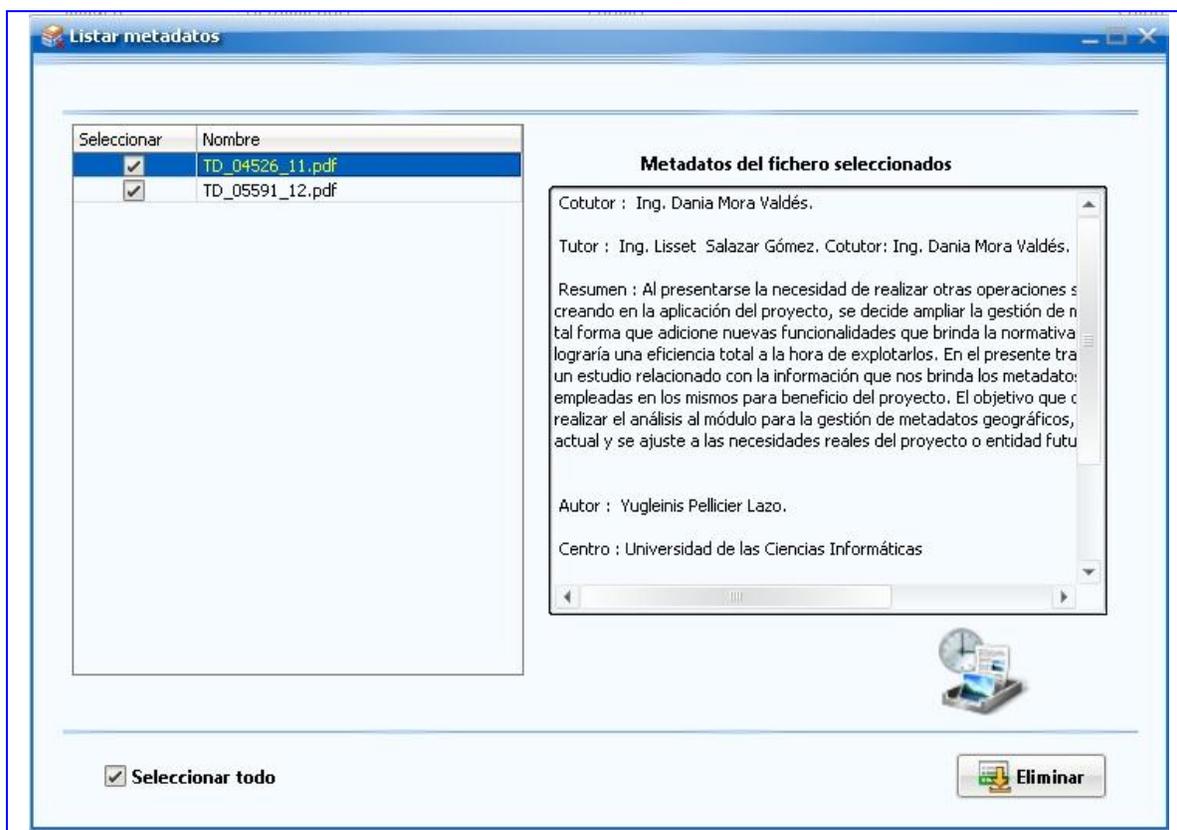


Tabla 2. 3 Descripción del Caso de Uso Mostrar Metadatos

## 2.6 Descripción de la Arquitectura

La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente, los principios que orientan su diseño y evolución. (University, 2013). La arquitectura nos permite establecer una línea guía que sea común para el equipo de trabajo a través de la utilización de patrones.

La arquitectura está representada mediante un modelo en capas (Presentación, Negocio y Acceso a Datos). La capa de Negocio presenta una arquitectura modular, donde cada componente separado realiza una actividad especializada y su funcionamiento es coordinado por un componente Mediador central, que a la vez funge como punto único de entrada para todos los componentes de la capa de Presentación y la capa de Acceso a Datos es la que se ocupa de obtener y persistir los datos.

La arquitectura así descrita contiene un número de patrones de diseño subordinados al patrón principal Mediator, que constituye la piedra angular de la organización y relaciones de todos los

componentes de negocio. A continuación se representa la estructura de la arquitectura adaptada a la solución propuesta:



Figura 2. 3 Representación de la arquitectura

### 2.7 Patrones de Diseño

*“Un patrón es una descripción de un problema y la solución a la que se da un nombre, y que se puede aplicar a nuevos contextos; idealmente, proporciona consejos sobre el modo de aplicarlo en varias circunstancias.”* (Larman, 1999) (9)

Los patrones de diseño pueden usarse durante el diseño del software. Una vez que se ha desarrollado el modelo de análisis, el diseñador puede examinar una representación detallada del problema que debe resolver y las restricciones que impone el problema. (Pressman, 2002)

#### Patrones GRASP

Los patrones GRASP<sup>12</sup> describen los principios fundamentales de la asignación de responsabilidades a objetos, además constituyen un apoyo para entender el diseño y aplica el razonamiento para el diseño de una forma sistemática, racional y aplicable. (Larman, 1999)

#### Patrón Creador

*“La creación de instancias es una de las actividades más comunes en un sistema orientado a objetos. En consecuencia, es útil contar con un principio general para la asignación de las responsabilidades*

<sup>12</sup> GRASP es un acrónimo que significa General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades)

*de creación. Si se asignan bien, el diseño puede soportar un bajo acoplamiento, mayor claridad, encapsulación y reutilización.” (Larman, 1999) (9)*

En el sistema este patrón se evidencia en el componente GeneratorBuilder, el cual tiene la responsabilidad de crear y configurar los generadores de XML en formatos Dublin Core y Marc21 según decida el usuario. La aplicación de este patrón GRASP es la causa raíz de la aplicación del patrón Gof Builder.

### **Patrón Bajo Acoplamiento**

El acoplamiento es una medida de fuerza con que un elemento está conectado a, tiene conocimiento de, confía en, otros elementos. Un elemento con bajo (o débil) acoplamiento no depende de demasiados elementos. Estos elementos pueden ser clases, subsistemas o sistemas. (Larman, 1999)

La aplicación del patrón Gof Mediator, que proporciona un punto único de contacto para todos los módulos de la aplicación, también permite que se mantenga el bajo acoplamiento entre estos componentes especializados. Ningún componente de la aplicación conoce a ninguno de los demás, comunicándose solo con el componente central de control.

### **Patrón Experto**

Experto es un patrón que se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a objetos. Con él no se pretende designar una idea oscura ni extraña; expresa simplemente la "intuición" de que los objetos hacen cosas relacionadas con la información que poseen. (Larman, 1999)

Cada componente de la aplicación posee una función especializada, por ejemplo el TikaManager es el encargado de gestionar el procesamiento de metadatos a partir de archivos mediante el uso de la librería Tika, mientras que el GeneratorBuilder es el encargo de decidir, a partir de la configuración de la aplicación y el formato especificado por el usuario que generador de XML será utilizado para procesar los archivos. Al separar de esta manera las responsabilidades también se separa la información que cada uno de estos componentes debe poseer para realizar su trabajo y se aíslan los posibles cambios en islotes separados que facilitan el mantenimiento y la refactorización.

### **Patrones GOF<sup>13</sup>:**

A principios de los años 90 con la publicación del libro Design Patterns, se establecen veintitrés patrones de diseño Gof. Los patrones Gof se descubren como una forma indispensable de enfrentarse a la programación a raíz del libro “Design Patterns-Elements of Reusable Software” de Erich Gamma, Richard Helm, Ralph Jonson y John Vlissides, a partir de entonces estos patrones son conocidos como los patrones de la pandilla de los cuatro (GoF, gang of four). (Pressman, 2002)

Los patrones GOF según su propósito se clasifican en tres grupos. Los creacionales tratan la creación de instancias. Los estructurales tratan la relación entre clases, la combinación clases y la formación de estructuras de mayor complejidad. Mientras que los de comportamiento tratan la interacción y cooperación entre clases. A continuación se muestran los que se utilizaron en el desarrollo del sistema:

### **Patrón Singleton:**

El Singleton o única instancia es un patrón que asegura que una clase tiene una sola instancia y proporciona en ocasiones una visibilidad global o un único punto de acceso a ella en lugar de cualquier otra visibilidad. (Larman, 1999)

En el sistema este patrón se evidencia en la clase Controller que tiene la responsabilidad de controlar todas las funcionalidades del sistema, y restringe la creación de objetos pertenecientes al resto de las clases.

### **Patrón Mediator**

Este patrón define un objeto que encapsula la forma en que interactúan un grupo de objetos, promoviendo así un acoplamiento débil al evitar las referencias explícitas entre los objetos y permitiendo, por tanto, que su interacción se modifique de forma independiente. (Informática)

En el sistema este se evidencia en la clase Controller al ser la que interactúa con el resto de las clases.

### **Patrón Builder**

---

<sup>13</sup> Del inglés *Gang of Four* o Banda de Cuatro.

El patrón de construcción, es uno más de los patrones creacionales de diseño. Separar la construcción y la representación de un objeto complejo, para así permitir que el mismo proceso de construcción sirva para crear diferentes representaciones. (Informática)

Este patrón permite separar la construcción de un objeto complejo de su representación, de modo que el mismo proceso de construcción pueda crear diferentes representaciones de este objeto. Este se evidencia en la clase `GeneratorBuilder` que tiene la responsabilidad de crear y configurar los generadores de XML con sus propios objetos.

### **2.8 Modelos de Diseño**

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales tienen impacto en el sistema a desarrollar. Además sirve de abstracción de la implementación del sistema y es, de ese modo, utilizada como entrada fundamental de las actividades de implementación. (Jacobson, y otros, 2000)

#### **2.8.1 Diagrama de Clases del Diseño**

El diagrama de clases de diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. (Jacobson, y otros, 2000) A continuación se muestra el diagrama de clases del diseño:



### **2.8.2 Diagrama de Secuencia**

Los diagramas de secuencia incluyen la interacción de los mensajes entre los objetos que se definen en el modelo conceptual y otras clases de objetos. (Rodríguez, 1999)

Un diagrama de secuencia muestra una interacción que está organizada como una secuencia temporal. En particular, muestra los objetos que participan en la interacción mediante sus líneas de vida y mediante los mensajes que se intercambian, organizados en forma de una secuencia temporal.

A continuación se muestra el diagrama de secuencia del Caso de Uso: “Extraer Metadatos”:

#### **Diagrama de Secuencia :“Extraer Metadatos”**

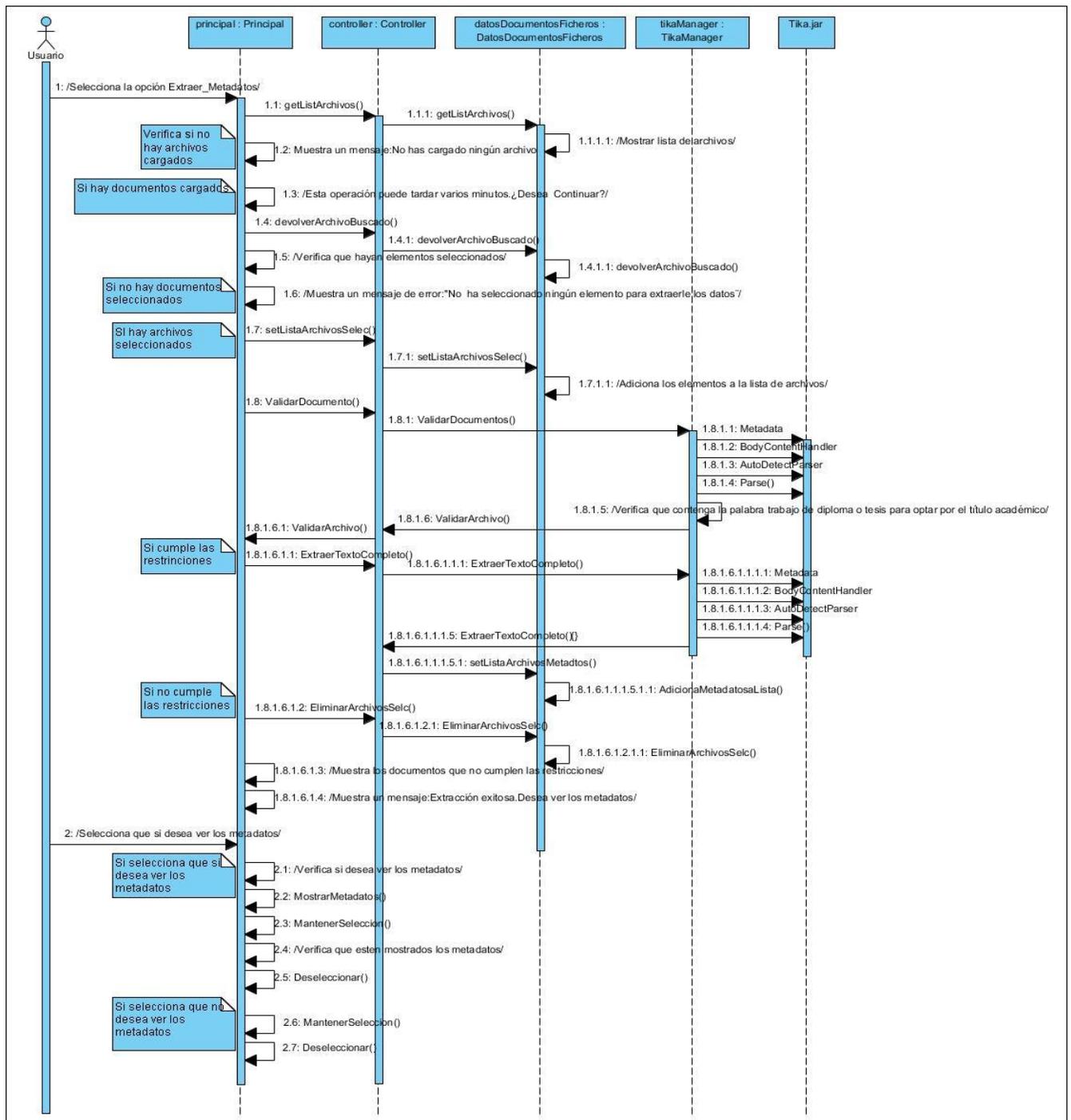


Figura 2. 5 Diagrama de Secuencia del Caso de Uso Extraer Metadatos

**Caso de Uso Extendido Mostrar Metadatos**

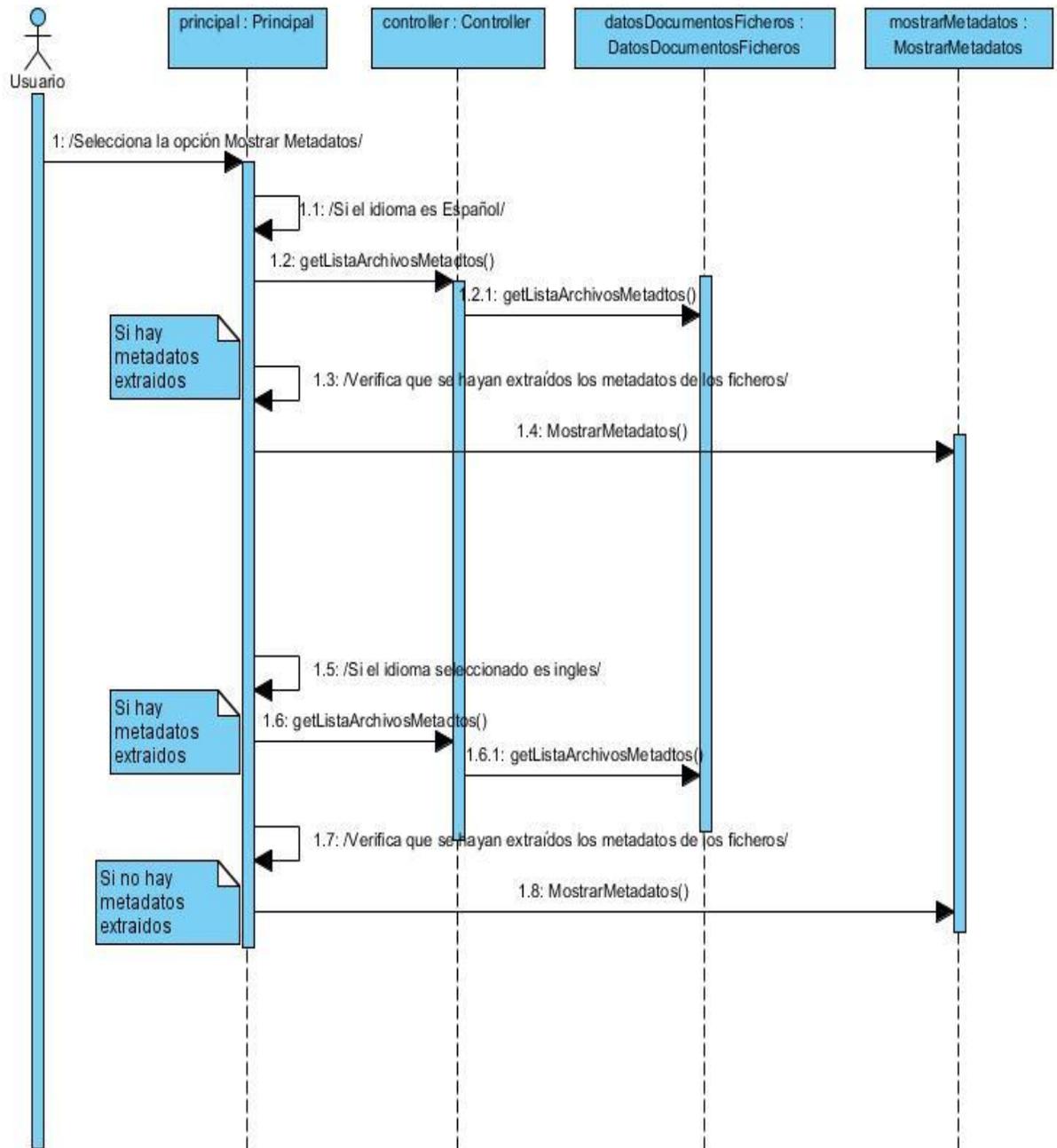


Figura 2. 6 Diagrama de Secuencia del Caso de Uso Extendido Mostrar Metadatos

**Sección 1: “Seleccionar todos los ficheros “del CU Mostrar Metadatos**

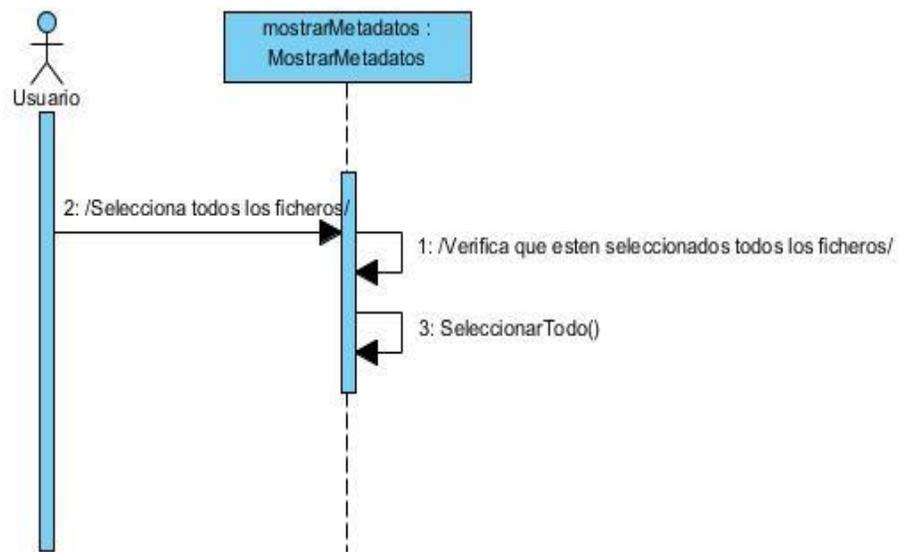


Figura 2. 7 Diagrama de Secuencia Sección 1 “Seleccionar todos los ficheros “

**Sección 2: “Visualizar metadatos” del CU Mostrar Metadatos**

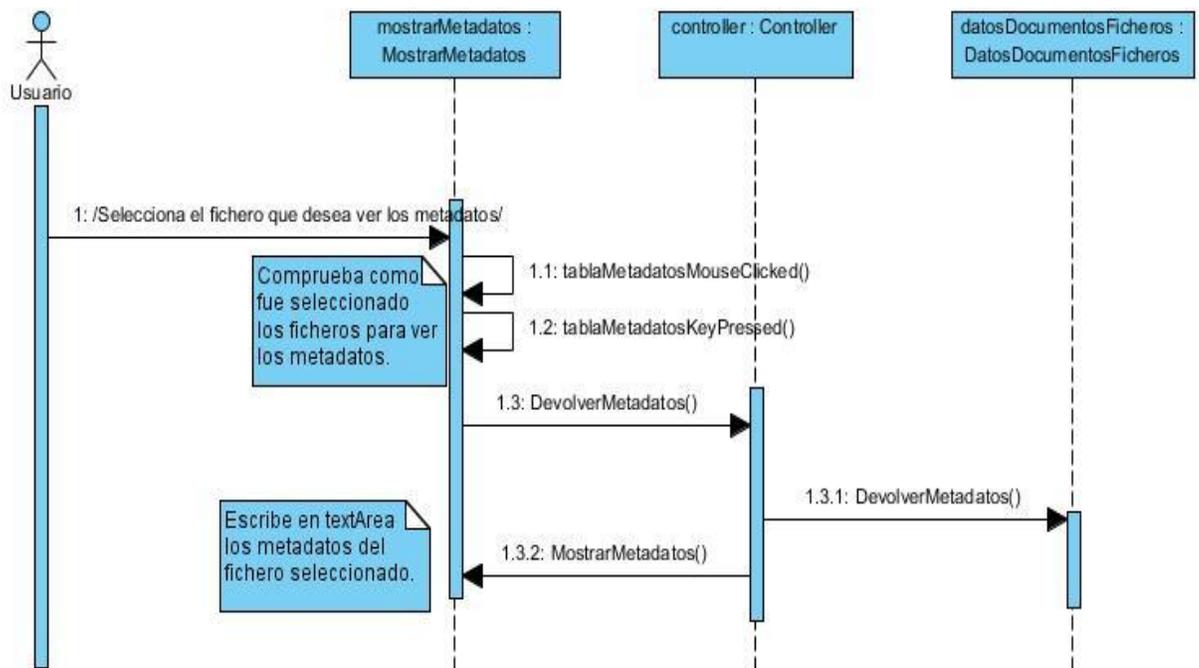


Figura 2. 8 Diagrama de Secuencia Sección 2 “Visualizar metadatos”

**Sección 3: “Eliminar Ficheros” del CU Mostrar Metadatos**

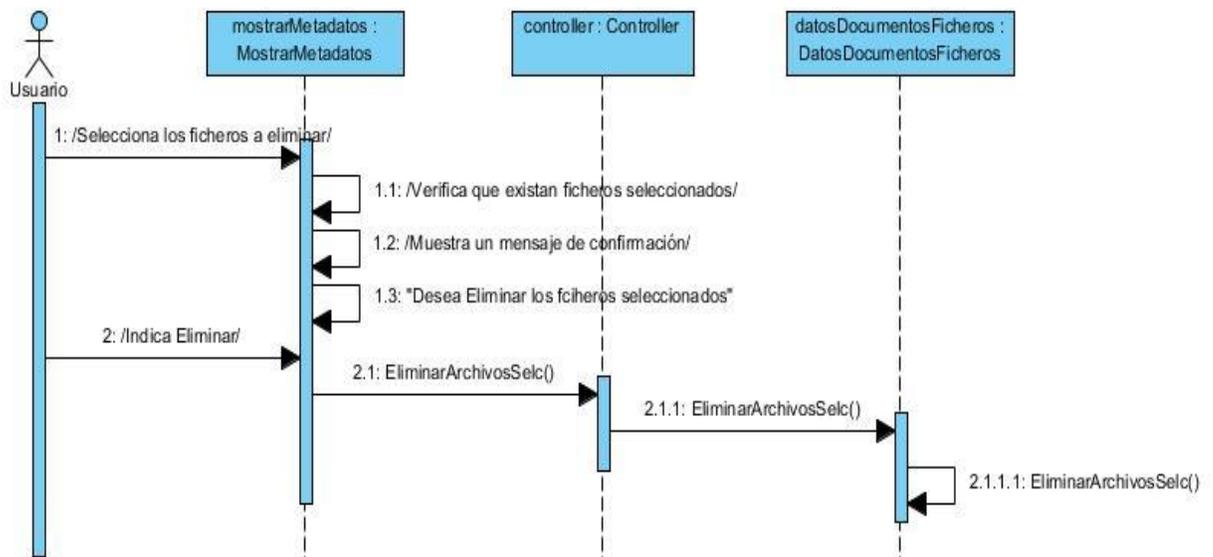


Figura 2. 9 Diagrama de Secuencia Sección 3 “Eliminar Ficheros”

### 2.9 Modelo Entidad-Relación

El Modelo de Entidad Relación es un modelo de datos basado en una percepción del mundo real que consiste en un conjunto de objetos básicos llamados entidades y relaciones entre estos objetos, implementándose en forma gráfica a través del Diagrama Entidad Relación. (Storti, et al., 2007)

Se puede definir como entidad a *“cualquier objeto, real o abstracto, que existe en un contexto determinado o puede llegar a existir y del cual deseamos guardar información.”* (Storti, et al., 2007)(10)

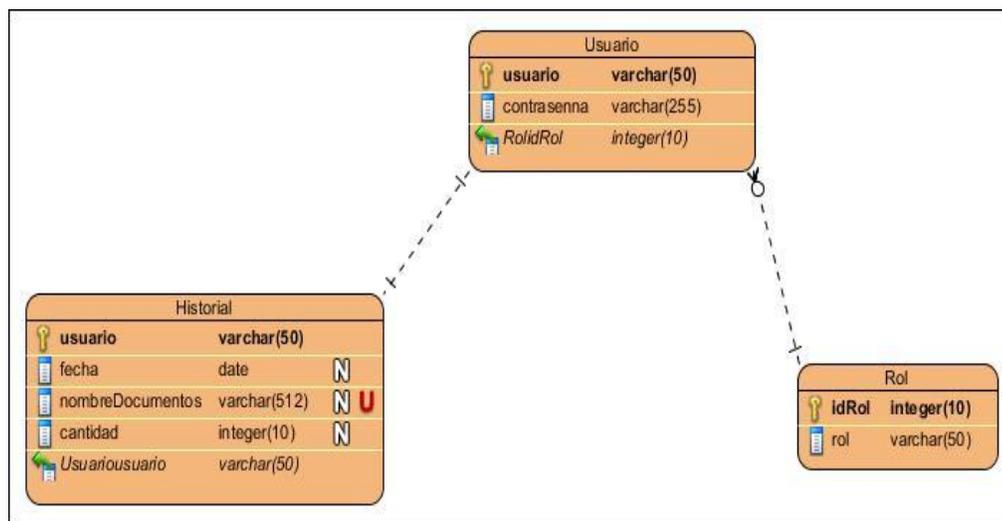


Figura 2. 10 Modelo de Datos

## 2.10 Conclusiones

En el presente capítulo se desarrolló la propuesta de solución del sistema, obteniéndose a partir de un análisis las funcionalidades que debe tener la herramienta. Se realizó el modelo de dominio mostrándose todas las clases conceptuales utilizadas, además se realizó el diagrama de caso de uso. Para guiar la creación de la herramienta se definió la arquitectura y los patrones de diseño. Al finalizar el capítulo se confeccionó el diagrama de clases del diseño con el objetivo de especificar correctamente las clases de software que se deben utilizar para dar paso a la implementación. Por último se realizó el modelo de entidad relacional para representar las tablas de la base datos que se utilizaron.

## Capítulo 3: Implementación y Prueba

### 3.1 Introducción

En este capítulo se describe cómo los elementos del modelo del diseño se implementan en términos de componentes. Además de llevar a cabo el proceso de pruebas al sistema, en el cual se expresan las diferentes pruebas realizadas. También se realizaron los casos de prueba, obteniendo un resultado de estas pruebas realizadas a la aplicación.

### 3.2 Diagrama de Componentes

Un diagrama de componentes representa las dependencias lógicas entre componentes software, sean éstos componentes de código fuentes, binarios o ejecutables. Algunos componentes existen en tiempo de compilación, enlace o ejecución. (Rumbaugh, y otros, 2000). A continuación se representa el diagrama de componente:

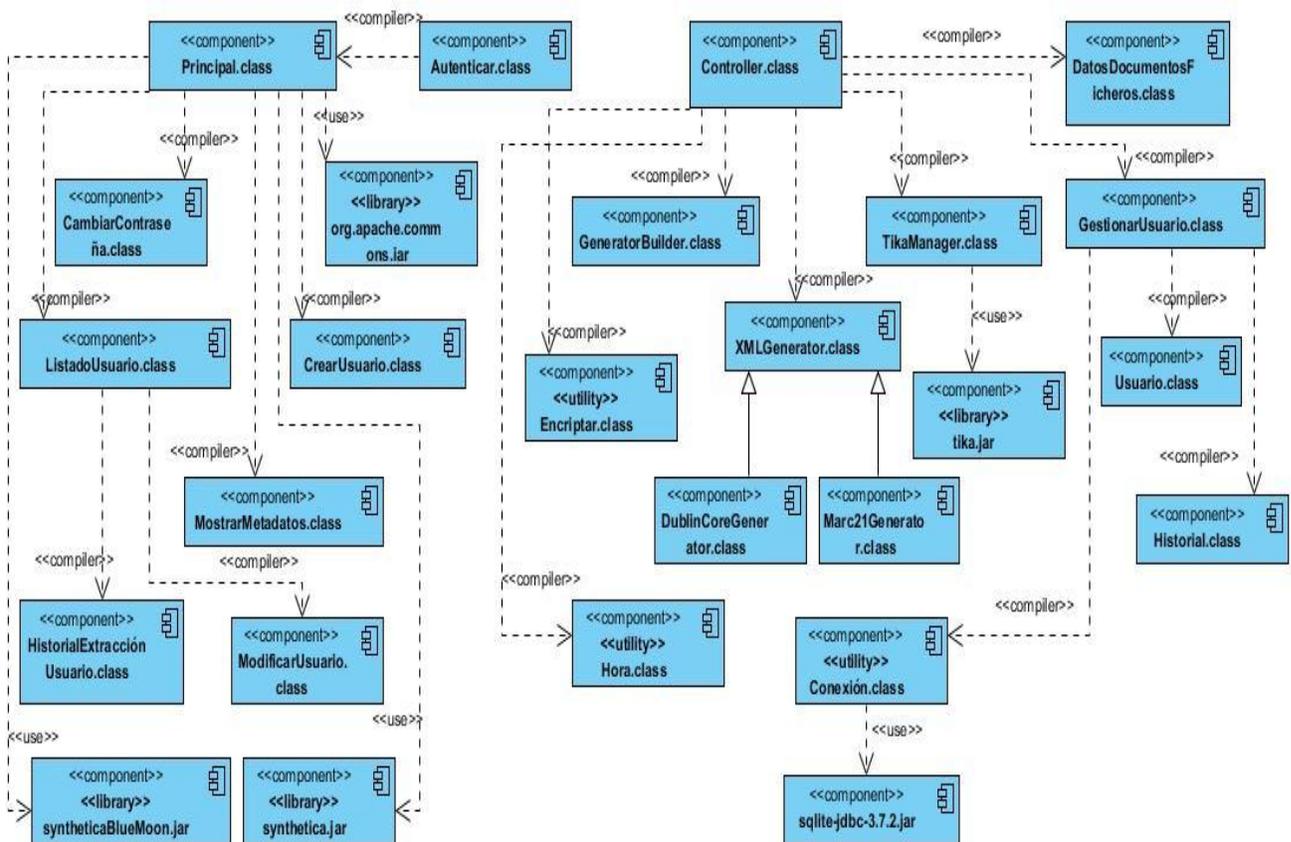


Figura 3. 1 Diagrama de Componente

### 3.3 Diagrama de Despliegue

El diagrama de despliegue muestra la configuración de los nodos de proceso en el tiempo de ejecución, los enlaces de comunicación entre ellos, y las instancias de componente y los objetos que residen en ellas. (Rumbaugh, y otros, 2000)

A continuación se muestra el diagrama de despliegue del sistema, el cual está compuesto por un nodo PC\_Cliente que representa las estaciones de trabajo de los usuarios que utilizarán la aplicación. Dicho nodo necesita tener copiada la aplicación y el fichero de la base de dato catalogar y la plantilla properties para un funcionamiento correcto del sistema.

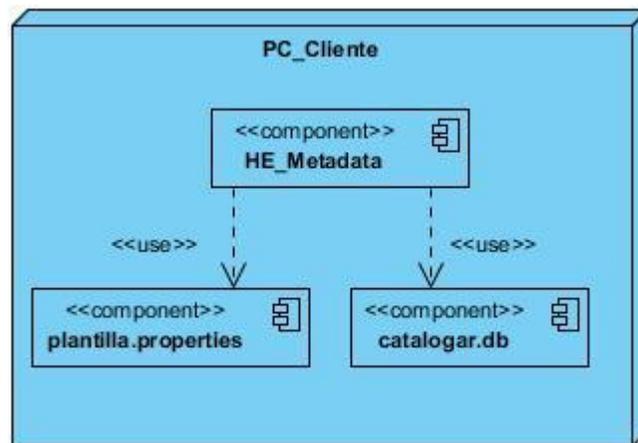


Figura 3. 2 Diagrama de Despliegue

### 3.4 Prueba

Proceso de evaluar un producto particular para determinar si un producto tiene defectos. Las pruebas de Software es un proceso de evaluar un sistema ya sea manual o automático y verificar que este satisface los requisitos o identifica diferencias entre lo esperado y los resultados actuales. *“La prueba no puede asegurar la ausencia de defectos: sólo puede demostrar que existen defectos en el software”*. (Tabares, 2011)(11)

#### 3.4.1 Pruebas Unitarias

Las pruebas unitarias son las encargadas de detectar errores en los datos, y en la lógica. Estas aseguran que un único componente de la aplicación produzca una salida correcta para una

determinada entrada. Este tipo de pruebas validan la forma en la que las funciones y métodos trabajan en cada caso particular.

Se realizan pruebas unitarias porque (Mellado, 2004):

- ❖ Aseguran la calidad del código entregado es la mejor forma de detectar errores tempranamente en el desarrollo. No obstante, esto no asegura detectar todos los errores, por tanto las pruebas de integración y aceptación siguen siendo necesarias.
- ❖ Permiten encontrar errores en los inicios del desarrollo, teniendo en cuenta que mientras más temprano se corrijan los errores, menos costará corregirlos.
- ❖ Fomentan el cambio, las pruebas unitarias facilitan que el programador cambie el código para mejorar su estructura (lo que se ha dado en llamar refactorización), puesto que permiten hacer pruebas sobre los cambios y así asegurarse de que los nuevos cambios no han introducido errores.
- ❖ Las pruebas funcionales se hacen más sencillas pues la mayoría de los aspectos individuales de cada unidad ya están probados a través de las pruebas unitarias. De este modo, las pruebas funcionales deben centrarse solo en verificar la correcta cooperación de las distintas unidades y en los funcionamientos generales del programa.

Para la realización de pruebas unitarias existen herramientas y entornos de desarrollo (frameworks) que facilitan su creación en multitud de lenguajes de programación. Se utilizará para la automatización de las pruebas unitarias JUnit en su versión 4.0, para probar cada una de las funcionalidades del sistema implementado.

NetBeans JUnit es un conjunto de bibliotecas creadas por Erich Gamma y Kent Beck utilizadas en programación para hacer pruebas unitarias de aplicaciones Java. JUnit es un conjunto de clases que permite realizar la ejecución de clases Java de manera controlada, para poder evaluar si el funcionamiento de cada uno de los métodos de la clase se comporta como se espera. (Peña, 2010)

A continuación se muestra un ejemplo de las pruebas realizadas a la aplicación con JUnit 4.0:

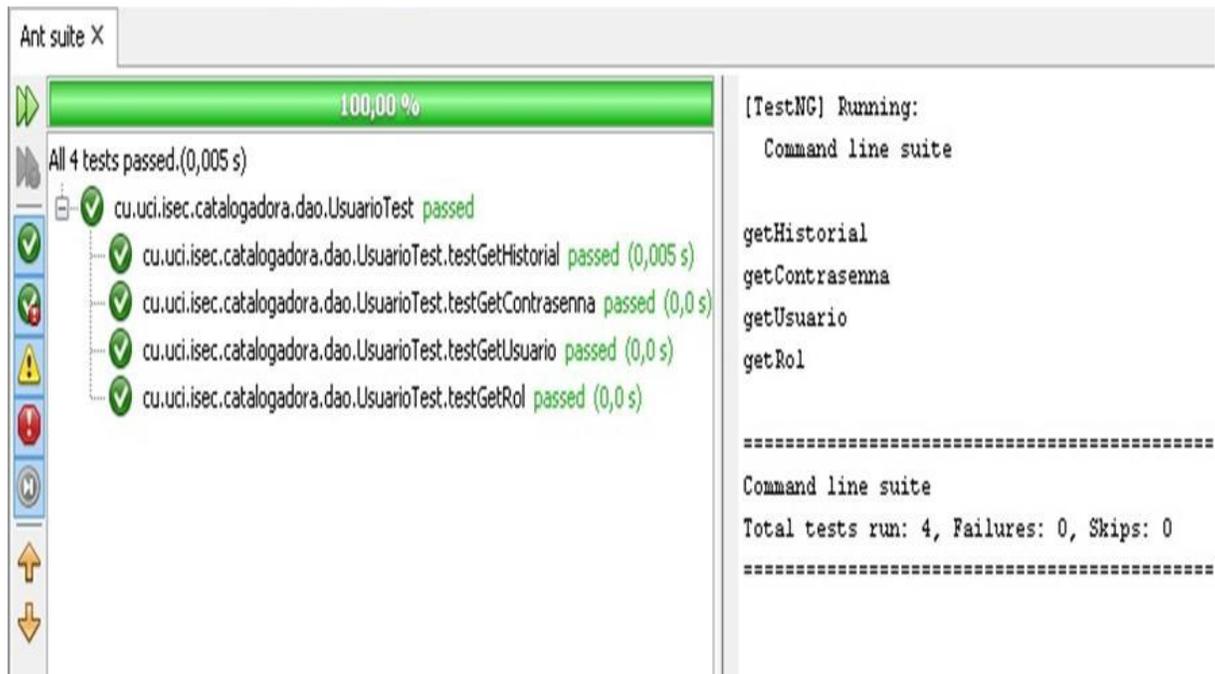


Figura 3. 3 Pruebas Unitarias

### 3.4.2 Prueba de Aceptación

Las pruebas de aceptación se elaboran a lo largo de la iteración, en paralelo con el desarrollo del sistema, y adaptándose a los cambios que el sistema sufra. (Pressman, 2002). El método de prueba seleccionado es el de caja negra el cual se centra en los requisitos funcionales del software. Dentro de este tipo de prueba, se utilizó la técnica de partición de equivalencia.

Las pruebas de caja negra permiten encontrar: (Pressman, 2002)

- ❖ Funciones incorrectas o ausentes.
- ❖ Errores de interfaz.
- ❖ Errores en estructuras de datos o en accesos a las bases de datos externas.
- ❖ Errores de rendimiento.
- ❖ Errores de inicialización y terminación.

Para confeccionar los casos de prueba de caja negra existen distintos criterios; algunos de ellos son: (Pressman, 2002)

- ❖ Técnica de la Partición de Equivalencia: Divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
- ❖ Técnica del Análisis de Valores Límites: Prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
- ❖ Técnica de Grafos de Causa-Efecto: Permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

### Partición de Equivalencia

La partición equivalente es un método de prueba de caja negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Un caso de prueba ideal descubre de forma inmediata una clase de errores, que de otro modo requerirían la ejecución de muchos casos antes de detectar el error genérico. (Pressman, 2002)

### 3.4.3 Diseño de Casos de Prueba

#### Caso de Uso Gestionar Listado de Documentos

##### Descripción general

Caso de uso que inicia cuando el usuario tiene ya cargados los documentos en la interfaz principal selecciona una de las opciones, invertir selección, seleccionar todos los documentos o eliminar documento. El proceso requiere que los documentos estén mostrados para que el usuario pueda ejecutar las opciones que desee.

##### Condiciones de Ejecución

Debe existir el listado de documentos ya cargados y seleccionados todos porque así lo muestra el sistema.

##### Sección a probar CU Gestionar Listado de Documentos

Escenarios de la sección	Descripción	Respuesta del sistema	Flujo central
<b>EC 1.1</b> Invertir documentos seleccionados.	Se invierten los documentos del listado.	El sistema marca o desmarca los documentos del listado.	<ol style="list-style-type: none"> <li>1. Seleccionar los documentos a invertir.</li> <li>2. Oprime el botón "<b>Invertir Documentos</b>".</li> <li>3. Se muestra los documentos que fueron marcados desmarcados y se marcan los documentos desmarcados.</li> </ol>
<b>EC 1.2</b> Eliminar documentos.	Se eliminan los documentos seleccionados.	El sistema elimina los documentos.	<ol style="list-style-type: none"> <li>1. Seleccionar los documentos a eliminar.</li> <li>2. Seleccionar la opción "<b>Eliminar</b>", para eliminar los documentos.</li> </ol>
<b>EC 1.3</b> Seleccionar todos los documentos.	Se selecciona todos los documentos.	El sistema selecciona todos los documentos	<ol style="list-style-type: none"> <li>1. Seleccionar la opción seleccionar todos los documentos.</li> </ol>

**Tabla 3. 1 Diseño de Casos de Prueba CU Gestionar Listado de Documentos**

**Descripción de Variable**

No	Nombre de campo	de	Clasificación	Valor Nulo	Descripción
1	seleccionar		campo de selección	No	Se seleccionan los documentos.

**Tabla 3. 2 Descripción de Variables CU Gestionar Listado de Documentos**

## Caso de Uso Crear Usuario

### Descripción general

El caso de uso inicia cuando un usuario autenticado como Administrador selecciona en la interfaz principal la opción crear usuario.

### Condiciones de Ejecución

Que el usuario autenticado sea administrador del sistema.

#### ❖ Sección a probar Crear Usuario

Escenarios de la sección	Descripción	Respuesta del sistema	Flujo central
<b>EC 1.1</b> Crear usuario	Se crea el usuario exitosamente.	El sistema registra la información referente al usuario creado.	<ol style="list-style-type: none"> <li>1. Selecciona la opción "<b>Crear Usuarios</b>".</li> <li>2. Se muestra el formulario para insertar los datos.</li> <li>3. Introducir los datos y oprime el botón "<b>Aceptar</b>".</li> </ol>

Tabla 3. 3 Diseño de Casos de Prueba CU Crear Usuario

#### ❖ Descripción de Variable

No	Nombre de Campo	Clasificación	Valor Nulo	Descripción
1	Nombre de Usuario	Campo de texto	No	Cadena de caracteres
2	Rol de usuario	Lista desplegable	No	Listado de valores
3	Nueva Contraseña	Campo de texto	No	Cadena de caracteres
4	Confirmar Contraseña	Campo de texto	No	Cadena de caracteres

Tabla 3. 4 Descripción de Variables CU Crear Usuario

## Caso de Uso Gestionar Listado de Usuario

### Descripción general

El caso de uno inicia cuando un usuario autenticado como Administrador selecciona un usuario en la interfaz de listar usuarios y ejecuta las diferentes opciones: modificar, eliminar usuario, ver historial de exportación de metadatos a los diferentes registros bibliográficos o eliminar usuario.

### Condiciones de Ejecución

Deben existir usuarios creados.

#### ❖ Sección a probar CU Gestionar Listado de Usuario

Escenarios de la sección	Descripción	Respuesta del sistema	Flujo central
<b>EC 1.1</b> Buscar Usuarios	Se buscan los usuarios según el criterio de búsqueda seleccionado: - Rol - Usuario	Muestra los usuarios buscados.	<ol style="list-style-type: none"> <li>1. Selecciona la opción "<b>Listar Usuarios</b>".</li> <li>2. Se muestra el formulario listar para realizar la búsqueda.</li> <li>3. Selecciona los criterios de búsqueda por <b>usuario o rol</b>.</li> <li>4. Selecciona la opción "<b>Buscar</b>".</li> </ol>
<b>EC 1.2</b> Modificar Usuario	Se actualizan los datos del usuario exitosamente.	El sistema actualiza los datos del usuario.	<ol style="list-style-type: none"> <li>1. Selecciona la opción "<b>Modificar usuario</b>".</li> <li>2. Se muestra el formulario para modificar el usuario.</li> <li>3. Introducir los datos y oprime el botón "<b>Aceptar</b>".</li> </ol>
<b>EC 1.3</b> Ver Historial	Se obtiene la información de la exportación de metadatos a los diferentes registros bibliográficos de un usuario seleccionado.	Se muestra todo lo referente a las exportaciones realizadas por el usuario seleccionado.	<ol style="list-style-type: none"> <li>1. Se selecciona el usuario.</li> <li>2. Selecciona la opción "<b>Ver Historial</b>".</li> <li>3. Se muestra el historial del usuario seleccionado.</li> </ol>
<b>EC 1.4</b> Eliminar	Se elimina el usuario seleccionado.	El sistema elimina el usuario.	<ol style="list-style-type: none"> <li>1. Selecciona la opción "<b>Eliminar</b>".</li> <li>2. Cierra.</li> </ol>

Tabla 3. 5 Diseño de Casos de Prueba CU Gestionar Listado de Usuario

### 3.4.4 Resultado de Pruebas

#### ❖ No conformidades detectadas

No	No Conformidad	Ubicación	Clasificación	Estado
1	No se muestra un mensaje de error cuando el usuario selecciona la opción invertir documentos y no tiene	EC1.1 CU Gestionar Listado de Documentos.	Validación	Resuelta

	documentos cargados en la interfaz principal.			
2	No se muestra un mensaje de error cuando el usuario no selecciona ningún documento y selecciona la opción eliminar.	EC1.2 CU Gestionar Listado de Documentos.	Validación	Resuelta
3	No se muestra un mensaje de error cuando el usuario no selecciona un usuario en la interfaz de listar usuario y oprime la opción modificar usuario.	EC1.2CUGestionar Listado de Usuario.	Validación	Resuelta
4	No se muestra un mensaje de error cuando el usuario selecciona la opción buscar y no ha escogido ningún criterio de búsqueda	EC1.1CUGestionar Listado de Usuario.	Validación	Resuelta

**Tabla 3. 6 No conformidades detectadas**

### 3.5 Conclusiones

En este capítulo se realizó el diagrama de componente así como el diagrama de despliegue, las pruebas unitarias con JUnit a cada una de las clases implementadas. Además se utilizó el método de caja negra el cual se centra en la validación de los requisitos funcionales del software utilizándose la técnica de partición de equivalencia. Todas las no conformidades detectadas fueron corregidas satisfactoriamente.

## **Conclusiones Generales**

Al culminar el desarrollo del presente trabajo se concluye que:

- ❖ A partir del estudio de las herramientas de extracción de metadatos en documentos digitales y los formatos bibliográficos Marc 21 y Dublin Core, se desarrolló una herramienta informática que facilita la carga masiva de documentos científicos, permitiendo la extracción de sus metadatos y exportación de estos a los formatos Marc21 o Dublin Core.

## **Recomendaciones**

Sobre la presente investigación los autores recomiendan:

- ❖ Incorporarle nuevas funcionalidades a la herramienta basadas en la carga de documentos en diferentes formatos para la extracción masiva de sus metadatos y la exportación a otros formatos bibliográficos.

### Referencia Bibliográfica

1. **José A. Senso, Antonio de la Rosa Piñero.** Scielo (Scientific Electronic Library Online). [En línea] Brasil. [Citado el: junio 10, 2013.] <http://www.scielo.br/pdf/ci/v32n2/17038.pdf/>.
2. **Luis Carlos Alvarez Fernández, Javier Ruiz García. 2012.** *Repositorio Institucional de la Universidad de las Ciencias Informáticas:Actualidad y Proyecciones.* Universidad de las Ciencias Informáticas. Habana : s.n., 2012. científico.
3. **Real Academia Española, RAE. 2001.** *Diccionario de la lengua española. Vigésiam segunda edición.* Real Academia Española. Madrid,España : s.n., 2001.
4. **Congreso, Biblioteca del. 2008.** Datos Bibliograficos Biblioteca del Congreso Oficina de Desarrollo de Redes y Normas Marc. [En línea] 01 08, 2008. [Citado el: abril 27, 2013.] <http://www.loc.gov/marc/bibliographic/bdintro.pdf>.
5. **REBIUM. 1994.** CRUE REBIUM Red de Bibliotecas Universitarias Españolas. [En línea] 1994. [http://www.rebiun.org/opencms/opencms/handle404?exporturi=/export/docReb/informe\\_jordi\\_prats.doc](http://www.rebiun.org/opencms/opencms/handle404?exporturi=/export/docReb/informe_jordi_prats.doc).
6. **Patón, Dr. Eduardo Fernández Medina. 2006-2007.** INGENIERÍA DEL SOFTWARE I Cuarto Curso. [En línea] 2006-2007. <http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema04.pdf>.
7. **Guevara, Jorge Martinez Ladron de.** *Fundamentos de la Programación en Java.* Facultad de Informática Universidad Complutense de Madrid. España : s.n. pp. 1,2.
8. **Jalón, Javier García de, et al. 2000.** *Aprenda Java como si estuviera en primero.* 2000.
9. **Larman, Craig. 1999.** *UML y Patrones Introducción al análisis y diseño orientado a objetos y al proceso unificado.* México : PRENTICE HAL, 1999. Vol. 2da Edición.
10. **Storti, Guillermo, Gladys Ríos and Campodónico, Gabriel . 2007.** [En línea] 2007. [Citado el: mayo 16, 2013.] [http://www.belgrano.esc.edu.ar/matestudio/carpeta\\_de\\_access\\_introduccion.pdf](http://www.belgrano.esc.edu.ar/matestudio/carpeta_de_access_introduccion.pdf).
11. **Tabares, Marta Silvia. 2011.** *Introducción a las Pruebas de Software.* 2011.

## Bibliografía

- Aquilar, Luis Joyanes and Fernández Azuela, Matilde. 2001.** *Java 2 Manual de Programación*. 2001.
- Compendio.** Google Desktop. [En línea] [Citado el: mayo 12, 2013.] <http://googledesktop.com/>.
- Compendio 2004.** Unidad Docente de Ingeniería del Software (UDIS). [En línea] junio 8, 2004. [Citado el: junio 14, 2013.] [http://is.ls.fi.upm.es/docencia/masterTI/ARS/docs/Manual\\_M2C1U11.pdf](http://is.ls.fi.upm.es/docencia/masterTI/ARS/docs/Manual_M2C1U11.pdf).
- Congreso, Biblioteca del. 2008.** Datos Bibliograficos Biblioteca del Congreso Oficina de Desarrollo de Redes y Normas Marc. [En línea] 01 08, 2008. [Citado el: abril 27, 2013.] <http://www.loc.gov/marc/bibliographic/bdintro.pdf>.
- Corporation, Microsoft. 2013.** Windows Search. [En línea] 2013. <http://www.microsoft.com/latam/windows/desktopsearch/search/features.aspx>.
- Documentacion, KDE.** Elementos de UML. [En línea] [Citado el: marzo 2013, 2013.] <http://docs.kde.org/stable/es/kdesdk/umbrello/uml-elements.htm>.
- Erik Hatcher, Otis Gospodnetic. 2005.** *Lucene in action*. Madrid : Leganés, 2005.
- Falgueras, Campderrich Benet. 2003.** *Ingeniería de Software*. s.l. : UOC, 2003.
- Finder, Macintosh. 2010.** Wordiq.com. [En línea] 2010. [http://www.wordiq.com/definition/Macintosh\\_Finder](http://www.wordiq.com/definition/Macintosh_Finder).
- Godoy, Dra. Daniela. 2012.** Minería de Datos Web. [En línea] 2012. [Citado el: enero 17, 2013.] <http://www.exa.unicen.edu.ar/catedras/ageinweb/>.
- Guevara, Jorge Martinez Ladron de.** *Fundamentos de la Programación en Java*. Facultad de Informática Universidad Complutense de Madrid. España : s.n. pp. 1,2. Informática 64. [En línea] [Citado el: enero 25, 2013.] <http://www.informatica64.com/foca.aspx>.
- Informática, Facultad de.** *Patrones del Gang of Four*. Facultad de Informática - Unidad Docente de Ingeniería de Software, Unidad Politécnica de Madrid. España : s.n.
- Jacobson, Ivar, Booch, Grady and Rumbaugh, James. 2000.** *El Proceso Unificado de Desarrollo de Software*. s.l. : Addison Wesley, 2000.
- Jalón, Javier García de, et al. 2000.** *Aprenda Java como si estuviera en primero*. 2000.
- JDOM. 2012.** JDOM. [En línea] noviembre 8, 2012. [Citado el: mayo 23, 2013.] <http://www.jdom.org>.
- José A. Senso, Antonio de la Rosa Piñero.** Scielo (Scientific Electronic Library Online). [En línea] Brasil. [Citado el: junio 10, 2013.] <http://www.scielo.br/pdf/ci/v32n2/17038.pdf/>.

- KRUP. 2000.** *The Rational Unified Process: An Introduction*. s.l. : Addison Wesley, 2000.
- Larman, Craig. 1999.** *UML y Patrones Introducción al análisis y diseño orientado a objetos y al proceso unificado*. México : PRENTICE HAL, 1999. Vol. 2da Edición.
- Luis Carlos Alvarez Fernández, Javier Ruiz García. 2012.** *Repositorio Institucional de la Universidad de las Ciencias Informáticas:Actualidad y Proyecciones*. Universidad de las Ciencias Informáticas. Habana : s.n., 2012. científico.
- Luis Carlos, Eduardo. 2010.** *Implantación de un Repositorio Institucional en la Universidad de las Ciencias Informáticas*. Habana : s.n., 2010.
- Martínez, Jairo Chapela. 2007.** *Introducción al entorno de desarrollo Eclipse*. 2007.
- Maven, Apache. 2012.** The Apache Software Foundation. [En línea] 2012. [Citado el: enero 21, 2013.] <http://tika.apache.org/1.2/formats.html>.
- Mellado, Julio. 2004.** *Estrategia de pruebas de Líneas de Productos de Sistemas de Tiempo Real especificados con diagramas de estado jerárquico*. España : UNIVERSIDAD POLITÉCNICA DE MADRID, 2004.
- NetBeans. 2010.** Welcome to NetBeans. [En línea] 2010. [Citado el: abril 28, 2013.] <http://www.eclipse.org/>.
- Patón, Dr. Eduardo Fernández Medina. 2006-2007.** INGENIERÍA DEL SOFTWARE I Cuarto Curso. [En línea] 2006-2007. <http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema04.pdf>.
- Patricia Testa, Paula Ceriotto. 2008.** *Descripción de Objetos Digitales:Metadatos*. s.l. Universidad Nacional de Cuyo Centro Universitario. 2008.
- Peña, Juan Manuel Fernández. 2010.** Pruebas de Unidad utilizando junit. [En línea] mayo 5, 2010. [Citado el: marzo 8, 2013.] <http://picandocodigo.net/2010/nuevos-proyectos-de-alto-nivel-de-apache/>.
- Pressman, Roger S. 2002.** *Ingeniería de Software. Un enfoque práctico*. Madrid : Mc Graw Hil, 2002.
- Real Academia Española, RAE. 2001.** *Diccionario de la lengua española. Vigésima segunda edición*. Real Academia Española. Madrid,España : s.n., 2001.
- REBIUM. 1994.** CRUE REBIUM Red de Bibliotecas Universitarias Españolas. [En línea] 1994. [http://www.rebiun.org/openccms/openccms/handle404?exporturi=/export/docReb/informe\\_jordi\\_prats.doc](http://www.rebiun.org/openccms/openccms/handle404?exporturi=/export/docReb/informe_jordi_prats.doc).
- Rodríguez, Luz María Hernández. 1999.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. Cámara Nacional de la Industria. . México : Mexicana, 1999. p. 499.

- Rumbaugh, James, Jacobson, Ivar and Booch, Grady. 2000.** *El Lenguaje Unificado de Modelado. Manual de Referencia.* Madrid : Pearson Educación, 2000. Manual .
- Sommerville, Ian. 2005.** *Ingeniería del software.* España : Pearson Educación, 2005.
- Soto, Andrés Ignacio Martínez. 2009.** Seminarios de Formación PolinuX. [En línea] abril 23, 2009. [Citado el: mayo 20, 2013.] <http://www.aplicacionesempresariales.com/sqlite-el-motor-de-base-de-datos-agil-y-robusto.html>.
- Storti, Guillermo, Gladys Ríos and Campodónico, Gabriel . 2007.** [En línea] 2007. [Citado el: mayo 16, 2013.] [http://www.belgrano.esc.edu.ar/matestudio/carpeta\\_de\\_access\\_introduccion.pdf](http://www.belgrano.esc.edu.ar/matestudio/carpeta_de_access_introduccion.pdf).
- Tabares, Marta Silvia. 2011.** *Introducción a las Pruebas de Software.* 2011.
- Universidad. 2010.** Metadatos Recuperación y Acceso a la Información. [En línea] Universidad Carlos III de Madrid, 2010. [Citado el: abril 27, 2013.] <http://www.metadatos-xmlrdf.com/metadatos/dublin-core>.
- University, Carnegie Mellon. 2013.** Software Engineering Institute. [En línea] 2013. [Citado el: junio 11, 2013.] <http://www.iso-architecture.org/ieee-1471/defining-architecture.html>.
- 2011.** Visual Paradigm. [En línea] noviembre 26, 2011. [Citado el: diciembre 16, 2012.] <http://www.visual-paradigm.com>.
- Zealand, Library National of. 2003.** Metadata Extraction Tool. *Metadata Extraction Tool.* [En línea] 2003. [Citado el: diciembre 12, 2012.] <http://www.natlib.govt.nz/services/get-advice/digital-libraries/metadata-extraction-tool>.

## Glosario de Términos

Abreviaturas	Significado
TIC	Tecnologías de la Información y las Comunicaciones
SIGB	Sistema Integrado de Gestión Bibliotecaria
OPAC	(Online Public Access Catalog) Es un catálogo automatizado de acceso público en línea de los materiales de una biblioteca. Generalmente, tanto el personal de la biblioteca como el público tienen acceso a él en varias terminales dentro de la biblioteca o desde el hogar vía Internet.
PAC	Procesamiento analítico sintético de la documentación
UNIX	Unix (registrado oficialmente como UNIX®) es un sistema operativo portable, multitarea y multiusuario.
HTML	HTML, HyperText Markup Language (Lenguaje de Marcado de Hipertexto)
API	API (Interfaz de Programación de Aplicaciones): conjunto de funciones y procedimientos (métodos) que ofrece una biblioteca para ser utilizado por otro software como una capa de abstracción.
JDOM	Java Document Object Model (Documento de Modelado de Objetos en Java)
SQL server	Microsoft SQL Server es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional.
RUP	Rational Unified Process o Proceso Unificado de Desarrollo.
Dicotómica	Pertenciente o relativo a la división entre dos partes.
Información	Es un conjunto ordenado de datos los cuales son manejados según la necesidad del usuario.
MARC21	Es un registro calcográfico legible por máquina (MACHINE- Readable Cataloging) "Legible por máquina" significa que un tipo particular de máquina, una computadora, puede leer e interpretar los datos contenidos en un registro calcográfico
Metadatos	El término «metadatos» no tiene una definición única. Según la definición más difundida de metadatos es que son «datos sobre datos». También hay muchas declaraciones como «informaciones sobre datos», «datos sobre informaciones» e

---

	«informaciones sobre informaciones».
Software Libre	Se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.
XML	Extensible Markup Language (Lenguaje de Marcas Extensible). Es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). XML no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas.
UML	Lenguaje Unificado de Modelado
bytecode	El bytecode es un código intermedio más abstracto que el código máquina.
CASE	Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora
IDEs	Integrated Development Environment o Entorno de Desarrollo

**TablaGT 1. 1 Glosario de Términos**