

**Universidad de las Ciencias Informáticas**

**Facultad 4**



**Análisis y Diseño de un módulo para la gestión de metadatos en la herramienta  
de autor CRODA.**

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

**Autor:** *Mirurgia Avila Rodríguez.*

**Tutores:** *Ing. Dunia María Colomé Cedeño*

*Ing. Lilian Vigoa Machín.*

*Ciudad de La Habana, junio 2011*

*“Año 53 de la Revolución”*

*Declaración de autoría*

Por este medio declaro que soy la única autora de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los \_\_\_\_\_ días del mes de \_\_\_\_\_ del 2011.

Mirurgia Avila Rodríguez

---

Firma de la autora

Ing. Dunia Ma. Colomé Cedeño

---

Firma de la tutora

Ing. Lilian Vigoa Machín

---

Firma de la tutora

*Agradecimientos*

*A Ramiro González por hacer posible la incorporación de este trabajo en la herramienta CRODA, por apoyarme, por ser tan buen compañero de trabajo.*

*A mi tutora Dunia María por haberme dado la oportunidad de desarrollar este tema de tesis, por corregirme y guiarme.*

*A Lilian Vigoa por corregirme en esta investigación.*

*A Osvaldo Estable por toda su cooperación.*

*A mi tribunal y a mi oponente por todas sus recomendaciones.*

*Dedicatoria*

*Este trabajo lo dedico a mi madre Mirurgia por quererme tanto, por ser mi guía y mi razón de ser, por apoyarme en estos cinco años; por creer siempre en mí y por ayudarme a realizar este maravilloso sueño de convertirme en ingeniera.*

*A mi familia por estar siempre apoyándome.*

*A mi eterno amor Reinier Alcayde por amarme y por apoyarme también en estos cinco años.*

*A mi hermana Juliet por quererme como tal.*

*A mi amiga Arleny Acanda, por ser siempre tan leal, hospitalaria, amorosa, siempre dispuesta a ayudar en lo que fuera.*

*A todos mis compañeros aquí en la Universidad.*

## **Resumen**

Las herramientas de autor son actualmente de gran importancia y utilidad dentro del e-learning (aprendizaje electrónico), estas permiten la creación de los objetos de aprendizaje, elementos que luego de su creación se espera que puedan ser localizados y usados a través de su descripción empleando metadatos. Los metadatos son el primer acercamiento que un usuario puede tener con un recurso, conociendo así sus características generales, técnicas, educativas, legales, dependiendo del esquema de metadatos empleado, además, son los que facilitan la realización de búsquedas de Objetos de Aprendizaje en los repositorios donde se almacenan.

En el presente trabajo se realiza el análisis y diseño de un módulo para la gestión de metadatos que fortalezca gradualmente a la herramienta de autor CRODA (**Creando Objetos De Aprendizaje**) en su proceso de creación de Objetos de Aprendizaje y Diseños Instruccionales, favoreciendo la actividad del llenado de metadatos, aumentando la oportunidad de localizar los objetos dentro de un repositorio y contribuyendo a que la descripción de los objetos de aprendizaje y diseños instruccionales creados en la herramienta, se realice con una mayor facilidad y a través de un mismo componente.

En este trabajo se obtiene la fundamentación teórica necesaria para la realización del mismo, para ello se utiliza una bibliografía actualizada de los temas relacionados, se exponen básicamente los requisitos funcionales y no funcionales con los que debe contar el sistema, llegando así a los casos de uso necesarios y su descripción textual, además de desarrollar el análisis y diseño.

## Índice

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA</b> .....	<b>7</b>
<b>1 EL ENTORNO E-LEARNING. FORMACIÓN A DISTANCIA A TRAVÉS DE LAS TIC</b> .....	<b>7</b>
<b>1.1 PLATAFORMAS QUE INTERACTÚAN EN EL E-LEARNING</b> .....	<b>8</b>
<b>1.1.1 HERRAMIENTAS DE AUTOR</b> .....	<b>9</b>
<b>1.2 TECNOLOGÍA OBJETOS DE APRENDIZAJE</b> .....	<b>11</b>
<b>1.3 ESTÁNDARES EN EL E-LEARNING</b> .....	<b>12</b>
<b>1.3.1 ESTÁNDARES PARA EL EMPAQUETAMIENTO DE RECURSOS</b> .....	<b>13</b>
<b>1.3.2 ESTÁNDARES PARA DESCRIBIR RECURSOS</b> .....	<b>15</b>
<b>1.4 LOS METADATOS Y LOS PROCESOS PARA SU CREACIÓN</b> .....	<b>18</b>
<b>1.5 TECNOLOGÍAS Y HERRAMIENTAS PARA EL DESARROLLO DEL MÓDULO</b> .....	<b>21</b>
<b>1.5.1 METODOLOGÍAS DE DESARROLLO DE SOFTWARE</b> .....	<b>21</b>
<b>1.5.2 HERRAMIENTAS CASE</b> .....	<b>25</b>
<b>1.5.3 TECNOLOGÍAS DE IMPLEMENTACIÓN</b> .....	<b>26</b>
<b>1.5.3.1 FRAMEWORK DE DESARROLLO</b> .....	<b>27</b>
<b>1.5.3.2 LENGUAJES DE IMPLEMENTACIÓN</b> .....	<b>28</b>
<b>1.5.3.3 GESTORES DE BASE DE DATOS</b> .....	<b>30</b>

<b>1.5.3.4 SERVIDORES WEB .....</b>	<b>32</b>
<b>CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA. FLUJO DE TRABAJO REQUERIMIENTOS .....</b>	<b>34</b>
<b>2.1 EL MODELO DE DOMINIO O MODELO CONCEPTUAL. NECESIDAD DE SU ELABORACIÓN .....</b>	<b>34</b>
<b>2.1.1 DIAGRAMA DE CLASES DEL MODELO DE DOMINIO .....</b>	<b>35</b>
<b>2.2 REQUERIMIENTOS DEL SISTEMA.....</b>	<b>38</b>
<b>2.3 IDENTIFICACIÓN DE ACTORES DEL SISTEMA.....</b>	<b>44</b>
<b>2.3 IDENTIFICACIÓN DE CASOS DE USO DEL SISTEMA .....</b>	<b>45</b>
<b>2.5 DIAGRAMA DE LOS CASOS DE USO IDENTIFICADOS EN EL SISTEMA.....</b>	<b>46</b>
<b>2.3.4 DESCRIPCIÓN TEXTUAL DE LOS CASOS DE USO IDENTIFICADOS .....</b>	<b>47</b>
<b>CAPÍTULO 3. REPRESENTACIÓN TÉCNICA DEL SISTEMA. FLUJO DE TRABAJO ANÁLISIS Y DISEÑO.</b>	<b>64</b>
<b>3.1 ANÁLISIS .....</b>	<b>64</b>
<b>3.2 ARQUITECTURA PROPUESTA.....</b>	<b>65</b>
<b>3.3 DISEÑO .....</b>	<b>66</b>
<b>CONCLUSIONES GENERALES .....</b>	<b>68</b>
<b>RECOMENDACIONES.....</b>	<b>69</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>70</b>

### Introducción

El avance de la ciencia y la tecnología en la actualidad ha provocado cambios en diversas áreas, sin estar exento de ello el ámbito educativo. Los cambios en esta área están relacionados con la forma en que el estudiante recibe los contenidos, demuestra los conocimientos adquiridos, también en la forma en que los profesores imparten clases o cursos y la manera en que elaboran los mismos. Esta vinculación de las tecnologías y el proceso de enseñanza-aprendizaje ha dado surgimiento al denominado e-learning, el cual se traduce como aprendizaje electrónico e implica dentro de sí todos los métodos, tecnologías y recursos que posibilitan el aprendizaje por cualquier medio electrónico, es una vía de proporcionarle a personas con acceso a las tecnologías la oportunidad de aprender haciendo uso de recursos educativos, sin la necesidad de un profesor.

El e-learning no sustituye el método tradicional de enseñanza, pero si ayuda en gran medida trayendo consigo mayor interacción o participación del estudiante en las actividades propuestas, además de romper las barreras de tiempo y espacio. Con el desarrollo del e-learning surge la necesidad de intercambiar recursos entre plataformas para usar y reutilizar los mismos. Una de las tecnologías que surgieron para darle respuesta a esta problemática son los Objetos de Aprendizaje (OA), elementos con características muy particulares que los diferencian de cualquier otro recurso digital, resaltando entre ellas su durabilidad, accesibilidad e interoperabilidad. Disímiles han sido las definiciones emitidas sobre los OA, ejemplo de ellas las emitidas en (LTSC, 2002), (Guzmán, 2005) y (Chiappe Laverde, y otros, 2007). En este trabajo se toma la siguiente definición: *“Cualquier recurso con una intención formativa, compuesto de uno o varios elementos digitales, descrito con metadatos, que pueda ser utilizado y reutilizado dentro de un entorno e-learning”*. (Guzmán, 2005)

Todas las definiciones sobre los OA, anteriormente citadas y estudiadas en este trabajo, concuerdan en que los OA deben estar descritos a partir de metadatos y que entre sus elementos distintivos está la reutilización. Estos dos componentes se encuentran estrechamente relacionados ya que un factor que propicia la reutilización de un OA es que el mismo esté descrito siguiendo un esquema de metadatos, siendo estos un medio por el cual el profesor puede conocer las características del recurso sin necesidad de acceder al mismo, informándose si el recurso le es útil para sus objetivos, ya sea para el uso

educacional o la reutilización. Es importante tener presente que un OA que no contenga un amplio contenido en la semántica del esquema de metadatos empleado, no será adecuadamente localizado y por consiguiente su uso y reutilización no serán óptimos.

Para la gestión de los OA, varias han sido las herramientas desarrolladas, las cuales permiten su creación y almacenamiento. Entre las más importantes se encuentran los Repositorios de Objetos de Aprendizaje (ROA) y las Herramientas de Autor (HA). Los ROA son creados para almacenar organizadamente OA, permitiendo así su localización, recuperación, catalogación, reutilización, revisión y eliminación. Existen dos tipos de ROAs, primeramente los que almacenan recursos con sus metadatos, donde el objeto y su descripción se encuentran en un mismo sitio, y los que sólo almacenan metadatos y el objeto se referencia en otro repositorio generalmente (Guzmán, 2005). Las HA posibilitan la creación de OA, dependiendo de las características propias de cada una, permiten a los profesores la creación de variadas actividades como ejercicios de diferentes tipos, describir, importar y exportar objetos creados, crear y describir los diseños instruccionales para puntualizar actividades que posteriormente realizarán alumnos y profesores, indicando en qué momento y condiciones, con qué OA y servicios lo harán, entre otras funcionalidades.

Un elemento a tener en cuenta durante la creación y almacenamiento de un OA, para que sea interoperable, son los estándares, un avance de gran importancia dentro del e-learning. Existen actualmente estándares que permiten que los sistemas se comuniquen e intercambien información de forma transparente, estos se denominan estándares de interoperabilidad, ejemplo de ellos son Open Knowledge Initiative (O. K. I.), Instructional Management Systems - Digital Repositories Interoperability (IMS-DRI) y Simple Query Interface (SQI). También se encuentran los estándares de contenidos como Sharable Content Object Reference Model (SCORM) y la especificación *Content Packaging* de Instructional Management System (IMS CP), para empaquetar un OA y los estándares para la descripción de recursos, como Learning Object Metadata (LOM) y Dublin Core. Con el uso de los estándares es posible organizar, localizar, recuperar, intercambiar y reutilizar recursos, es decir, garantizar su interoperabilidad con otros sistemas que manejen los mismos estándares, ya que estos son como patrones, modelos o guías que las instituciones adoptan o crean para realizar diferentes procesos.

Entre los estándares mencionados, resulta de interés para esta investigación LOM por sus beneficios para la descripción de los recursos. Este estándar brinda sesenta y cuatro elementos de información que a pesar de ofrecer cierta riqueza semántica al recurso, puede provocar que la descripción sea tediosa con su uso. La descripción de los recursos es una actividad de relevancia para la reutilización de los mismos, sin embargo en ocasiones los profesores no le prestan la debida atención o desconocen los valores correctos a incorporar en cada uno de los metadatos. La selección correcta del esquema de metadatos y la asignación adecuada de sus valores, dan a los contenidos las propiedades necesarias para potenciarlos como recursos reutilizables, asequibles y durables. (García Peñalvo, y otros, 2006). Es necesario entonces que las HA, faciliten la actividad del llenado de los metadatos, brindando funcionalidades que guíen y ayuden a realizar una correcta gestión de los mismos.

Variadas son las herramientas de autor existentes en el mundo como eXeLearning, RELOAD, AUTORe, Udutu, entre otras, estas emplean distintos esquemas de metadatos para la descripción de los recursos, entre los más empleados se encuentran LOM y Dublin Core. También otras herramientas como los ROA, RHODA<sup>1</sup>, e-prints, JORUM y otros emplean metadatos en la creación de OA y en la realización de búsquedas. Sin embargo en estas herramientas no se tienen en cuenta aspectos que apoyen al profesor en la descripción de los OA, que hagan de esta una actividad interesante y motivadora para los docentes; simplemente permiten añadir metadatos a los recursos creados con una interfaz amigable y con elementos de ayuda en algunos casos, algunas herramientas como Udutu no permiten la descripción de los recursos.

La herramienta de autor CRODA, desarrollada en el Departamento de Producción de Herramientas Educativas del centro Tecnologías para la Formación (FORTES) de la Facultad 4 de la Universidad de las Ciencias Informáticas (UCI), en el proceso de creación de OA ha empleado el estándar LOM para la descripción de los mismos, posibilitando una caracterización amplia de estos, debido a la gran cantidad de elementos de información que brinda, como se ha mencionado anteriormente. Sin embargo en la versión 1.0 de CRODA la actividad relacionada con la gestión de los metadatos presenta varias deficiencias, como:

---

<sup>1</sup> Repositorio de objetos de aprendizaje desarrollado en la Universidad de las ciencias informáticas (UCI).

- No permite definir elementos de obligatoriedad según intereses y características de las instituciones que la utilicen.
- El completamiento de los metadatos por parte del sistema es de vital importancia en su gestión, en CRODA 1.0, solo cinco son autocompletados automáticamente, estos son los siguientes: los elementos *Versión* y *Contribución* de la categoría *Ciclo de Vida*, el elemento *Esquema de metadatos* de la categoría Meta-Metadatos, los elementos *Costo*, *derecho de autor* y *otras restricciones* de la categoría Derechos, pudiendo existir un mayor número de metadatos a autocompletar que facilita dicha actividad a su autor.
- No permite la exportación/importación y publicación de instancias de metadatos, imposibilitando su reutilización.
- No se tiene concebida la gestión de los metadatos como un módulo que pueda ser utilizado para describir, no solamente OA, sino también los diseños instruccionales que se crearán en la herramienta en su versión 2.0.

A partir de lo planteado anteriormente se resume que en CRODA 1.0 no se tienen en cuenta aspectos que apoyen al profesor en la descripción de los OA y de los diseños instruccionales, que hagan de esta una actividad interesante y fácil para los docentes, el profesor necesita elementos de ayuda que lo guíen durante la inserción de metadatos, así como interfaces amigables que satisfagan sus necesidades. Por lo que se hace necesario el incremento y optimización de las funcionalidades referentes a la gestión de los metadatos de la herramienta de autor CRODA.

Referente a lo anteriormente expuesto se plantea el siguiente problema de investigación ¿Cómo favorecer la gestión de metadatos en la herramienta de autor CRODA que contribuya a la creación de OA y diseños instruccionales reutilizables?

Se precisa como objeto de estudio el proceso de gestión de metadatos.

El objetivo general de este trabajo es realizar el análisis y diseño de un módulo para la gestión de metadatos en la herramienta de autor CRODA que contribuya a la creación de OA y diseños instruccionales reutilizables.

El campo de acción lo conforma la gestión de metadatos en la herramienta de autor CRODA.

Se plantea como idea a defender que si se optimizan e incrementan las funcionalidades relacionadas con la gestión de metadatos, en la herramienta de autor CRODA, se favorece la descripción de los OA y los diseños instruccionales, contribuyendo de esta forma a su reutilización.

Para satisfacer el objetivo planteado anteriormente se conforman las siguientes **tareas de investigación**:

- Selección del esquema de metadatos a emplear.
- Investigación sobre el proceso de gestión de metadatos en herramientas de autor existentes e investigaciones al respecto.
- Estudio de roles que intervienen en la gestión de metadatos.
- Estudio sobre las principales dificultades en la gestión de metadatos empleando la herramienta CRODA.
- Selección de las herramientas necesarias para la construcción del módulo.
- Desarrollo de la ingeniería de Requerimientos del módulo para la gestión de metadatos en CRODA, según la metodología seleccionada.
- Desarrollo del Análisis y Diseño del módulo para la gestión de metadatos en CRODA, según la metodología seleccionada.

Los **métodos teóricos** empleados en el transcurso de esta investigación fueron el **Histórico-Lógico** para estudiar el origen y evolución que han tenido los metadatos, así como los estándares y conceptos relacionados con los mismos. El **Analítico – Sintético** para el análisis de la documentación actual relacionada con el tema y las diferentes propuestas y avances existentes en el mundo del e-learning, extrayendo de ello lo necesario para realizar la propuesta del tema en cuestión. También es empleado el método teórico **Modelación** que permitió la definición y descripción de las funcionalidades que conforman el módulo de gestión de metadatos y la modelación de diseños de interfaz, siendo estas actividades la base para la integración del módulo propuesto a la herramienta de autor CRODA.

También se utilizó el método empírico **Observación**, este permitió observar y obtener información sobre cómo se gestionan los metadatos en otras herramientas y proyectos existentes que ayuden a resolver la

problemática planteada. Además, se empleó el método de **Entrevista** (Ver anexo 2), su uso constituyó un medio para adquirir información y una mayor visión de la investigación acerca de este tema.

Este trabajo está conformado por tres capítulos, el **Capítulo 1** aborda de forma concreta algunos de los temas más importantes relacionados con el e-learning. Se hace un estudio de los estándares LOM y SCORM y de los diferentes procesos para la creación de metadatos que han sido propuestos en disímiles proyectos e investigaciones actuales y se observa cómo se realiza el mismo en varias herramientas de autor. Además, se hace un análisis de las herramientas y metodologías utilizadas para el desarrollo del análisis y diseño de dicho módulo, así como para su implementación en CRODA.

En el **Capítulo 2** se hace una descripción de la solución que se propone, realizando primeramente el modelo de dominio para comprender el ámbito donde se desarrolla este módulo. Se desarrolla el flujo de trabajo Requerimientos de la metodología RUP, exponiendo básicamente los requisitos funcionales y no funcionales con los que debe cumplir el sistema, llegando así a los casos de uso necesarios, donde además se realiza una descripción textual de los mismos y se efectúa un diseño de los prototipos de interfaz de usuario de cada caso de uso identificado.

El **Capítulo 3** se enfoca principalmente en el flujo de trabajo Análisis y Diseño, donde a través del desarrollo de los artefactos del análisis se consigue una comprensión más precisa de los requisitos, refinándolos y estructurándolos, utilizando el lenguaje de los desarrolladores para proporcionar una visión general del sistema. También en este capítulo se desarrollan los artefactos del Diseño para crear un plano del modelo de implementación, modelar el sistema y la arquitectura para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen.

## Capítulo I: Fundamentación teórica

### Introducción

En este capítulo se realiza la fundamentación teórica de esta investigación, abordando temas de relevancia para la misma. Se efectúa una selección del esquema de metadatos a emplear, en la solución del problema planteado y se estudian los diferentes procesos para la creación de metadatos que han sido propuestos en disímiles proyectos e investigaciones actuales, además, se observa cómo se realiza el mismo en algunas herramientas de autor, ya que son útiles en el desarrollo de la propuesta de solución. Asimismo, se hace un análisis de las herramientas y metodologías utilizadas para el análisis y diseño del módulo para la gestión de metadatos, así como las necesarias para su implementación en CRODA.

### 1 El entorno e-learning. Formación a distancia a través de las TIC

El e-learning es un sistema que hizo su aparición después de la creación de las redes e internet en la década de los noventa y es el ámbito donde se desarrolla este trabajo. Algunos autores lo definen como *“un conjunto de aplicaciones y servicios orientados a facilitar la enseñanza a través de Internet/Intranet, que facilitan el acceso a la información y la comunicación con otros participantes”* (TTnet, 2005).

Otros plantean que el e-learning es la *“capacitación no presencial que, a través de plataformas tecnológicas, posibilita y flexibiliza el acceso y el tiempo en el proceso de enseñanza-aprendizaje, adecuándolos a las habilidades, necesidades y disponibilidades de cada discente, además de garantizar ambientes de aprendizaje colaborativos mediante el uso de herramientas de comunicación síncrona y asíncrona, potenciando en suma el proceso de gestión basado en competencias”* (García Peñalvo, 2005). Esta es una definición aun más amplia en la que se mencionan términos como plataformas tecnológicas y herramientas de comunicación síncrona y asíncrona. Estos términos se relacionan estrechamente ya que las plataformas del e-learning pueden incluir comunicaciones asíncronas, síncronas o mezclar ambos conceptos.

El e-learning brinda al proceso de enseñanza-aprendizaje una gran variedad de ventajas (Ayuntamiento de Cartagena, 2011):

- **Se alternan diversos métodos de enseñanza:** Los participantes pueden trabajar individualmente o de manera grupal.
- **Permite flexibilidad horaria:** El alumno accede en el momento que dispone de tiempo.
- **Aumenta el número de destinatarios:** Esta modalidad de formación se puede dirigir a una audiencia mucho más amplia.
- **Favorece la interacción:** Los alumnos pueden comunicarse unos con otros, con el tutor y con los recursos on-line disponibles en Internet.
- **Disposición de recursos on-line y multimedia:** Internet proporciona acceso instantáneo e ilimitado a una gran cantidad de recursos, como textos, gráficos, videos y animaciones.

### *1.1 Plataformas que interactúan en el e-learning*

Dentro del e-learning se tienen principalmente dos tipos de plataformas: las que se utilizan para impartir y dar seguimiento administrativo a los cursos en línea y las que se utilizan para la gestión de los contenidos digitales. (Guzmán 2005). Las primeras son conocidas como Sistemas de Gestión de Aprendizaje, LMS por sus siglas en inglés. Los LMS poseen funcionalidades variadas en dependencia de las características propias de cada plataforma, entre las más comunes se encuentran el seguimiento de cursos con evaluaciones, la comunicación a través de chat, foros y otras herramientas con los profesores y entre los propios estudiantes. Ejemplos de estas herramientas son Moodle<sup>2</sup>, Edustance<sup>3</sup> y Claroline<sup>4</sup>.

---

2 <http://www.moodle.org>

3 <http://www.edustance.com/>

4 <http://www.claroline.net>

Por su parte, los sistemas para la gestión de contenidos para el aprendizaje, LCMS por sus siglas en inglés, la integración de herramientas como, las HA para la creación de OA y los ROA, conforman un LCMS.

### 1.1.1 Herramientas de autor

Las herramientas de autor son aplicaciones que disminuyen el esfuerzo a realizar por profesores, maestros, educadores, etc., ofreciéndoles indicios, guías, elementos predefinidos, ayudas y una interfaz amigable para crear materiales educativos y/o cursos en formato digital. Estas se pueden clasificar en tres tipos, las que permiten la creación de materiales educativos digitales, las que pueden generar todos los materiales a incluir en el curso y su publicación y las que generan simulaciones. (Montero, J. L; Herrero, E, 2008). Actualmente existe una diversidad de herramientas de autor que poseen diferentes características en cuanto a funcionalidades, ejemplo de ellas: Reload, eXeLearning y Udu. A continuación se presenta un estudio de cada una de estas herramientas.

#### **Reload (Reusable elearning Object Authoring & Delivery)**

Reload es una herramienta de autor gratuita y de código abierto, que puede editar, pre-visualizar y empaquetar contenidos en dos tipos de estándares (SCORM, IMS). En Reload se encuentran varias herramientas, entre ellas los softwares Reload Editor y Reload SCORM Player, ambas de códigos abiertos y gratuitos. También Reload tiene un empaquetador de contenidos mediante IMS Content Packaging 1.1.4, SCORM 1.2 y SCORM 2004. (Reload, 2008)

Con Reload es posible:

- Probar los contenidos empaquetados con el uso de SCORM player.
- Crear, editar, exportar e importar paquetes de contenidos con el uso de Reload Editor.
- Empaquetar contenidos creados con otras herramientas de autor.
- Entregar contenido a usuarios finales usando la herramienta de guardado previo de contenidos.
- Describir los contenidos con el editor de metadatos de una manera fácil.

En cuanto a la gestión de metadatos esta herramienta cuenta con un editor que posibilita la creación de instancias de metadatos con LOM, este editor tiene una interfaz muy amigable con vista de árbol, vista de

formularios, elementos de ayuda, sin embargo aun es muy complicada para profesores y solamente se puede utilizar para la descripción de los OA. En el caso de la descripción de los diseños instruccionales la herramienta no permite al usuario realizar una descripción empleando el editor, solo posibilita importar una descripción en formato XML, empleando cualquier tipo de estándar.

### **eXeLearning**

EXeLearning es una herramienta de autor de código abierto para la creación de recursos educativos. Es un programa de gran utilidad para los docentes, ya que permite construir contenido web didáctico sin necesidad de ser experto en la edición y marcado con XML ó HTML, permite exportar contenido como páginas web y los recursos pueden exportarse en formatos de paquete de contenido de IMS, SCORM 1.2 (exe, 2008). Esta herramienta tiene una interfaz muy amigable y permite la catalogación de metadatos con el estándar Dublin Core de una manera muy simple con elementos de ayuda y una interfaz basada solamente en formularios. Esta herramienta tiene una desventaja en cuanto a la gestión de metadatos, ya que permite el empaquetamiento con el estándar SCORM 1.2, el cual propone y referencia el uso de LOM como esquema para la descripción, sin embargo como ya se ha dicho anteriormente, eXeLearning permite la descripción de recursos con el esquema Dublin Core. Esto proporciona un problema de interoperabilidad, ya que al reutilizar un recurso empaquetado en eXeLearning con SCORM 1.2, en otra herramienta que utilice el mismo estándar pero con el esquema LOM, no sería posible gestionar los metadatos. Además Dublin Core limita la descripción de los recursos, más adelante se estudiará este esquema de metadatos.

### **Udutu**

Udutu<sup>5</sup> es una herramienta de autor web que permite la creación de cursos completamente multimedia, permite añadir contenidos en flash, en audio, en vídeo, documentos, imágenes, ejercicios de autoevaluación y otros contenidos de manera que se pueda hacer uso de ellos a la hora de la creación de cursos. Una vez creado el curso, Udutu permite exportarlo a formatos como SCORM 1.2 y SCORM 2004, lo cual lo hace compatible con todos los LMS que permiten la importación. Esta herramienta es libre permite el trabajo colaborativo y es multiplataforma, sin embargo no permite la descripción de recursos con el empleo de metadatos, siendo esta su principal desventaja, pues los recursos creados, no tendrán

---

<sup>5</sup> <http://www.udutu.com>

descripción por lo que se vería afectado su uso y reutilización.

Por todas las desventajas de las tres herramientas analizadas se llega a la conclusión que ninguna provee una gestión de metadatos totalmente eficiente. Reload y eXeLearning poseen interfaces amigables basadas en formularios y elementos de ayuda en el llenado de los elementos de información. Siendo estas características en la edición de metadatos un factor de mucha importancia, se pretenden tomar como guía para desarrollar en CRODA 2.0 una interfaz para la edición de metadatos que posea múltiples vistas que interactúen entre sí y al igual que en eXeLearning brindar al usuario un sistema de ayuda para cada elemento de información.

### 1.2 Tecnología Objetos de Aprendizaje

Los OA como ya se define en la introducción de este trabajo surgieron a partir del e-learning con la necesidad de compartir y reutilizar contenidos educativos. El término OA no posee una única definición, ya que ha sido interpretado por diferentes autores en diferentes contextos. La IEEE asocia los OA como *“cualquier entidad, digital o no, susceptible de ser usada en aprendizaje, educación o formación.”* (LTSC, 2002).

La siguiente definición hace referencia a características de gran importancia en los OA como su reutilización, propósito educativo y se refiere a los metadatos como una estructura externa del OA, aunque no siempre se comporta de esta manera, *“Una entidad digital, autocontenible y reutilizable, con un claro propósito educativo, constituido por al menos tres componentes internos editables: contenidos, actividades de aprendizaje y elementos de contextualización. A manera de complemento, los objetos de aprendizaje han de tener una estructura (externa) de información que facilite su identificación, almacenamiento y recuperación: los metadatos”* (Chiappe Laverde, y otros, 2007).

En este trabajo se toma la siguiente definición *“Cualquier recurso con una intención formativa, compuesto de uno o varios elementos digitales, descrito con metadatos, que pueda ser utilizado y reutilizado dentro de un entorno e-learning”*. (Guzmán, 2005)

Las definiciones expuestas aquí y otras estudiadas revelan los desacuerdos y diferencias que existen referentes al término OA. Algunos autores ven a estos como entidades con carácter digital o no, para otros

autores son específicamente recursos electrónicos. Otros los ven como entidades que poseen un contenido y un claro propósito educativo. Otros autores incluyen algunas características de los OA como parte de su definición para hacer énfasis en la importancia que tienen para ellos esas características, como el uso de metadatos o la reutilización.

Precisamente cada una de las características propias de los OA como la reutilización, la interoperabilidad, la granularidad y la descripción a través de metadatos, son las que diferencian a estos de un recurso digital cualquiera y las que los hacen hoy, una tecnología importante y necesaria dentro del e-learning y el proceso de enseñanza aprendizaje .

### *1.3 Estándares en el e-learning*

Un estándar se define como *“un conjunto de elementos, que propone un grupo u organismo reconocido, para describir un determinado dominio o tipo de recursos”* (Berlangua, y otros, 2006). Estos surgieron con la necesidad de convertir las plataformas del e-learning en plataformas interoperables, es decir que pudieran comunicarse e integrarse de una forma efectiva aun siendo plataformas diferentes. Diseñar estándares que puedan ser usados en cualquier entorno, es un reto difícil que solo hace unos pocos años comenzó a desarrollarse, sin embargo existen actualmente organizaciones y proyectos que se han enfrascado en el desarrollo de diferentes trabajos, enfocados en la realización de estándares para aprovechar la gran ventaja que ofrecen las plataformas del e-learning integradas entre sí. A continuación se explican algunos de estos grupos y proyectos según (Guzmán 2005):

#### **IMS Global Consortium Inc.**

Cuenta con miembros de organizaciones comerciales, educativas y gubernamentales dedicadas a definir y distribuir arquitecturas abiertas para actividades de educación en línea. Uno de sus resultados es lo que se conoce como el estándar de empaquetamiento IMS.

#### **Advanced Distributed Learning (ADL), Organización de Aprendizaje Distribuido Avanzado.**

En 1997 el Departamento de Defensa de Estados Unidos y la Oficina de Ciencia y Políticas Tecnológicas de la Casa Blanca lanzan la iniciativa ADL. La misión de ADL es proveer acceso de la más alta calidad en

educación y entrenamiento, en cualquier lugar y en cualquier momento. Para cumplir con estos objetivos crean el modelo SCORM.

**ARIADNE (Alliance of Remote Instructional Authoring and Distribution Networks for Europe)**

Alianza de autoría instruccional remota y redes de distribución para Europa. Es un proyecto de investigación y desarrollo tecnológico de telemática para la educación y el entrenamiento, patrocinado por la Unión Europea. El proyecto se enfoca al desarrollo de herramientas y metodologías para producir, administrar y reutilizar elementos pedagógicos basados en computadora, así como el currículo de entrenamiento a distancia.

**IEEE/LTSC (Institute of Electrical and Electronics Engineers/Learning Technology Standards Committee).**

IEEE es una asociación internacional, cuya misión es promover los procesos ingenieriles para la creación, desarrollo, integración, compartición y aplicación del conocimiento sobre tecnologías electrónicas y de información. Dentro de su organización cuenta con el Comité de Estándares para Tecnología del Aprendizaje (LTSC), que se encarga de desarrollar estándares técnicos, recomendaciones y guías para la tecnología educativa.

Grupos y proyectos, entre ellos ADL, han realizado investigaciones y propuestas sobre la gestión de metadatos. Más adelante en la realización de este trabajo se abordará sobre estos estudios realizados.

*1.3.1 Estándares para el empaquetamiento de recursos*

Los estándares de empaquetamiento son los que permiten intercambiar materiales entre herramientas y sistemas, por ejemplo, una herramienta de autor, un sistema de gestión del aprendizaje o una biblioteca digital de recursos educativos, todas en sí, serían capaces de interpretar los paquetes creados por cada una, independientemente de la forma y el lugar en los que dichos paquetes hayan sido producidos, solamente deben soportar los mismos estándares. De la diversidad de estándares creados para el empaquetamiento por distintas organizaciones, se analizarán en este trabajo las propuestas más empleadas en varias herramientas dentro del campo del e-learning. Estos son los estándares IMS Content Packaging y SCORM. Algunos estándares de empaquetamiento referencian y proponen el uso de

esquemas de metadatos para la descripción de los recursos, dichos esquemas se analizarán más adelante en esta investigación.

### **IMS Content Packaging**

El IMS CP (Instructional Management System Content Packaging) es una variante de estándar para el empaquetamiento de contenidos promovida por IMS Global Learning Consortium. IMS CP describe estructuras de datos que pueden ser usadas para intercambiar información entre sistemas que deseen importar, exportar, agregar, y desagregar paquetes de contenido. La especificación habilita exportar contenido de un LMS o repositorio digital e importarlo hacia otro mientras se retiene la información que describe los medios en el paquete de contenido IMS y su estructura, tales como una tabla de contenidos o cual página HTML mostrar primero, este estándar propone el uso del esquema LOM para la descripción de los recursos. (IMS, 2011)

### **SCORM**

El estándar Sharable Content Object Referente Model (SCORM), tuvo su surgimiento cuando el Departamento de Defensa de Estados Unidos junto al proyecto ADL, desarrollaron un modelo de referencia común para el empaquetamiento de contenidos y su comunicación con los LMS. El modelo SCORM es un conjunto de estándares y especificaciones para compartir, reutilizar, importar y exportar OA. Este modelo describe cómo las unidades de contenidos se relacionan unas con otras a diferentes niveles de granularidad, cómo se comunican los contenidos con el LMS, define cómo empaquetar los contenidos para importarse y exportarse entre plataformas, y describe las reglas que un LMS debe seguir a fin de presentar un aprendizaje específico. (García Peñalvo, y otros, 2006) Dentro del conjunto de estándares que propone SCORM se encuentra LOM para la descripción de los recursos.

SCORM brinda tres libros, el libro del Ambiente de Desempeño (RTE), el Manual de Secuencia y Navegación (SN) y el Modelo de Agregación de Contenidos (CAM), este último en su versión 1.2 propone un modelo de metadatos donde se define qué elementos de información son obligatorios en los metadatos del esquema LOM, usados para marcar los componentes descritos en el CAM. Aunque los distintos libros de SCORM tienen la intención de ser independientes, existen áreas que se entrecruzan o donde hay cobertura mutua. (ADL, 2006)

SCORM es el estándar empleado en CRODA 1.0 para el empaquetamiento de los OA, una de las características de SCORM es que propone cualidades necesarias para lograr OA reutilizables efectivamente, entre estas se encuentra la accesibilidad, que consiste en que *“el contenido de un OA debe ser identificable y ubicable cuando se necesite, para los requerimientos formativos necesarios. Debe poder conocerse su adecuación a los objetivos sin necesidad de obtener el propio contenido mediante la provisión de información suficiente acerca de cada OA.”* (Fernández, y otros., 2007) Esto lo provee el estándar SCORM mediante la propuesta del uso del esquema LOM para la descripción de los OA.

### 1.3.2 Estándares para describir recursos

La necesidad de localizar, reutilizar y clasificar recursos digitales, es la razón por la que distintas organizaciones se han involucrado en procesos de elaboración de esquemas de metadatos para establecer con estos un conjunto de reglas semánticas, sintácticas y de contenido que pretenden describir OA o recursos digitales. Dentro de estos esquemas se encuentra Dublin Core<sup>6</sup>, Canadian Core Learning Resource Metadata Application Profile (Cancore)<sup>7</sup>, LOM-ES<sup>8</sup>, LOM<sup>9</sup> entre otros, seguidamente se estudian los más empleados en el mundo.

#### Dublin Core

Dublin Core es un esquema de metadatos elaborado por la DCMI (Dublin Core Metadata Initiative)<sup>10</sup>. Este esquema está enfocado al ámbito bibliotecario, no es tan amplio como LOM en cuanto a posesión de elementos, puesto que Dublin Core solo posee quince y LOM sesenta y cuatro. Aun así sigue siendo uno de los de mayor difusión en el mundo, por su simplicidad (tan sólo 15 elementos: Contributor, Coverage,

---

<sup>6</sup> Disponible en <http://dublincore.org/>.

<sup>7</sup> Disponible en <http://www.cancore.ca/>.

<sup>8</sup> Una adaptación de LOM creado y utilizado por la comunidad educativa de España.

<sup>9</sup> Para consultar los 64 elementos propuestos por LOM consultar [www.uvs.sld.cu/archivos/lomv1spanish.rar/](http://www.uvs.sld.cu/archivos/lomv1spanish.rar/)

<sup>10</sup> 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.

Creator, Date, Description, Format, Identifier, Language, Publisher, Relation, Rights, Subject, Source, Title y Type), cada elemento es opcional y puede repetirse, además pueden aparecer en cualquier orden. La principal desventaja de este esquema es que al ser aplicado para la descripción de un OA o de cualquier otro recurso, resulta poco descriptivo, la información que brinda no es suficiente.

### **Learning Object Metadata (LOM)**

El IEEE junto al LTSC, desarrolló en el año 2002 el estándar LOM (Learning Object Metadata) para la descripción interoperable de los OA. El propósito de este estándar multi-parte es facilitar la búsqueda, evaluación, adquisición y uso de los objetos educativos, por ejemplo, por alumnos, profesores o procesos automáticos de software. (LTSC, 2002).

Los esquemas para la descripción de recursos, proponen un conjunto de elementos de información, en el caso de LOM, posee sesenta y cuatro elementos de información, un número bastante grande que facilita una amplia descripción del OA. Algunos esquemas reúnen sus elementos relacionados dentro de diferentes grupos, a los que llaman categorías, para una mejor organización. Los elementos de información de LOM se encuentran agrupados en nueve categorías (LTSC, 2002):

- La categoría *General* agrupa la información que describe un objeto educativo de manera global.
- La categoría *Ciclo de Vida* agrupa las características relacionadas con la historia y el estado actual del objeto educativo y aquellas que le han afectado durante su evolución.
- La categoría *Meta-Metadatos* agrupa la información sobre la propia instancia de Metadatos (en lugar del objeto educativo descrito por la instancia de metadatos).
- La categoría *Técnica* agrupa los requerimientos y características técnicas del objeto educativo.
- La categoría *Uso Educativo* agrupa las características educativas y pedagógicas del objeto.
- La categoría *Derechos* agrupa los derechos de propiedad intelectual y las condiciones para el uso del objeto educativo.
- La categoría *Relación* agrupa las características que definen la relación entre el objeto educativo en cuestión y otros objetos educativos relacionados.

- La categoría *Anotación* permite incluir comentarios sobre el uso educativo del objeto e información sobre cuándo y por quién fueron creados dichos comentarios.
- La categoría *Clasificación* describe el objeto educativo en relación a un determinado sistema de clasificación.

LOM se considera el más general de todos los esquemas, ya que abarca una gran cantidad de metadatos, que son reflejados en los demás esquemas existentes, por lo que será el esquema a utilizar debido al gran factor descriptivo que contiene y a que es recomendado por SCORM siendo incluso referenciado por este.

### **Obligatoriedad de elementos en LOM**

Como se mencionó anteriormente, el LTSC proporciona, a grandes rasgos, sesenta y cuatro elementos para los metadatos, muchas investigaciones afirman que esta cantidad de elementos es más de lo que se necesita. Sin embargo el LTSC plantea que todos los elementos de información de LOM son opcionales. Esto implica que al construir una instancia de metadatos en XML, el desarrollador puede escoger y elegir qué elementos utilizar. (ADL, 2006).

En el modelo de metadatos descrito en el CAM de SCORM 1.2, se definen elementos obligatorios para cada componente de un OA (Ver anexo 1). Los elementos de obligatoriedad, son los que se deben describir obligatoriamente para cada componente de un OA, estos se definen para reducir un poco la instancia de metadatos y enfocar las búsquedas en un rango mínimo de elementos y así obtener mayor éxito en los resultados, además posibilita al usuario una guía para saber con qué elementos debe describir un OA en general y con qué elementos debe describir por ejemplo una imagen o ejercicio incluido dentro del OA.

Si no se presentan requisitos en cuanto a qué elementos se deben utilizar al crear instancias de metadatos, podría reducirse la oportunidad de buscar y localizar los objetos dentro de un repositorio u otro tipo de sistema. Al especificar requisitos relacionados con el conjunto de elementos obligatorios en las instancias de metadatos, aumenta la oportunidad de buscar, encontrar y reutilizar los recursos (ADL, 2006). Resultan interesantes las preguntas de qué elementos son importantes en una institución para los profesores o alumnos que ejecutan búsquedas de OA, y qué elementos son necesarios para describir cada componente de los OA individualmente. Por ejemplo una categoría que SCORM 1.2 propone como

opcional y muchas veces es la de más interés por los profesores, es la categoría Uso Educativo. Por todo lo anteriormente analizado se considera importante que las herramientas insistan en que el profesor complete de forma obligatoria algunos elementos de información, que resulten necesarios e importantes en cada caso. De esta forma el profesor tendría un guía para describir los recursos y estos se beneficiarían en cuanto a su descripción.

En la cuarta edición del CAM (ADL, 2009), de la versión 1.3 o también conocida como 2004 de SCORM, no existen requisitos de obligatoriedad para los metadatos, se exige el uso del esquema LOM, pero todos los elementos del mismo son opcionales. Las HA como CRODA deben brindar a las instituciones que la empleen para crear sus contenidos, la posibilidad de establecer los elementos obligatorios, en los que se necesita enfocar la realización de búsquedas en los repositorios donde se almacenarán esos contenidos en determinado momento y establecer como obligatorios los elementos que se consideren necesarios en cada componente de un OA.

Por ejemplo, el repositorio RHODA que se emplea en la UCI y almacena los OA creados en CRODA, una de las categorías por las cuáles permite realizar una búsqueda es por la categoría Uso Educativo, sin embargo esta no es considerada de uso obligatorio en las instancias de metadatos de los OA creados en CRODA, esto disminuye en gran medida la posibilidad de efectuar una búsqueda y obtener resultados satisfactorios empleando los elementos de esta categoría. Igualmente así podría ocurrir con otra categoría u elemento, ya sea con la utilización de CRODA en la UCI o en otra institución.

### *1.4 Los metadatos y los procesos para su creación*

Los metadatos han sido tratados en variadas publicaciones y estudios, en la siguiente definición de metadatos se hace mención de términos de la actualidad tal como *servicios y aplicaciones*, sin embargo los metadatos han tenido su surgimiento desde la antigüedad, cuando los bibliotecarios describían los libros manualmente: *“Los metadatos describen el contenido, la calidad, el formato y otras características de un recurso, constituyendo un mecanismo para caracterizar datos y servicios de forma que usuarios (y aplicaciones) puedan localizarlos y acceder a ellos”*. ( Grupo de Catalogadores de Información Geográfica, 2011). En otras publicaciones se hace referencia a los beneficios que aportan los metadatos (Universidad Carlos III de Madrid, 2011) :

- Adhieren contenido, contexto y estructura a los objetos de información, asistiendo de esta forma al proceso de recuperación de conocimiento desde colecciones de objetos.
- Permiten generar distintos puntos de vista conceptuales para sus usuarios o sistemas, y liberan a estos últimos de tener conocimientos avanzados sobre la existencia o características del objeto que describen.
- Permiten el intercambio de la información sin la necesidad de que implique el intercambio de los propios recursos.
- Son esenciales para sostener un crecimiento de una Web a mayor escala, permitiendo búsquedas, integración y recuperación del conocimiento desde un mayor número de fuentes heterogéneas.

Es el objetivo de la informática como ciencia y una tarea difícil, convertir los procesos que ejecutan los humanos en procesos computacionales, por ejemplo, la descripción a través de metadatos de un recurso es aún una actividad en la que la intervención humana es imprescindible, sin embargo, distintas investigaciones han implementado mecanismos para generar programáticamente algunos elementos de los esquemas de metadatos, aunque existen categorías y elementos en estos esquemas que por sus características pedagógicas requieren de una descripción manual por expertos en la materia. Aun así, algunas propuestas plantean la importancia de que la gestión de los metadatos se automatice (Berlanga, y otros, 2006).

Otras variantes para la descripción de recursos han sido emitidas por varios proyectos y organizaciones, entre ellos JORUM<sup>11</sup>, que es un proyecto del Joint Information Systems Committee (JISC) del Reino Unido cuyo objetivo es crear un ROA, el mismo realizó una investigación encaminada a la elaboración de metadatos para los OA, donde se identifican varios modelos para la creación de metadatos:

- Que el autor cree el metadato.
- Que cree el metadato un especialista en información.
- Que el autor incluya cierta información y el especialista en información otra.

---

11 <http://www.jorum.ac.uk>

- Que cree el metadato otra persona y que, antes de ser publicado, alguien perteneciente al proyecto lo valide (JORUM, 2006).

También el equipo perteneciente al proyecto JORUM y otros autores en (Jorum, y otros, 2006), desarrollan un flujo de trabajo para la creación de metadatos en el que primeramente se autocompletan automáticamente un grupo de elementos de información, luego los contribuidores chequean y mejoran estos elementos automáticos, completan otros y publican el OA, seguidamente los catalogadores, quienes son expertos en descripción de OA, son responsables de generar un grupo de metadatos enfocados en las propiedades educativas del recurso y elementos de difícil automatización, y el último escenario del flujo de trabajo lo complementan los revisores, quienes chequean, revisan y rechazan las instancias de metadatos a una fase anterior del flujo de trabajo si es necesario.

Otros proyectos como ADL, exactamente no propone un flujo para la creación de metadatos, en cambio desarrollaron una guía para la creación de contenidos reutilizables con SCORM 2004 (ADL, 2008), donde se presenta un grupo de roles y responsabilidades para la creación de estos contenidos. Entre los roles propuestos se encuentra el Bibliotecario de Contenidos, específicamente tiene la responsabilidad de crear, mantener y aprobar los registros de metadatos de los assets, SCO (Sharable Content Object), agregaciones, y los paquetes de contenido, define los elementos de información del esquema de metadatos de interés en la organización. Funciona con otros miembros del equipo para asegurar el desarrollo y gestión de metadatos precisos, efectivos y normalizados.

Otra variante para la descripción de recursos es la reutilización de metadatos, se plantea que en el contexto e-learning hablar de reutilización lleva de inmediato al concepto de OA y se asocia siempre a la reutilización de objetos entre plataformas, sin embargo los objetos digitales permiten la reutilización tanto de los recursos como de sus metadatos, este intercambio es posible gracias al uso de estándares de metadatos y a los lenguajes como XML Extensible Markup Language (López Guzmán, y otros, 2008). Una funcionalidad que puede ser empleada para facilitar la actividad del llenado de metadatos, es la reutilización de los mismos. Para observar ejemplos de la codificación en XML del esquema LOM (Ver Anexos 18 y 19).

Son variadas las investigaciones acerca de cómo debe ser la creación de los metadatos, en las analizadas en este trabajo predomina la propuesta de la incorporación de roles que sean especialistas en la

información que intervengan en la gestión de metadatos y la automatización de algunos de los elementos de información, además de la reutilización de metadatos. Todas estas propuestas e investigaciones realizadas se tomarán en cuenta para resolver la problemática planteada en este trabajo.

### *1.5 Tecnologías y herramientas para el desarrollo del módulo*

En el desarrollo del Análisis y Diseño del módulo presentado se adoptarán tecnologías y herramientas. Para la selección de estas se debe realizar un estudio de las más conocidas actualmente analizando sus características y ventajas además de mantener las políticas de desarrollo del centro FORTES, en su actual migración al software libre. Al mismo tiempo como el módulo que se propone será incorporado a la herramienta de autor CRODA en su versión 2.0, se hace una propuesta de las tecnologías de desarrollo necesarias, analizando las empleadas en la herramienta actual y otras de interés, ya que el módulo presentado puede ser incorporado a otras herramientas o sistemas.

#### *1.5.1 Metodologías de desarrollo de software*

La Ingeniería del software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de este después que se utiliza (Sommerville, 2005). La base y guía de la ingeniería del software actual son las metodologías de desarrollo de software, estas abarcan dependiendo de las características propias de cada una, todo un conjunto de actividades necesarias en un orden correcto para controlar un proceso de desarrollo de software desde inicio a fin.

La creación de un software o parte de este, es un proceso riguroso en el mundo de la ingeniería informática por lo difícil que resulta el control de cambios, el trabajo en equipo de forma auténtica, entre otros aspectos que ocasionarían dificultades cómo la pérdida de tiempo, capital, insatisfacción en clientes y desarrolladores o el fracaso total del software. La correcta elección y aplicación de una metodología solucionaría estas dificultades. Actualmente, dentro de las metodologías más conocidas se encuentran Rational Unified Process (RUP) y eXtreme Programming (XP), estas se traducen al español como: Proceso Unificado de Desarrollo y Programación Extrema respectivamente.

## **XP (eXtreme Programming)**

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Las principales características de XP se definen en tres apartados: historias de usuario, roles, proceso y prácticas, las mismas se detallan a continuación: (Calderón, y otros, 2007)

**Historias de Usuario:** Son la técnica utilizada para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales.

### **Roles XP:**

Los roles que propone la metodología XP para el desarrollo de un software son los siguientes:

*Programador:* El programador escribe las pruebas unitarias y produce el código del sistema.

*Cliente:* Escribe las historias de usuario y las pruebas funcionales para validar su implementación.

*Encargado de pruebas (Tester):* Ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente.

*Encargado de seguimiento (Tracker):* Proporciona realimentación al equipo. Verifica el grado de acierto entre las estimaciones realizadas y el tiempo real.

*Entrenador (Coach).* Es responsable del proceso global. Debe proveer guías al equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.

*Consultor:* Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto, en el que puedan surgir problemas.

*Gestor (Big boss):* Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas.

**Proceso XP:** El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega (Release), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto. El ciclo de desarrollo consiste en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.

2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1.

**Prácticas XP:** La principal suposición que se realiza en XP es la posibilidad de disminuir la mítica curva exponencial del costo del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione. Esto se consigue gracias a las tecnologías disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de un grupo de prácticas.

### **RUP (Rational Unified Process)**

RUP es un proceso de desarrollo de software perteneciente al grupo de metodologías pesadas, como cualquier proceso de desarrollo de software define quién hace qué, cómo y cuándo. RUP define cuatro elementos: trabajadores (roles), que responden a la pregunta ¿Quién?, las actividades que responden a la pregunta ¿Cómo?, los artefactos (productos), que responden a la pregunta ¿Qué? y los flujos de trabajo de las disciplinas que responde a la pregunta ¿Cuándo? A continuación se detallan estos elementos (EVA, 2007):

*Trabajadores (“quién”):* Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos.

*Actividades (“cómo”):* Es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.

*Artefactos (“qué”):* Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.

*Flujo de actividades (“Cuándo”):* Secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable.

El ciclo de vida de RUP está centrado en la arquitectura, guiado por casos de uso y es iterativo e incremental. Este ciclo de vida propone nueve flujos de trabajo de los cuales los seis primeros son considerados flujos de ingeniería y los tres restantes como flujos de apoyo. Estos flujos son: Modelado de

Negocio, Requerimientos, Análisis y Diseño, Implementación, Pruebas, Despliegue, Configuración y Administración de Cambios, Administración de Proyecto y Entorno. RUP define cuatro fases en el desarrollo del software:

**Conceptualización (Concepción o Inicio):** Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.

**Elaboración:** Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen.

**Construcción:** Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtienen uno o varios release del producto que han pasado las pruebas.

**Transición:** El release ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores.

### Lenguaje de modelado utilizado por RUP

En la construcción de un software es de gran importancia la creación de modelos que interrelacionen la vida real y el sistema a construir. Los modelos se componen de otros modelos o artefactos, diagramas y documentos que describen cosas. En este aspecto es importante mencionar UML, que no es más que un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos.

Es importante resaltar que UML no es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso, es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos y describe lo que supuestamente hará un sistema, pero no dice cómo implementarlo. Cuenta con estructuras gráficas como: Elementos (Clases, Casos de Uso, etc.), Relaciones (Dependencia, Asociación, etc.) y Diagramas (de comportamiento, de implementación y de estructura estática).

### Análisis de las metodologías de desarrollo de software

Teniendo en cuenta las características del módulo a desarrollar, las particularidades del equipo de trabajo del proyecto y las características propias de las metodologías antes mencionadas, se llega a la conclusión que en el desarrollo de este módulo no se utilizará la metodología XP, debido a que define poca documentación, lo que constituye una desventaja al pensar en el posterior mantenimiento del sistema, el

equipo de proyecto está conformado mayoritariamente por estudiantes, los cuales necesitan consultar una documentación como base, cuando el equipo se disuelva y sea necesario realizar algún cambio o mejora.

Por tanto la metodología más adecuada para dirigir el proceso de desarrollo del software es RUP, a pesar de estar recomendada para proyectos extensos y grandes grupos de trabajo, esta define claramente actividades realizadas por roles generando a su paso artefactos que sustentan el proceso de desarrollo del producto. RUP constituye una metodología robusta y como tal basa su éxito en mantener una documentación ordenada a lo largo de todo el proceso de desarrollo, siendo este último un aspecto importante, teniendo en cuenta que el personal del proyecto es muy cambiante y muchas veces de poca experiencia, por lo que la documentación permitirá que no se pierdan elementos del producto que pueden ser de utilidad para su continuidad. Además el lenguaje utilizado por RUP para la modelación del sistema es UML, el cual brinda amplias posibilidades en la representación, es de fácil uso y conocido por el equipo de proyecto.

### *1.5.2 Herramientas CASE*

Una herramienta CASE es aquella que permite crear los artefactos necesarios siguiendo una metodología en la construcción de un software. En el marco de la metodología RUP, es de mucha importancia en las fases de Análisis y Diseño e Implementación ya que es donde se generan la mayor cantidad de artefactos. A dichos artefactos al ser modelados con estas herramientas, es muy fácil realizarles los cambios que ocurren en el diseño de un software, además de que se hace un software con mucha más calidad y rapidez. A continuación se describen dos de las herramientas CASE más empleadas en el mundo.

#### **Rational Rose Enterprise Edition**

Esta herramienta cubre todo el ciclo de vida de un software, facilita el desarrollo de software en equipo basado en la metodología RUP, cada rol tiene su propia vista de arquitectura (vista de Casos de Uso, vista Lógica, vista de Componentes y vista de Despliegue), es compatible con el lenguaje UML y es uno de los productos más completos de la familia Rational Rose, soporta patrones de Análisis, ANSI C++, Rose J y Visual C++, Enterprise JavaBeans 2.0, e ingeniería directa e inversa para algunas de las construcciones

más comunes de Java, esta herramienta proporciona un lenguaje de modelado común que permite crear más deprisa software de calidad .(IBM, 2009)

Es una herramienta que provee sus funciones de una manera más organizada que Visual Paradigm, los componentes y estereotipos son visualmente más estéticos que los que proporciona Visual Paradigm, es una herramienta muy recomendada y además profesional, pero tiene dos desventajas en su contra, pues obliga al usuario a desarrollar en máquinas con el sistema operativo Windows y a veces resulta difícil el trabajo con ella, ya que hay que efectuar varios pasos para la creación de los artefactos y la incorporación de los componentes.

### **Visual Paradigm para UML**

Visual Paradigm es una herramienta CASE que soporta todo el ciclo de vida del desarrollo de un software: Análisis y Diseño, Construcción, Pruebas y Despliegue. Además permite dibujar todos los diagramas de clases, caso de uso, diagramas de actividades, etc., genera código y documentación desde los diagramas y posibilita el diseño de prototipos de interfaz de usuario. Proporciona además diferentes tutoriales que sirven para un mejor entendimiento de la herramienta.

Permite el diseño de software con el UML, posibilita la captura de requisitos con diagrama de requisitos SysML, brinda la posibilidad de diseñar bases de datos con el diagrama entidad relación, en general proporciona un entorno unificado de diseño de software para el analista de sistemas y desarrollador de software para analizar, diseñar y mantener aplicaciones de software en una disciplina (visualparadigm, 2009). Es una herramienta que está disponible en múltiples sistemas operativos como Windows, Linux y Unix y es muy fácil su uso, siendo estas sus principales ventajas. Teniendo en cuenta las ventajas y desventajas de estas herramientas analizadas se decide la selección de Visual Paradigm como herramienta CASE en este trabajo.

### *1.5.3 Tecnologías de implementación*

Específicamente para la implementación del módulo presentado se realiza una propuesta de las tecnologías y herramientas necesarias, tales como el framework de desarrollo, los lenguajes y otras

tecnologías relacionadas con el código a implementar. Para la selección de estas se realiza un estudio de las más conocidas actualmente.

### 1.5.3.1 Framework de desarrollo

Los marcos de trabajo o frameworks de desarrollo surgieron por la necesidad de facilitar y optimizar la creación de aplicaciones web. Dentro de los frameworks más usados actualmente se encuentran Symfony y Zend Framework. A continuación se describen las características de estos.

#### **Zend framework**

Se trata de un framework para el desarrollo de aplicaciones y servicios Web con PHP, brinda soluciones para construir sitios web modernos, robustos y seguros. Además implementa el patrón MVC, es orientado a objetos y sus componentes tienen un bajo acoplamiento por lo que se puede usar en forma independiente. También este framework brinda un estándar de codificación. (Zend, 2011)

#### **Symfony**

Según (Fabien Potencier, 2008), Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Se diseñó para que se ajustara a los siguientes requisitos:

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y \*nix estándares).
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

Symfony tiene numerosas características como lo son su publicación bajo una licencia de software libre, la amplia documentación existente, su creación sólo para PHP 5, para obtener el máximo rendimiento de

PHP y aprovechar todas sus características. Además la herramienta de autor CRODA está desarrollada sobre este framework por lo que continuar su desarrollo empleando Symfony mantendrá una mejor organización y no romperá con la arquitectura, por lo tanto se propone como framework para el futuro desarrollo del módulo.

### 1.5.3.2 Lenguajes de implementación

#### **XML**

Se propone el uso del lenguaje XML, ya que los metadatos LOM se codifican empleando este lenguaje, el mismo se define como: *“un conjunto de reglas que se usan para definir etiquetas semánticas las cuales organizan un documento en diferentes partes. Siendo así un meta lenguaje que define sintaxis para definir otros lenguajes etiquetados estructurados”* (W3C, 2008). Su principal característica es que no posee etiquetas definidas desde un principio, por lo que el propio autor las define como desee. La codificación en XML del LOM es, fundamentalmente, un grupo de reglas que describen cómo crear instancias de metadatos LOM en XML.

#### **HTML (HyperText Markup Language)**

Es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. Además, puede describir, hasta un cierto punto, la apariencia de un documento. Es un lenguaje de composición de documentos y especificación de ligas de hipertexto que define la sintaxis y coloca instrucciones especiales que no muestra el navegador, aunque sí le indica cómo desplegar el contenido del documento, incluyendo texto, imágenes y otros medios soportados. Este lenguaje se propone para la construcción de las vistas del módulo presentado. (desarrolloweb, 2011)

#### **CSS (Cascading Style Sheets)**

CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML casi siempre. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

Algunas ventajas de utilizar CSS son:

- Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.
- Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o, incluso, a elección del usuario.
- El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño. (desarrolloweb, 2011)

### JavaScript

JavaScript es un lenguaje de programación del lado del cliente ampliamente utilizado en el mundo del desarrollo web por ser muy versátil, compatible con la mayoría de los navegadores modernos, además de ser muy potente, tanto para la realización de pequeñas tareas, como para la gestión de complejas aplicaciones. Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas. (Álvarez, 2008)  
En este trabajo se propone el uso de la librería de java script ExtJS.

### ExtJs

ExtJs es una librería desarrollada en JavaScript que permite el desarrollo de aplicaciones web interactivas empleando tecnologías como Ajax, DHTML y DOM. ExtJs proporciona una interfaz de usuario enriquecida, muy parecida a las que se encuentran en las aplicaciones de escritorio. Esto permite a los desarrolladores web concentrarse en las funcionalidades de las aplicaciones y no en los detalles técnicos. Puede trabajar en conjunto con otras librerías si se definen adaptadores para las mismas, incluye componentes de interfaz de usuario del alto performance y personalizables, cuenta con un modelo de componentes extensibles, un API fácil de usar y licencias Open Source (GPL) y comerciales. El framework de ExtJS cuenta con un conjunto de componentes para incluir dentro de una aplicación web, como: cuadros y áreas de texto, radiobuttons y checkboxes, editor HTML, árbol de datos, pestañas, paneles divisibles en secciones etc. (Sencha, 2006)

Uno de los objetivos del diseño de este módulo es resolver los problemas que presenta la herramienta con la interfaz de usuario en la creación de metadatos, es necesario el diseño de una interfaz de usuario amigable que satisfaga las necesidades de los usuarios y que sea compatible con diferentes navegadores, por todo lo anterior, por las múltiples propiedades y eventos que posee ExtJs y por haber sido empleada en la creación de toda la interfaz de usuario de CRODA, se propone entonces para la futura incorporación de este módulo a la misma.

### **PHP**

Como lenguaje de desarrollo se propone PHP ya que este lenguaje es multiplataforma, está orientado al desarrollo de aplicaciones web manteniendo acceso a información almacenada en una base de datos, el código fuente escrito en PHP es transparente al navegador y al usuario haciendo la programación segura y confiable. Presenta capacidad de conexión con la mayoría de los motores de base de datos destacándose principalmente en MySQL y PostgreSQL y es el empleado actualmente por el equipo de CRODA.

#### *1.5.3.3 Gestores de base de datos*

Un sistema gestor de base de datos (SGBD) es un software que sirve de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan, estos permiten manejar de forma clara sencilla y ordenada la construcción y manipulación de una base de datos para diversas aplicaciones. Entre los más populares y empleados mundialmente se encuentran Oracle y PostgreSQL.

### **Oracle**

El SGBD Oracle, fabricado por Oracle Corporation, utiliza la arquitectura cliente/servidor. Ha incorporado en su sistema el modelo objeto-relacional, pero al mismo tiempo garantiza la compatibilidad con el tradicional modelo relacional de datos, así ofrece un servidor de bases de datos híbrido. Es uno de los más conocidos y ha alcanzado un buen nivel de madurez y de profesionalidad. Es robusto y muy completo, cuenta con características como su estabilidad y escalabilidad, además de ser multiplataforma. (Oracle Corporation, 2011) Sin embargo Oracle no cuenta con una licencia libre, la gran potencia que

tiene y su elevado precio hacen que sólo se utilice en empresas muy grandes y multinacionales. Todas estas desventajas llevan a Oracle a no ser muy extendido como otros sistemas gestores de base de datos y por tanto no se propone para el desarrollo futuro de este módulo.

### **PostgreSQL**

PostgreSQL es un potente servidor de base de datos relacional orientado a objetos de uso libre y de código abierto (open source), incluye extensa documentación, incluyendo un enorme manual y algunos ejemplos de ensayo, se ejecuta en la mayoría de los sistemas operativos más utilizados en el mundo incluyendo Linux, varias versiones de UNIX y en Windows. A continuación otras características del gestor PostgreSQL (Postgresql, 2007): cuenta con comunidades muy activas y varias en español, es altamente adaptable a las necesidades del cliente, presenta un soporte nativo para los lenguajes más populares del medio: PHP, C, Perl, Python, soporta todas las características de una base de datos profesional (triggers, store procedures, funciones, secuencias, relaciones, reglas, tipos de datos definidos por usuarios). Además permite realizar transacciones, subselects, disparadores, vistas, claves foráneas. Por todas las características y numerosas ventajas de PostgreSQL, se propone como gestor de base de datos a utilizar en la incorporación del módulo a CRODA.

### **eXist-db**

Debido a que los metadatos están expresados en ficheros XML llevarlos a una base de datos relacional traería varios inconvenientes como garantizar que se mantenga la integridad entre el esquema y la estructura física del XML, por lo que se propone para la futura implementación del módulo el empleo del gestor de base de datos xml eXist-db. El mismo posibilita guardar un XML en su estructura original y realizarle consultas para la obtención de información o modificación de su contenido.

Exist-db es un SGBD de código abierto construido usando tecnología XML. Almacena datos en XML de acuerdo al modelo de dato de ese mismo formato. Soporta múltiples estándares de tecnologías (web principalmente) haciéndolo una excelente plataforma para el desarrollo de aplicaciones basadas en la web (Exist, 2008). A continuación se mencionan algunos de estos estándares soportados por Exist-db:

- Xquery 1.0 / Xpath 2.0 / XSLT 1.0 ó XSLT 2.0.

- Interfaces HTTP: REST, WebDav, SOAP, XMLRPC, Atom Publishing Protocol.
- Base de datos XML: XMLDB, XUpdate, XQuery.

### 1.5.3.4 Servidores web

Los servidores web son grandes proveedores de información para todo tipo de usuarios. Estos surgieron con motivo de la necesidad que tenían o requerían algunas empresas de compartir información con algún grupo de clientes. Entre los más conocidos principalmente en el mundo del software libre se encuentran Apache y Roxen.

#### **Roxen**

Es un servidor web de licencia GNU, desarrollado en el lenguaje "Pike". Sus características más destacables son (Cibernetia, 2011):

- Multiplataforma: puede ejecutarse en Windows, Linux, MAC OS/X, Solaris, etc.
- Código libre.
- Interfaz de administración basada en web, completa y fácil de usar.
- Soporte para gráficos integrado que posibilita, mediante etiquetas de RXML (la extensión de HTML propia de Roxen), la generación de imágenes, títulos, gráficos, etc.
- Acceso integrado a bases de datos, que permite el acceso a PostgreSQL, Oracle, MySQL, etc.
- MySQL integrada.
- Programación de servidor con PHP, RXML, Java, Perl y CGI.
- Soporte criptográfico fuerte.

Roxen es uno de los pocos casos en los que un excelente producto (ha sido siempre uno de los servidores web más estables, rápidos y con mayor número de prestaciones y facilidades) no ha triunfado, ya que siempre ha sido eclipsado por Apache.

### **Apache**

Apache es un servidor web robusto cuya implementación se realiza de forma colaborativa. Entre sus principales características se encuentra el spelling, la cual es una prestación que permite definir una página de error para los recursos no encontrados que sugiera al usuario algunos nombres de recurso parecidos al que solicitaba para el caso de que hubiese cometido un error al escribir. El status proporciona una página web generada por el servidor donde éste muestra su estado de funcionamiento, nivel de respuesta, etc. Además Apache es modular, de código abierto, multi-plataforma, Popular (fácil conseguir ayuda/soporte) (Apache Software Foundation, 2011). Apache es el servidor web que se propone a utilizar en el desarrollo de este módulo, por las características antes mencionadas y por haber sido empleado en el desarrollo de CRODA 1.0.

### **Conclusiones**

En este capítulo a través del análisis bibliográfico y el estudio realizado se expusieron los principales conceptos y tecnologías en el e-learning, como la tecnología OA y el estándar para describirlos, se seleccionó el esquema de metadatos LOM para emplearlo en la propuesta de solución, así como los procedimientos de creación de metadatos a tomar en cuenta.

Quedaron seleccionadas además la metodología de desarrollo de software RUP y la herramienta CASE Visual Paradigm, ya que por sus características estudiadas, resultaron las apropiadas para el desarrollo del módulo, además se realizó una propuesta de las tecnologías y herramientas necesarias para su futura implementación.

## *Capítulo 2: Características del sistema. Flujo de trabajo Requerimientos*

### **Introducción**

Este capítulo constituye la propuesta de solución para favorecer la gestión de metadatos en la herramienta de autor CRODA 1.0. Para la descripción de la solución se desarrolla el modelo de dominio, describiendo las clases y relaciones que lo conforman, en aras de lograr la utilización de un vocabulario común. Se realiza la especificación de los requerimientos funcionales y no funcionales que debe presentar el sistema. Además, se identifican y describen los casos de uso del sistema y sus actores. De manera general el capítulo recoge entre otros aspectos el flujo de trabajo Requerimientos de la metodología RUP analizada en el capítulo anterior.

### *2.1 El modelo de dominio o modelo conceptual. Necesidad de su elaboración*

El flujo de trabajo Modelamiento del Negocio determina el funcionamiento del negocio. Es un modelo real de las personas grupos o entidades que se benefician de la actividad desarrollada en el negocio (actores del negocio) y la interacción con las personas que realizan estas actividades (trabajadores del negocio). En este flujo se describen las entidades, actividades y procesos (casos de uso del negocio) que se realizan, para entender las implicaciones y ámbito del negocio, que sirva de guía en la realización de los restantes flujos de trabajo.

En el estudio realizado del problema planteado para este trabajo, no fue posible identificar procesos del negocio, ya que está altamente centrado en tecnologías informáticas, pues el objetivo de este módulo es la gestión de la descripción estandarizada de recursos digitales, haciéndose difícil determinar actores del negocio y procesos de negocio. Por lo que se hace necesaria la realización del modelo de dominio.

El modelo de dominio o modelo conceptual es una representación visual estática del entorno real donde se desarrollará el sistema. Representa un diagrama con los objetos que existen (reales) relacionados con el proyecto que se va a acometer y las relaciones que hay entre ellos. Estos objetos no son clases de software (aunque algunos objetos del Modelo de Dominio pueden terminar siéndolo). (EVA, 2008) Cada objeto debe poseer un nombre representativo que contribuya a utilizar un lenguaje común que facilite la



## *Capítulo 2: Características del sistema. Flujo de trabajo Requerimientos.*

**CRODA:** Herramienta de autor web cuyo objetivo principal es permitir la creación de OA, reutilizables, accesibles, duraderos e interoperables, de forma flexible, empleando los estándares SCORM y LOM.

**OA:** Recurso digital dirigido a la formación, fácilmente identificado y encontrado a través de sus metadatos, con capacidades para integrarse en sistemas e-learning diferentes, para ser usados con distintos propósitos educativos y para combinarse con otros objetos formando una sola entidad.

**Diseño instruccional:** Diseño que refleja cómo debe desarrollarse el aprendizaje de una actividad educativa, definiendo qué se pretende, a quién estará dirigida, qué recursos y actividades serán necesarios, cómo se evaluará, entre otros aspectos.

**Metadatos de diseño instruccional:** Conjunto de elementos de información necesarios para describir cada componente del diseño instruccional.

**SCORM:** Conjunto de estándares y especificaciones para compartir, reutilizar, importar y exportar OA.

**Agregación de Contenido:** Componente del modelo de contenido de SCORM, empleado para identificar y describir el paquete de contenido en sí, representa los elementos de un OA como un todo.

**Metadatos de Agregación:** Conjunto de elementos de información necesarios para describir el paquete de contenido en su conjunto.

**Assets:** Un asset es un componente del modelo de contenido de SCORM, es la forma más básica de representar o construir recursos para el aprendizaje, es una representación electrónica de medios, texto, imágenes, sonidos, páginas web, objetos de evaluación u otros recursos.

**Metadatos de Assets:** Los metadatos de un asset son un conjunto de elementos necesarios que describen a los assets independientemente de su contexto.

**SCOs (sharable content object):** Objeto de contenido compartible, es un componente del modelo de contenido de SCORM, constituye una recopilación de uno o más assets que representa un recurso de manera individual.

**Metadatos de SCOs:** Los metadatos de un SCO son un conjunto de elementos de información necesarios que describen a un SCO independientemente de su contexto.

**Organizaciones de contenido:** Componentes del modelo de contenido de SCORM que representan un conjunto de actividades relacionadas entre sí.

**Metadatos de organización:** Los metadatos de una organización son un conjunto de elementos de información necesarios que describen una organización independientemente de su contexto.

### **Descripción del diagrama de clases del modelo de dominio**

El autor es un rol autenticado en CRODA que utiliza la misma para elaborar uno o más OA, a estos OA a su vez, si el autor lo considera puede asociarle uno o varios diseños del aprendizaje, creados por los diseñadores instruccionales. El diseñador instruccional también es un rol autenticado en CRODA que utiliza la misma para la creación de estos diseños que son además, descritos con metadatos. CRODA genera uno o más OA empaquetados siguiendo el estándar SCORM para posibilitar el intercambio de los contenidos entre sistemas, este estándar contiene en su modelo de contenidos un conjunto de componentes, la agregación de contenido compuesta por Asset, SCOs y organizaciones de contenido, cada uno de estos componentes se describen individualmente con metadatos con el objetivo de facilitar su búsqueda y utilización.

Como se puede apreciar en el diagrama, la descripción del OA y de los diseños instruccionales a partir de los metadatos es realizada por los autores y diseñadores instruccionales respectivamente, por lo cual las personas que asumen estos roles, tienen dos actividades principales. Las actividades que se enfocan en la creación del OA y del diseño de aprendizaje, las cuales implican tener tiempo y conocimiento pedagógico, y la actividad de describir estos OA y diseños del aprendizaje, la cual requiere de la intervención de expertos en la catalogación, para obtener una mejor descripción. En CRODA 1.0 no existe un rol que se encargue específicamente de la catalogación de los recursos, que conjuntamente con los autores y diseñadores instruccionales asegure el desarrollo y gestión de metadatos precisos y efectivos.

Además, se aprecia en el diagrama que cada componente de un OA y los diseños instruccionales deben describirse con metadatos en cada caso, sin embargo no existe la posibilidad de puntualizar la

obligatoriedad de elementos de información para cada uno de ellos según intereses de la organización donde se emplee la herramienta CRODA.

## *2.2 Requerimientos del sistema*

La ingeniería de requerimientos es un proceso extenso y arduo ya que desarrollar un sistema o parte de este, cambia el entorno y las relaciones entre las personas, así que es importante identificar a todos los implicados, considerar sus necesidades y asegurar que entiendan las implicaciones de los cambios a realizar en el sistema. Se pueden emplear varias técnicas para obtener los requerimientos del cliente, tales como las tormentas de ideas, casos de uso, comparación de terminología, estudio de otros sistemas y entrevistas. La entrevista realizada en este trabajo (Ver Anexo 2) ha servido tanto para la realización del marco teórico así como para identificar las necesidades de los profesores que interactúan con CRODA 1.0 y además, para obtener ideas que beneficiaron la modelación de las funcionalidades del módulo.

### **Requisitos funcionales del sistema**

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

**R 1 Insertar metadatos en forma de formulario:** el sistema debe permitir al usuario insertar los elementos de información en forma de formulario.

**R 2 Insertar metadatos en forma de árbol:** el sistema debe permitir al usuario insertar elementos de información en forma de árbol.

**R 3 Permitir interactividad entre las vistas de árbol y formulario:** el sistema debe ser capaz de reflejar en la vista de árbol todo lo que el autor modifique en la vista de formularios y viceversa.

**R 4 Guardar cambios efectuados en los metadatos:** el sistema debe brindar la posibilidad al usuario de guardar los cambios efectuados en los metadatos.

**R 5 Deshacer cambios efectuados en los metadatos:** el sistema debe brindar la posibilidad al usuario de deshacer los cambios efectuados en los metadatos, borrando los cambios introducidos por el usuario que no hayan sido guardados.

**R 6 Autocompletar metadatos de forma automática en la agregación de contenidos del OA:** el sistema debe ser capaz de autocompletar los siguientes metadatos de manera automática. De dicho autocompletamiento no podrán ser modificados los siguientes elementos: Identificador de la Categoría General, Tipo, Fecha y Versión de la Categoría Ciclo de Vida, Esquema de metadatos, Fecha e identificador de la categoría Meta-metadatos y Tamaño de la categoría Técnica.

Pertenece a la categoría General:

- Estructura: Jerárquica.
- Identificador: IDE-MD5 (time ()).

Pertenece a la categoría Ciclo de Vida.

- Contribución.
- Tipo: Autor y Creador.
- Fecha.
- Estado: Borrador.
- Versión: 1.0.
- Entidad: correo electrónico del Autor y el creador.

Pertenece a la categoría Meta-Metadatos:

- Contribución (Tipo): Autor por defecto. Si el usuario solicita una edición al bibliotecario de contenido, en el elemento Tipo debe aparecer el rol Bibliotecario de contenido.
- Esquema de metadatos: LOM V1.0
- Fecha
- Identificador

Pertenece a la categoría Técnica:

- Formato: para el OA será Aplicación ZIP.
- Requisitos (Tipo): Sistema operativo.
- Nombre: Multi-SO.
- Tamaño.

## *Capítulo 2: Características del sistema. Flujo de trabajo Requerimientos.*

Pertenece a la categoría Uso educativo:

-Destinatario: por defecto tendrá el valor Aprendiz.

**R 7 Mostrar todos los metadatos:** El sistema debe mostrar al usuario todos los elementos del esquema LOM, incluyendo los obligatorios, opcionales y reservados en cada caso que corresponda y diferenciándolos entre sí.

**R 8 Mostrar ayuda para especificar los elementos de información:** el sistema debe ser capaz de mostrar al usuario una ayuda sobre los elementos de información que brinda el esquema de metadatos LOM. Al lado de cada elemento de información aparecerá un ícono, cada vez que el usuario de un clic sobre el mismo, el sistema debe mostrar el significado del elemento y un pequeño ejemplo.

**R 9 Exportar archivo XML de los metadatos:** el sistema debe brindar al usuario la posibilidad de exportar los metadatos en un archivo XML, a una dirección especificada.

**R 10 Importar archivo XML de los metadatos:** el sistema debe brindar al usuario la posibilidad de importar los metadatos expresados en un archivo XML, desde una dirección especificada.

**R 11 Publicar metadatos:** el sistema debe posibilitar al usuario publicar una instancia de metadatos, para que pueda ser visualizada y utilizada por otros usuarios.

**R 12 Listar metadatos publicados:** el sistema debe posibilitar al usuario listar todas las publicaciones de metadatos que han sido realizadas

**R 13 Visualizar metadatos publicados:** el sistema debe permitir al usuario visualizar cada instancia de metadatos que ha sido publicada, mostrando cada elemento de información con su valor correspondiente.

**R 14 Reutilizar metadatos:** el sistema debe permitir al usuario reutilizar cada instancia de metadatos que ha sido publicada, cargándola en la instancia que está siendo editada por el usuario.

**R 15 Buscar metadatos:** el sistema debe permitir al usuario realizar una búsqueda de metadatos, según el tipo de metadatos que sea (metadatos de OA, metadatos de DI) y/o los elementos de información que el usuario desee encontrar.

**R 16 Solicitar al bibliotecario de contenido la edición de los metadatos:** el sistema debe permitir al autor y al diseñador instruccional solicitar al bibliotecario de contenido la edición de los metadatos de los OA o del diseño instruccional.

**R 17 Establecer valores predeterminados a elementos de información:** el sistema debe permitir al bibliotecario de contenidos establecer valores predeterminados para los siguientes metadatos.

- Idioma: Categoría General.
- Ámbito: Categoría General.
- Idioma: Categoría Meta-metadatos.
- Costo: Categoría Derechos.
- Derecho de autor y otras restricciones: Categoría Derechos.
- Descripción: Categoría Derechos.
- Localización: Categoría Técnica.

**R 18 Establecer metadatos obligatorios para los OA creados con SCORM 1.2:** el sistema debe permitir al bibliotecario de contenido definir además de los establecidos por SCORM 1.2, los elementos y/o categorías que no pueden dejar de describirse en cada componente de los OA creados con SCORM 1.2, porque han sido metadatos definidos como importantes en la organización. Los metadatos obligatorios establecidos por SCORM 1.2 aparecerán ya marcados y el bibliotecario de contenidos no podrá desmarcarlos.

**R 19 Visualizar la lista de solicitudes enviadas para la edición de metadatos de OA y DI:** el sistema debe ser capaz de mostrar a los bibliotecarios de contenido una lista de solicitudes enviadas para la edición de metadatos de OA y DI realizados por los autores y diseñadores instruccionales.

**R 20 Visualizar el contenido y estructura del OA:** el sistema debe posibilitar a los bibliotecarios de contenido visualizar el contenido y estructura del OA al cuál le editarán los metadatos.

**R 21 Terminar edición:** el sistema debe permitir al bibliotecario de contenidos terminar la edición de los metadatos, automáticamente el sistema debe enviar un mensaje de notificación a los autores del OA informándole que ya se terminó la edición. El sistema elimina el OA o el DI de la lista del bibliotecario de contenido.

**R22 Visualizar diseño instruccional:** el sistema debe permitir al bibliotecario de contenido visualizar el diseño instruccional al cual le está editando los metadatos.

**R 23 Enviar mensaje de aviso por vencimiento de la edición:** un bibliotecario de contenido podrá tener un OA o un diseño instruccional en su lista solamente por 7 días. Al cumplirse este plazo, el sistema coloca el mismo nuevamente en la lista general de envíos y envía un mensaje de notificación al bibliotecario de contenido.

**R 24 Mostrar mensaje de aviso por exceder la cantidad máxima de metadatos en edición:** el sistema debe ser capaz de mostrar un mensaje de aviso al bibliotecario de contenido si este excede la cantidad máxima de solicitudes que puede atender para editar los metadatos, propuesta por el administrador.

**R 25 Visualizar metadatos del OA:** el sistema debe permitir a los bibliotecarios de contenido visualizar todos los metadatos del OA al cual le están realizando la edición de los metadatos.

**R 26 Guardar envío de edición de metadatos en mis metadatos en edición:** el sistema debe permitir a los bibliotecarios de contenido escoger de todos los envíos realizados por los autores, los OA o DI a los que desee editar los metadatos y guardarlos en su lista metadatos a editar.

**R 27 Mostrar OA y DI con metadatos a editar guardados:** el sistema debe mostrar al bibliotecario de contenido una lista con todos los OA y DI que ha guardado en su lista para editar los metadatos.

**R 28 Configurar cantidad máxima de OA y diseño instruccional con metadatos a editar:** el sistema debe darle la posibilidad al administrador de configurar la cantidad máxima de OA y diseño instruccional, que un bibliotecario de contenidos puede poseer en su lista de OA con metadatos a editar.

**R 29 Insertar metadatos:** el sistema debe permitir al bibliotecario de contenidos, insertar metadatos al OA y al DI.

**R 30 Establecer metadatos obligatorios para los OA creados con SCORM 2004:** el sistema debe permitir al bibliotecario de contenido definir elementos y/o categorías que no pueden dejar de describirse en cada componente de los OA creados con SCORM 2004, porque han sido metadatos definidos como importantes en la organización.

**R 31 Establecer metadatos obligatorios para el diseño instruccional:** el sistema debe permitir al bibliotecario de contenido definir elementos y/o categorías que no pueden dejar de describirse en los diseños instruccionales.

**R32 Mostrar metadatos obligatorios:** el sistema debe mostrar al usuario solamente los metadatos obligatorios propuestos por el bibliotecario de contenidos y por SCORM 1.2, diferenciándolos entre sí.

### **Requisitos no funcionales del sistema**

Los requerimientos no funcionales son características que describen alguna restricción para la realización de algún requerimiento o conjunto de ellos e inclusive todos los requerimientos. Se consideran los atributos del sistema, propiedades que debe tener el producto. En este trabajo se identifican los siguientes:

- **Apariencia o interfaz externa**

La interfaz no debe contener gran cantidad de imágenes ni otros medios que ralenticen la navegación. El diseño de interfaz de este módulo debe ajustarse al de CRODA 1.0, en general ha de tener un estilo sobrio, formal y con una navegación sencilla y sugerente, así como la información y los textos han de ser concisos y legibles.

- **Soporte**

El sistema luego de implementado e incorporado a CRODA debe tener un funcionamiento estable y seguro y el rendimiento no se debe ver afectado por desperfectos o inestabilidades de funcionamiento de los servidores (Web y de bases de datos) que lo soporten.

- **Usabilidad**

El módulo podrá ser usado por cualquier persona que posea un nivel básico de conocimientos de computación. El mayor peso recae en el rol bibliotecario de contenidos ya que debe poseer conocimientos específicos de los elementos del esquema de metadatos LOM y sobre el estándar SCORM para desarrollar sus tareas en la herramienta. Debe presentar una interfaz especial para usuarios no expertos en informática, ya que los creadores de OA y DI mayormente son profesores y muchos no tienen un buen conocimiento computacional, una interfaz muy amigable y fácil de trabajar y que pueda ser usada por los usuarios fácilmente.

### *2.3 Identificación de actores del sistema*

Un actor es algo que se comunica con el sistema o producto y que es externo al sistema en sí mismo. La siguiente tabla muestra la relación de los actores que interactúan en el módulo.

#### **Identificación de actores del sistema.**

Actor	Descripción
Aut_DI	Actor del sistema que representa los roles Autor y Diseñador Instruccional de la herramienta CRODA.  <b>Autor:</b> Es el rol encargado de tomar el diseño instruccional básico y el esquema de contenidos del diseñador instruccional y escribe todos los textos de instrucción. Puede escribir y crear elementos de evaluación y de autoaprendizaje. Puede además, escribir los metadatos.  <b>Diseñador Instruccional:</b> Es la persona encargada de crear el diseño instruccional y describirlo a través de metadatos.
Administrador	Es el rol encargado de las funcionalidades administrativas de la herramienta de autor CRODA.

Bibliotecario de contenidos	Es el rol encargado de describir los OA y diseños instruccionales cuando los autores y diseñadores instruccionales soliciten una edición de los metadatos. Especifica valores predeterminados para algunos elementos de información y puntualiza la obligatoriedad de elementos de información de interés en la institución, manteniendo los registros de metadatos de los assets, SCO (Sharable Content Object), agregaciones, los paquetes de contenido y de los diseños instruccionales.
Aut_Bib_DI	Actor del sistema que representa los roles de Autor, Bibliotecario de contenidos y Diseñador instruccional de la herramienta CRODA.

### *2.3 Identificación de casos de uso del sistema*

Un caso de uso es un escenario que describe cómo el software va a ser usado en determinada situación. Los casos de uso deben cumplir los siguientes objetivos: definir los requisitos funcionales y operativos del sistema, diseñando un escenario de uso acordado por el usuario final y el equipo de desarrollo; proporcionar una descripción clara y sin ambigüedades de cómo el usuario final interactúa con el sistema y viceversa y proporcionar una base para la validación de las pruebas. (Pressman, 2002)

A continuación se listan los casos de uso identificados para el módulo presentado:

1. Gestionar metadatos.
2. Autocompletar metadatos.
3. Establecer metadatos obligatorios para los OA creados con SCORM 1.2.
4. Establecer metadatos obligatorios para el diseño instruccional.
5. Establecer metadatos obligatorios para los OA creados con SCORM 2004.

6. Exportar archivo XML.
7. Importar archivo XML.
8. Establecer cantidad máxima de envíos a atender.
9. Publicar metadatos.
10. Gestionar publicación de metadatos.
11. Enviar metadatos a edición.
12. Atender envíos de metadatos a editar.
13. Reutilizar metadatos.

### *2.5 Diagrama de los casos de uso identificados en el sistema*

Luego de identificar los casos de uso y actores del sistema, se realiza una representación gráfica de los procesos (casos de uso) y su interacción con los actores identificados. Cada caso de uso debe comunicarse con al menos un actor.

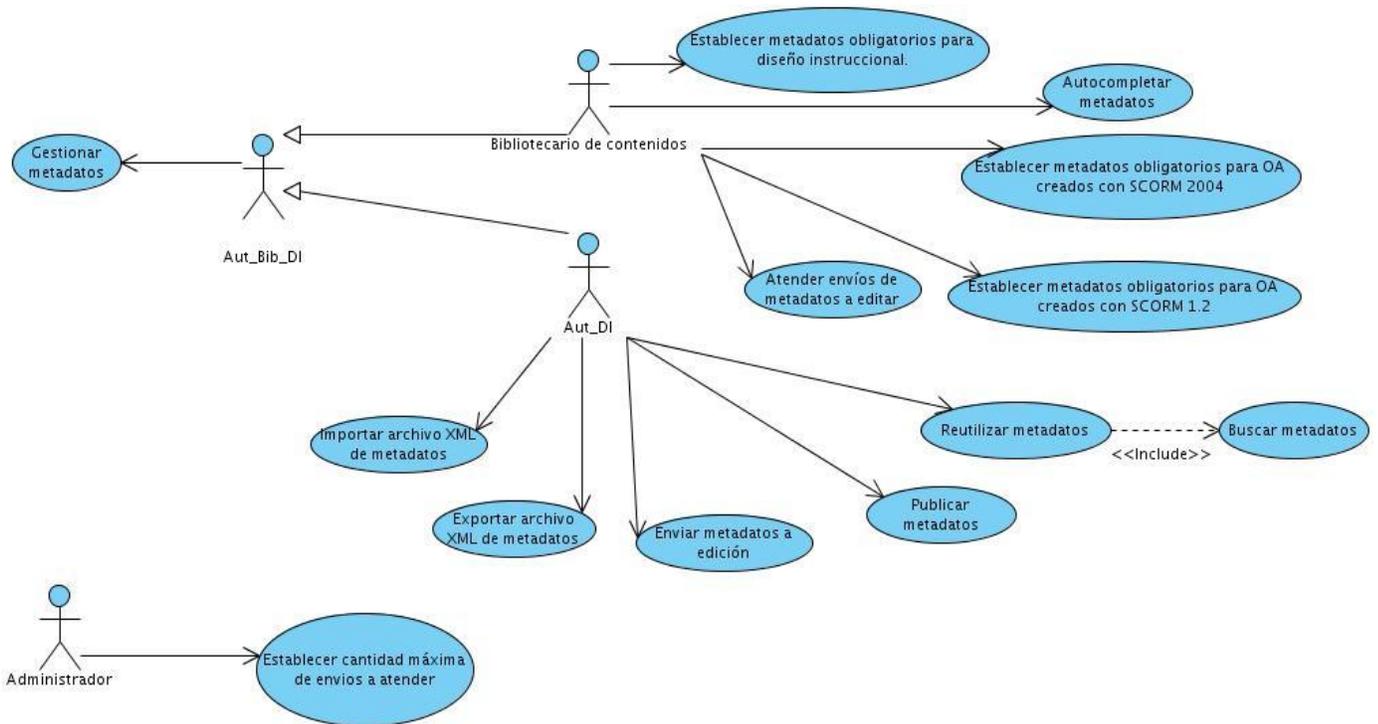


Figura 2 Diagrama de casos de uso del sistema.

### 2.3.4 Descripción textual de los casos de uso identificados

La descripción textual de cada caso de uso identificado en el sistema representa una explicación clara de la interacción entre el usuario y el sistema y viceversa. Mediante estas descripciones el programador podrá conocer un nivel de detalle de la interacción de los usuarios y el sistema, ofreciendo distintos escenarios en el que el usuario interactúa y el sistema tendrá que responder. Para cada caso de uso se realiza en este trabajo un diseño de prototipos de la interfaz de usuario, creando así un medio de comunicación efectiva entre el usuario y el sistema. Para ver la descripción textual de los casos de uso Gestionar metadatos y Atender envíos de metadatos a editar ver los Anexos 59 y 60 respectivamente. A continuación se muestran las restantes descripciones.

**Caso de Uso 2. Autocompletar metadatos.**

CU-2	Autocompletar metadatos.	
Actor:	Bibliotecario de contenidos	
Resumen:	El caso de uso se inicia cuando el bibliotecario de contenidos decide establecer un conjunto de valores que se pueden autocompletar para los metadatos de los OA, el sistema muestra los metadatos a ser autocompletados, el bibliotecario de contenidos autocompleta los valores de los mismos y los establece para todas las instancias de metadatos de los OA que se crean en la herramienta. Y finaliza así el caso de uso.	
Referencias	R17	
Prioridad	Secundario	
Precondiciones	El bibliotecario de contenidos debe estar autenticado en el sistema.	
Poscondiciones	Los metadatos fueron autocompletados correctamente.	
Flujo Normal		
Acción del actor	Respuesta del sistema	
1. El actor accede a la opción Autocompletar metadatos.	2. El sistema muestra los siguientes metadatos, ofreciéndole al actor la posibilidad de establecer valores para ellos: <ul style="list-style-type: none"> <li>- <b>Idioma</b> (Categoría General) para escoger el idioma del objeto de aprendizaje.</li> <li>- <b>Ámbito</b> (Categoría General) para escoger el país o países donde podrá aplicarse el OA.</li> <li>- <b>Idioma</b> (Categoría Meta-metadatos) para establecer el idioma con el que serán descritos los OA.</li> <li>- <b>Derechos de autor u otras restricciones</b> (Categoría Derechos) con valores: SI, NO, para establecer si el OA tiene o no derechos de autor u otras restricciones. (Ver sección 1)</li> </ul>	

*Capítulo 2: Características del sistema. Flujo de trabajo Requerimientos.*

	<p>- <b>Costo</b> (Categoría Derechos) con los valores: SI, NO, para establecer si el OA tendrá costo o no.</p> <p>-<b>Localización</b> (Categoría Técnica): tendrá por defecto la dirección del repositorio: <a href="http://roa.uci.cu">http://roa.uci.cu</a>. Esta podrá ser cambiada.</p> <p>El sistema permite además, establecer los metadatos seleccionados.</p>
3. El actor autocompleta los valores de los metadatos y accede a la opción Establecer.	4. El sistema verifica que los datos sean correctos y establece los valores insertados para todas las instancias de metadatos de los OA que se crean en la herramienta. Solamente para el OA en su totalidad (es decir, la Agregación de Contenido o el paquete) en su totalidad y finaliza el caso de uso.
Flujo Alternativo	
Acción del actor	Respuesta del sistema
3. El actor inserta una dirección URL incorrecta en el campo localización y presiona accede a la opción Establecer.	4. El sistema no establece los valores insertados y muestra un mensaje de aviso: "La dirección URL no es correcta".
<b>Sección 1: Establecer derechos de autor u otras restricciones</b>	
Acción del actor	Respuesta del sistema
1. El actor escoge la opción Si en el elemento Derechos de autor u otras restricciones.	2. El sistema muestra el elemento Descripción de la categoría Derechos para especificar las licencias de derecho de autor u otras restricciones para el uso de los objetos educativos.
3. El actor introduce los datos y accede a la opción Establecer.	4. El sistema verifica los datos entrados y establece los valores introducidos.
Prototipo de interfaz de usuario	

Ver Anexo 4

**Caso de uso 3. Establecer metadatos obligatorios para OA creados con SCORM 1.2.**

CU-3	Establecer metadatos obligatorios para OA creados con SCORM 1.2.	
Actor:	Bibliotecario de contenidos	
Resumen:	El caso de uso se inicia cuando el bibliotecario de contenido decide establecer un conjunto de metadatos como obligatorios para OA creados con SCORM 1.2, estos metadatos obligatorios serán de interés en la institución y ofrecen al usuario los elementos importantes que no debe dejar de completar en los OA creados con el estándar SCORM 1.2.	
Referencias	R18	
Prioridad	Crítico	
Precondiciones	El bibliotecario de contenidos debe estar autenticado en el sistema.	
Poscondiciones	Los metadatos fueron establecidos como obligatorios correctamente.	
Flujo Normal		
Acción del actor	Respuesta del sistema	
1. El actor accede a la opción Establecer metadatos obligatorios (SCORM 1.2).	2. El sistema permite establecer metadatos obligatorios para la Agregación de Contenido, Organizaciones de Contenido, SCOs y Assets. Para cada uno de estos componentes muestra todos los elementos del esquema LOM, con un campo para marcarlos, permitiéndole al bibliotecario establecerlos como obligatorios. Los metadatos obligatorios establecidos por SCORM 1.2 (Ver anexo 1) para cada componente aparecerán ya marcados y el bibliotecario de contenidos no podrá desmarcarlos. Muestra además la opción Establecer.	
3. El actor marca los metadatos que desea	4. El sistema establece como obligatorios los metadatos	

establecer como obligatorios en los OA y accede a la opción Establecer.	propuestos por el bibliotecario, mostrándolos en la vista de edición de metadatos junto a los obligatorios establecidos por SCORM 1.2 y finaliza así el caso de uso.
Prototipo de interfaz de usuario	
Ver Anexo 5	

**Caso de uso 4. Establecer metadatos obligatorios para el diseño instruccional.**

CU-4	Establecer metadatos obligatorios para el diseño instruccional.	
Actor:	Bibliotecario de contenidos	
Resumen:	El caso de uso se inicia cuando el bibliotecario de contenido decide establecer un conjunto de metadatos como obligatorios para el diseño instruccional y finaliza cuando realiza esta acción. Estos metadatos obligatorios serán de interés en la institución y ofrecen al usuario los elementos importantes y adecuados que no debe dejar de completar en los DI creados en CRODA.	
Referencias	R 31	
Prioridad	Crítico	
Precondiciones	El bibliotecario de contenidos debe estar autenticado en el sistema.	
Poscondiciones	Los metadatos fueron establecidos como obligatorios correctamente para los diseños instruccionales.	
CU asociados	-	
Flujo Normal		
Acción del actor	Respuesta del sistema	
1. El actor accede a la opción Establecer metadatos obligatorios para diseño instruccional.	2. El sistema muestra todos los elementos del esquema LOM con y permite al bibliotecario escoger los que desee establecer como obligatorios. Muestra además la opción Establecer.	

3. El actor escoge los metadatos que desea establecer como obligatorios para el diseño instruccional y accede a la opción Establecer.	4. El sistema establece como obligatorios los metadatos propuestos por el bibliotecario, mostrándolos en la vista de edición de metadatos y finaliza el caso de uso.
Prototipo de interfaz de usuario	
Ver Anexo 6	

**Caso de uso 5. Establecer metadatos obligatorios para OA creados con SCORM 2004.**

CU-5	Establecer metadatos obligatorios para OA creados con SCORM 2004	
Actor:	Bibliotecario de contenidos	
Resumen:	El caso de uso se inicia cuando el bibliotecario de contenido decide establecer un conjunto de metadatos como obligatorios para OA creados con SCORM 2004 y finaliza cuando realiza esta acción. Estos metadatos obligatorios serán de interés en la institución y ofrecen al usuario los elementos importantes que no debe dejar de completar en los OA creados con el estándar SCORM 2004.	
Referencias	R30	
Prioridad	Crítico	
Precondiciones	El bibliotecario de contenidos debe estar autenticado en el sistema.	
Poscondiciones	Los metadatos fueron establecidos como obligatorios correctamente.	
Flujo Normal		
Acción del actor	Respuesta del sistema	
1. El actor accede a la opción Establecer metadatos obligatorios (SCORM 2004).	2. El sistema permite establecer metadatos obligatorios para las Agregaciones de Contenido, Organizaciones de Contenido, SCOs y Assets. Para cada uno de estos componentes muestra todos los elementos del esquema LOM y permite al bibliotecario establecerlos como obligatorios. Muestra además la opción Establecer.	

3. El actor escoge los metadatos que desea establecer como obligatorios en los OA y accede a la opción Establecer.	4. El sistema establece como obligatorios los metadatos propuestos por el bibliotecario y finaliza el caso de uso.
Prototipo de interfaz de usuario	
Ver anexo 7	

**Caso de uso 6. Exportar archivo XML de los metadatos.**

CU-6	Exportar archivo XML de los metadatos.	
Actor:	Aut_DI	
Resumen:	El caso de uso se inicia cuando el actor decide exportar el archivo XML de la instancia de metadatos que está creando hacia una dirección física en el equipo donde está trabajando y finaliza cuando realiza la acción.	
Referencias	R9	
Prioridad	Opcional	
Precondiciones	El autor debe encontrarse en la edición de los metadatos. El diseñador instruccional debe encontrarse en la edición de los metadatos.	
Poscondiciones	El archivo XML se exportó correctamente.	
CU asociados	-	
Flujo Normal		
Acción del actor	Respuesta del sistema	
1. El actor accede a la opción Exportar metadatos.	2. El sistema ofrece mediante el navegador web la posibilidad de buscar la ubicación donde guardará el archivo.	
3. El actor busca la ubicación y guarda el archivo.	4. El sistema guarda en la ubicación seleccionada el archivo XML de los metadatos y finaliza el caso de uso.	
Prototipo de interfaz de usuario		

Ver anexo 8

**Caso de uso 7. Importar archivo XML de los metadatos.**

CU-7	Importar archivo XML de los metadatos.	
Actor:	Aut_DI	
Resumen:	El caso de uso se inicia cuando el actor decide Importar desde una dirección física del equipo donde está trabajando un archivo XML de los metadatos y finaliza cuando realiza la acción.	
Referencias	R10	
Prioridad	Opcional	
Precondiciones	El autor debe encontrarse en la edición de los metadatos. El diseñador instruccional debe encontrarse en la edición de los metadatos.	
Poscondiciones	Es importado un archivo XML a la instancia de metadatos.	
CU asociados	-	
<b>Flujo Normal</b>		
Acción del actor	Respuesta del sistema	
1. El actor accede a la opción Importar metadatos.	2. El sistema ofrece la posibilidad de buscar la ubicación del archivo, importarlo o cancelar la acción de importar el archivo.	
3. El actor busca la ubicación del archivo y accede a Importar el archivo seleccionado.	4. El sistema carga el archivo XML en la instancia de metadatos que se está creando y finaliza el caso de uso.	
<b>Flujo Alterno "Importar sin localizar el archivo"</b>		
Acción del actor	Respuesta del sistema	
3. El actor no busca la ubicación donde se encuentra el archivo y accede a la opción de Importar.	4. El sistema muestra el mensaje de aviso: "Debe buscar el archivo primeramente". Vuelve al paso 3 del Flujo Normal.	
<b>Flujo Alterno "Cancelar"</b>		
Acción del actor	Respuesta del sistema	
3. El actor accede a la opción Cancelar.	4. El sistema cierra la ventana de importar el archivo y	

	finaliza el caso de uso.
Flujo Alterno “Extensión del archivo no correcta”	
Acción del actor	Respuesta del sistema
3. El actor accede a la opción Examinar y busca un archivo que no es un XML de una instancia de metadatos LOM y accede a la opción de Exportar.	4. El sistema muestra el siguiente mensaje: “El archivo que usted seleccionó no es un archivo XML de metadatos del esquema LOM”. Regresa al paso 1 del flujo normal de eventos.
Prototipo de interfaz de usuario	
Ver anexo 9	

**Caso de uso 8. Establecer cantidad máxima de envíos a atender.**

CU-8	Establecer cantidad máxima de envíos a atender.
Actor:	Administrador
Resumen:	El caso de uso se inicia cuando el administrador decide establecer la cantidad máxima de envíos de edición de metadatos que los bibliotecarios de contenidos podrán atender y finaliza cuando realiza la acción.
Referencias	R25
Prioridad	Crítico
Precondiciones	El administrador debe estar autenticado en el sistema.
Poscondiciones	El actor estableció correctamente la cantidad máxima de envíos que un bibliotecario de contenidos puede atender.
CU asociados	-
Flujo Normal	
Acción del actor	Respuesta del sistema
1. El actor accede a la opción Cantidad de envíos de edición de metadatos.	2. El sistema ofrece la posibilidad de entrar la cantidad que desee y de establecer la misma.
3. El actor inserta la cantidad y accede a la opción Establecer.	4. El sistema limita la cantidad de envíos de OA y DI que un bibliotecario de contenidos puede tener en su lista

	para la edición de los metadatos, a la cantidad establecida por el administrador y finaliza el caso de uso.
Prototipo de interfaz de usuario	
Ver anexo 10	

**Caso de uso 9. Publicar metadatos.**

CU-9	Publicar metadatos.
Actor:	Aut_DI
Resumen:	El caso de uso se inicia cuando el actor decide publicar los metadatos para que sean visualizados y reutilizados por otros usuarios. El caso de uso termina cuando realiza estas acciones.
Referencias	R11
Prioridad	Crítico
Precondiciones	El autor debe encontrarse autenticado en el sistema y en la edición de los metadatos. El diseñador instruccional debe encontrarse autenticado en el sistema y en la edición de los metadatos. El actor debe haber guardado al menos una vez los cambios efectuados en el OA o el DI.
Poscondiciones	Los metadatos fueron publicados correctamente por el actor.
CU asociados	-
Flujo Normal	
Acción del actor	Respuesta del sistema
1. El actor accede a la opción Publicar metadatos	2. El sistema verifica que los metadatos no estén ya publicados y los publica con el título del OA, la fecha de publicación, descripción y muestra si la publicación es de metadatos de una agregación de contenido, de un ítem u organización o de un SCO.

	<p>Si la publicación ha sido realizada por un diseñador instruccional el sistema publica los metadatos con el título de diseño instruccional, la fecha de publicación y muestra si la publicación es de metadatos de un rol o actividad o ambiente u otro elemento del diseño instruccional.</p> <p>3. El sistema muestra el mensaje: “Sus metadatos han sido publicados para ser visualizados y reutilizados por otros usuarios.” Finaliza el caso de uso.</p>
Flujo Alterno	
Acción del actor	Respuesta del sistema
1. El actor accede a la opción Publicar metadatos.	2. El sistema verifica que los metadatos ya fueron publicados. Muestra al actor el siguiente mensaje: “Usted ya realizó una publicación de esta instancia de metadatos.” Finaliza el caso de uso.
Prototipo de interfaz de usuario	
Ver anexo 11	

**Caso de uso 10. Buscar metadatos.**

CU-10	Buscar metadatos. (Include del caso de uso Reutilizar metadatos).
Actor:	Aut_DI
Resumen:	El caso de uso se inicia cuando el actor decide realizar una búsqueda por criterios de metadatos publicados por todos los usuarios o listar todas las publicaciones realizadas por estos. El caso de uso termina cuando realiza estas acciones.
Referencias	R12, R15
Prioridad	Critico
Precondiciones	El actor debe encontrarse autenticado en el sistema y en la edición de los metadatos.

*Capítulo 2: Características del sistema. Flujo de trabajo Requerimientos.*

Poscondiciones	Las publicaciones de metadatos fueron buscadas y listadas correctamente por el actor.	
CU asociados	CU Reutilizar metadatos.	
Flujo Normal		
Acción del actor	Respuesta del sistema	
1. El actor decide realizar una búsqueda de metadatos publicados y accede a la opción Buscar metadatos.	2. El sistema brinda la posibilidad de escoger el tipo de metadatos que desea buscar mediante las opciones: <ul style="list-style-type: none"> <li>-Metadatos de Diseño Instruccional.</li> <li>-Metadatos de Objetos de Aprendizaje.</li> </ul> Permite al actor escoger los elementos del esquema LOM por los cuales desea realizar la búsqueda e introducir los valores que desea encontrar en esos elementos. Muestra además, la opción Buscar para ejecutar la búsqueda por los criterios establecidos	
3. El actor selecciona el tipo de metadatos que desea buscar.		
4. El actor selecciona los elementos por los cuales desea buscar, inserta los valores de los mismos y accede a la opción Buscar.	5. El sistema muestra una lista de publicaciones de metadatos que coinciden con los criterios de búsqueda introducidos por el actor y finaliza el caso de uso.	
Flujo Alterno "No selecciona el tipo de metadatos a buscar."		
Acción del actor	Respuesta del sistema	
3. El actor no selecciona el tipo de metadatos que desea buscar.		
4. El actor selecciona los elementos por los cuales desea buscar, inserta los valores de los mismos y accede a la opción Buscar.	5.El sistema muestra un mensaje de aviso: "Debe seleccionar el tipo de metadatos que desea buscar". Vuelve al paso 3 del flujo normal.	
Flujo Alterno "No inserta valores para los elementos de búsqueda".		

*Capítulo 2: Características del sistema. Flujo de trabajo Requerimientos.*

Acción del actor	Respuesta del sistema
4. El actor selecciona los elementos por los cuales desea buscar, no inserta ningún valor para estos elementos y accede a la opción Buscar	5. El sistema muestra una lista de publicaciones de metadatos que coinciden con los criterios de búsqueda (elementos seleccionados) introducidos por el actor y finaliza el caso de uso.
Flujo Alterno "No inserta elementos ni valores para la búsqueda"	
Acción del actor	Respuesta del sistema
4. El actor no selecciona ningún criterio de búsqueda y accede a la opción Buscar.	5. El sistema lista todos los metadatos públicos del tipo seleccionado y finaliza el caso de uso.
Flujo Alterno "No se encuentran resultados para los criterios de búsqueda".	
Acción del actor	Respuesta del sistema
4. El actor selecciona los elementos por los cuales desea buscar, inserta los valores de los mismos y accede a la opción Buscar.	5. El sistema no encuentra publicaciones de metadatos que coinciden con los criterios de búsqueda introducidos por el actor, muestra el mensaje de notificación: "No se encontraron metadatos publicados con los criterios de búsqueda introducidos". Finaliza el caso de uso.
Prototipo de interfaz de usuario	
Ver anexo 12.	

**Caso de uso 11. Enviar metadatos a edición.**

CU-11	Enviar metadatos a edición.
Actor:	Aut_DI
Resumen:	El caso de uso se inicia cuando el actor decide enviar una solicitud de edición de sus metadatos a los bibliotecarios de contenidos y termina cuando realiza esta acción.
Referencias	R16
Prioridad	Crítico.

*Capítulo 2: Características del sistema. Flujo de trabajo Requerimientos.*

Precondiciones	El actor debe encontrarse autenticado en el sistema y en la edición de los metadatos. El actor debe haber guardado al menos una vez el OA o el DI.
Poscondiciones	La solicitud de edición de metadatos fue enviada correctamente a los bibliotecarios de contenidos.
CU asociados	-
Flujo Normal	
Acción del actor	Respuesta del sistema
1. El actor accede a la opción Enviar metadatos a edición.	2. El sistema verifica que el actor no halla realizado una solicitud para la edición de metadatos del OA o DI y guarda la solicitud realizada. Por cada una el sistema guarda la fecha en que se envió, nombre del autor, título del OA o del DI, Tipo (Objeto de Aprendizaje o Diseño Instruccional).  Si el actor termina la creación del OA o del DI y su envío no ha sido atendido aún, por un bibliotecario de contenidos, el sistema automáticamente elimina este envío de la lista y finaliza el caso de uso.  Si su envío ha sido atendido no podrá terminar la creación hasta que el bibliotecario de contenido no termine su edición o se venza la misma y finaliza el caso de uso.
Flujo Alterno	
Acción del actor	Respuesta del sistema
1. El actor accede a la opción Enviar metadatos a edición.	2. El sistema verifica que la solicitud ya está creada y envía un mensaje de aviso al actor: "Ya usted realizó una solicitud de edición de metadatos para este OA o DI".

Prototipo de interfaz de usuario
Ver Anexo 15

**Caso de uso 13. Reutilizar metadatos.**

CU-10	Reutilizar metadatos
Actor:	Aut_DI
Resumen:	El caso de uso se inicia cuando el actor decide visualizar o reutilizar metadatos obtenidos en el resultado de una búsqueda de los metadatos publicados por otros usuarios y termina cuando realiza estas acciones. Este caso de uso permite al actor utilizar metadatos creados por otros usuarios para describir sus contenidos, el sistema posibilita editar estos metadatos o reutilizar exactamente los mismos.
Referencias	R13, R14.
Prioridad	Critico
Precondiciones	El actor debe encontrarse autenticado en el sistema y en la edición de los metadatos. El actor debe haber obtenido el resultado de una búsqueda de metadatos.
Poscondiciones	Los metadatos publicados fueron visualizados y utilizados correctamente por el actor.
CU asociados	-

**Flujo Normal**

Acción del actor	Respuesta del sistema
1. El actor realiza una búsqueda de metadatos publicados.	2. El sistema en cada publicación de metadatos listada o buscada por el usuario muestra las opciones. Visualizar (Ver sección 1). Reutilizar (Ver sección 2).

**Sección 1 :“Visualizar metadatos”**

*Capítulo 2: Características del sistema. Flujo de trabajo Requerimientos.*

Acción del actor	Respuesta del sistema
1. El actor accede a la opción Visualizar metadatos que se encuentra en las publicaciones listadas o buscadas.	2. El sistema muestra los elementos de información con su valor correspondiente en forma jerárquica. Ejemplo:  <b>General</b> Titulo: Inteligencia artificial Palabras claves: IA. <b>Ciclo de vida</b> Contribución: Autor Y Finaliza el caso de uso.
Sección 2 :“Reutilizar metadatos”	
Acción del actor	Respuesta del sistema
1. El actor accede a la opción Reutilizar metadatos que se encuentra en las publicaciones listadas.	2. El sistema ubica al actor en la vista de edición de los metadatos y carga el XML de los metadatos publicados en la instancia que el actor esté creando en ese momento y finaliza el caso de uso.
Prototipo de interfaz de usuario	
Ver anexos 13 y 14.	

**Conclusiones**

En el presente capítulo debido a la insuficiente conformación de los procesos del negocio se ha realizado un modelo de dominio para ofrecer una visión general del ámbito donde se desarrolla este módulo, brindando una base para el desarrollo del flujo de trabajo Requerimientos. Se determinaron un total de treinta y dos requerimientos funcionales y se identificaron los requerimientos no funcionales que debe tener el sistema, asimismo se desarrolló el modelo de casos de uso del sistema, donde se identificaron cuatro actores del sistema, uno de ellos, el Bibliotecario de contenidos representa un nuevo rol a incorporar en la herramienta CRODA.

Se crearon trece casos de usos implicando dentro de ellos los requerimientos identificados. De estos casos de uso, diez, fueron clasificados como críticos, representando las funcionalidades a priorizar ya que

## *Capítulo 2: Características del sistema. Flujo de trabajo Requerimientos.*

son las más necesarias e importantes para un correcto funcionamiento del sistema. A través de los artefactos generados se especificó qué es lo que el módulo debe hacer, siendo la base para definir a través de los artefactos del flujo de trabajo Análisis y Diseño, de qué manera lo debe hacer, en un lenguaje entendible al programador.

## *Capítulo 3. Representación técnica del sistema. Flujo de trabajo Análisis y Diseño*

### **Introducción**

El presente capítulo se centra en el flujo de trabajo Análisis y Diseño de la metodología RUP, es importante destacar que aunque RUP contempla las disciplinas de la ingeniería de software, Análisis y Diseño, en un mismo flujo de trabajo por estar muy relacionadas, son actividades diferentes con artefactos diferentes. En este capítulo primeramente se realiza la modelación de Análisis a través de la realización de los diagramas de clases del análisis y los diagramas de colaboración, luego se modela el diseño mediante los diagramas de clases del diseño con estereotipos web.

### *3.1 Análisis*

El análisis es una disciplina que consiste en obtener una visión de los requisitos funcionales, para ellos se analizan los requisitos que se obtuvieron en la captura de requerimientos, refinándolos y estructurándolos. El objetivo de hacerlo es conseguir una comprensión más precisa del sistema y una representación de las características del mismo. El análisis se realiza para ayudar a estructurar el sistema entero, incluyendo su arquitectura.

#### **Diagrama de clases del análisis**

Los diagramas de clases de análisis representan un modelo que describe las características y funcionamiento de los elementos que existen en el sistema, se centran en los requerimientos funcionales y se representan mediante tres estereotipos:

- **Interfaz:** se encargan de la modelación de toda la interacción que puede existir entre los actores y el sistema.
- **Control:** representan la coordinación, secuenciación, transacciones y a veces la lógica del negocio; se emplean a menudo para encapsular el control referido a un caso de uso.
- **Entidad:** representa la información de larga duración y persistente que se maneja en el sistema.

En este trabajo se realizó un diagrama de clases del análisis para cada caso de usos identificado, permitiendo un mejor entendimiento y seguimiento de los requerimientos involucrados en cada uno de ellos. (Ver Anexos del 20 al 32).

### **Diagramas de interacción**

Los diagramas de interacción representan una vista dinámica de la interacción entre los objetos identificados en el sistema y estos diagramas se clasifican en dos tipos: de colaboración y de secuencia. En este trabajo se realizaron los diagramas de colaboración del análisis. (Ver los anexos del 33 al 45).

### *3.2 Arquitectura propuesta*

La arquitectura de software es la manera en que se organiza un sistema, determina las relaciones entre sus componentes y los principios que orientan su diseño y evolución. Esta establece algunas reglas y conceptos fundamentales para que analistas, diseñadores, programadores, puedan trabajar en una misma línea permitiendo alcanzar los objetivos pertinentes. En este trabajo se propone para la realización del diseño y la implementación el patrón clásico del diseño web conocido como arquitectura Modelo-Vista-Controlador (MVC). El framework Symfony, propuesto para la implementación de este módulo, está basado en este patrón conformado por tres niveles:

- El Modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La Vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- El Controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación. (Fabien Potencier, 2008)

### 3.3 Diseño

El diseño es una disciplina que se realiza luego del análisis, haciendo un refinamiento del mismo. El diseño crea una representación o modelo del software, proporciona detalles acerca de las estructuras de datos, las arquitecturas, las interfaces y los componentes del software que son necesario para implementar el sistema. (Pressman, 2002)

#### Diagrama de clases del diseño para aplicaciones web

El módulo presentado se diseña para una herramienta web por lo que es necesario modelar las páginas, los enlaces entre estas, todo el código que irá creando las páginas, así como el contenido dinámico de estas, para ello en este trabajo se emplean los estereotipos de UML para diseñar los diagramas de clases del diseño para aplicaciones web. A continuación se muestran los tres estereotipos con los iconos utilizados para su representación en este trabajo. Para visualizar los diagramas realizados para cada caso de uso identificado ver los anexos del 46 al 58.

**Tabla de estereotipos web empleados.**

Ícono / Estereotipo	Explicación
 <<Client Page >>	Es la página web de formato HTML, que le brinda al usuario toda la interfaz de la aplicación. Cada página cliente es construida por una sola página de servidor.
 <<Server Page>>	Es la clase encargada del código ejecutable en el servidor. Esta clase es la encargada de construir las paginas clientes mediante una relación build.

 <p data-bbox="430 415 565 443">&lt;&lt;Form&gt;&gt;</p>	<p data-bbox="651 321 1417 506">Es una colección de elementos de entradas que están contenidos en la página cliente. Estas no tienen operaciones y se comunican con las paginas servidoras a través de una relación submit.</p>
-------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### Conclusiones

Como resultado de la elaboración de este capítulo se ha alcanzado primeramente con la realización del Análisis un refinamiento y mayor entendimiento de los requerimientos funcionales, logrando una corrección, mayor integridad y consistencia de lo que el sistema debe hacer, estableciendo una base que condujo a la realización del Diseño. Finalmente con la construcción de los artefactos del Diseño realizados en este trabajo, basados en el patrón arquitectónico modelo- vista -controlador, quedó estructurado el módulo en un lenguaje entendible al programador.

### *Conclusiones Generales*

Con la realización de esta investigación se analizaron aspectos relacionados con los metadatos y con la necesidad que existe de incorporar un módulo para la gestión de los mismos en la herramienta de autor CRODA, para disfrutar del beneficio que este proporciona, llegando así a las siguientes conclusiones:

- ✓ El esquema de metadatos LOM, se encuentra entre las principales soluciones para la descripción de recursos didácticos.
- ✓ Facilitar la descripción de los recursos a los profesores incide positivamente en el aumento de la reutilización de los mismos.
- ✓ Con los artefactos generados en los flujos de trabajo Requerimientos y Análisis y Diseño de la metodología RUP, se crearon las bases para la implementación del módulo que permita la gestión de metadatos en CRODA 2.0.

## *Recomendaciones*

Se recomienda para investigaciones futuras con el objetivo de perfeccionar y aplicar este módulo en la práctica:

- ✓ Teniendo en cuenta la creación colaborativa que se lleva a cabo en CRODA, se recomienda crear un historial de los cambios realizados a los metadatos por cada uno de los autores del OA, que sirva de consulta en cualquier momento de la creación.
- ✓ Validar la propuesta realizada para la gestión de metadatos en CRODA.
- ✓ Implementar el módulo propuesto en este trabajo para la gestión de metadatos en CRODA.

Referencias Bibliográficas

**ADL, Angelo Panar, [ed.]. 2009.** *Sharable Content Objetc Reference Model Versión 2004 . The SCORM Content Agregation Model.* 4ta edición. Alexandria : s.n., 2009. Vol. 1.

**ADL. 2008.** 4.1 SCORM Team Members. *ADL Guidelines for creating reusable content with SCORM 2004.* Alexandria : s.n., 2008, Vol. 1.

**ADL. 2006.** *Modelo de Referencia de Objetos de Contenido Compartido. Modelo de Agregación de Contenidos (CAM). Versión 1.3.2 (Borrador).* 2006.

**ADL. 2001.** *Sharable Content Objetc Reference Model Versión 1.2 . The SCORM Content Agregation Model.* [ed.] Philip Dodds (ADL). Alexandria : s.n., 2001. Vol. 1. 2001.

**Álvarez, Miguel Ángel. 2008.** Introducción a Javascript. Qué es JavaScript y las posibilidades que nos ofrece con respecto al HTML. [En línea] 2008a. [Citado el: 5 de febrero de 2010.] <http://www.desarrolloweb.com/articulos/490.php>.

**Apache Software Foundation. 2011.** Apache. [En línea] 2011. [Citado el: 6 de junio de 2011.] <http://httpd.apache.org/>.

**Ayuntamiento de Cartagena. 2011.** welearn.cartagena. [En línea] 2011. [Citado el: 6 de junio de 2011.] <http://welearn.cartagena.es/moodle/mod/book/view.php?id=509&chapterid=11>.

**Berlanga, Adriana J, y otros. 2006.** *Consideraciones para reforzar el Valor de los Metadatos en los Objetos de Aprendizaje.* Departamento de Informática y Automática, Instituto Universitario de Ciencias de la Educación, Universidad de Salamanca. Salamanca : Editorial de Universidad de Salamanca, 2006. Contribución en el Congreso de "Contenidos educativos reutilizables" en la Universidad de Cataluña.

**Calderón, Amaro, Valverde Rebaza, Sarah Dámaris y Carlos, Jorge. 2007.** *Metodologías Ágiles.* Perú: s.n., 2007.

**Cibernetia. 2011.** cibernetia. [En línea] 2011. [Citado el: 6 de junio de 2011.] [http://www.cibernetia.com/manuales/instalacion\\_servidor\\_web/3\\_otros\\_servidores\\_web.php](http://www.cibernetia.com/manuales/instalacion_servidor_web/3_otros_servidores_web.php).

- Chiappe Laverde, Andrés, Segobia Cifuentes, Yasbley y Rincón Rodríguez, Helda Yadira. 2007.** Educational Technology Research and Development. *Toward an instructional design model based on learning objects*. 2007. Vol. 55, 6. 10.1007/s11423-007-9059-0. Condinarmuca. Colombia.
- desarrolloweb. 2011.** [En línea] 6 de junio de 2011. <http://desarrolloweb.com/css>.
- desarrolloweb. 2011.** [En línea] 2011. [Citado el: 6 de junio de 2011.] <http://www.desarrolloweb.com/html>.
- EVA. 2007.** Entorno Virtual de Aprendizaje. [En línea] Universidad de las Ciencias Informáticas, 2007. [Citado el: 29 de noviembre de 2010.] <http://eva.uci.cu/mod/resource/view.php?id=34099>.
- EVA. 2008.** Entorno Virtual de Aprendizaje. [En línea] 2008. [Citado el: 2 de abril de 2011.] <http://eva.uci.cu/mod/resource/view.php?id=34198>
- Exe. 2008.** exelearning. [En línea] 2008. <Http://exelearning.org>.
- Exist. 2008.** exist. [En línea] 2008. [Citado el: 7 de enero de 2011.] <Http://exist.sourceforge.net>.
- Fabien Potencier, François Zaninotto. 2008.** *Symfony La guía definitiva*. [trad.] Miguel Sanchez, Luciano A. Andrade , Martín Palacio Pentucci Javier Eguíluz Pérez. s.l. : Apress, 2008. ISBN-13: 978-1590597866.
- Fernández, B, Moreno, B y Sierra, J. 2007.** Uso de estándares aplicados a las TICs en educación. 2007.
- García Peñalvo, Francisco José and López Guzmán, Clara. 2006.** *Estándares y Especificaciones para los Entornos e-learning: Convergencia en Contenidos y Sistemas*. Mexico : UNAM (México), 2006. En "Memorias del Encuentro Internacional de Educación Superior, Virtual Educa 2006".
- García Peñalvo, Francisco José. 2005.** Estado Actual de los sistemas e-learning. 2, Salamanca: Ediciones Universidad de Salamanca, 2005, *Teoría de la Educación: Educación y Cultura en la Sociedad de la Información.*, Vol. 6. ISSN 1138 9737.
- Grupo de Catalogadores de Información Geográfica. 2011.** metadatos.latingeo. [En línea] 2011. [Citado el: 6 de junio de 2011.] [http://metadatos.latingeo.net/metadatos/Que\\_son](http://metadatos.latingeo.net/metadatos/Que_son).

**Guzmán, Clara López. 2005.** *Los repositorios de Objetos de Aprendizaje como soporte a un entorno e-learning*. Salamanca : Universidad de Salamanca, 2005. Doctorado en procesos de formación en Espacios Virtuales.

**IBM. 2009.** IBM. [En línea] 2009. [Citado el: 14 de enero de 2011.] <http://www-142.ibm.com/software/products/es/es/enterprise/>.

**IMS. 2011.** imsglobal. [En línea] 2011. [Citado el: 20 de 3 de 2011.] [www.imsglobal.org/content/packaging/](http://www.imsglobal.org/content/packaging/).

**JORUM. 2006.** The JISC Online Repository for [learning and teaching] Materials: JORUM. Scoping and Technical Appraisal Study, Volume V: Metadata. [En línea] 6 de Mayo de 2006. [Citado el: 13 de diciembre de 2010.] <http://www.jorum.ac.uk/docs/pdf/>.

**Jorum y Baird, Kenny. 2006.** Automated Metadata. *Jorum*. [En línea] 26 de Julio de 2006. [Citado el: 15 de diciembre de 2010.] <http://www.jorum.ac.uk/docs/pdf/>.

**López Guzmán, Clara y García, Francisco. 2008.** Formación de repositorios de objetos de aprendizaje a través de la reutilización de los metadatos de una colección digital: de Dublin Core a IMS. Dirección General de Servicios de Cómputo Académico, Universidad Nacional Autónoma de México. Guadalajara : s.n., 2006. pág. 10.

**LTSC. 2002.** *IEEE Standard for Learning Object Metadata*. New York: IEEE -SA, 2002. IEEE Standard 1484.12.1.

**Montero, J. L; Herrero, E. 2008.** *Las herramientas de autor en el proceso de producción de cursos en formato digital*, Sevilla.España : Secretariado de Recursos Audiovisuales y Nuevas Tecnologías. Universidad de Sevilla., 2008, Pixel Bit Revista de medios y educación. ,33 , págs. 59-72. ISSN 1133-8482.

**Oracle Corporation. 2011.** Oracle. [En línea] 2011. [Citado el: 6 de junio de 2011.] <http://www.oracle.com>.

**Postgresql. 2007.** postgresql. [En línea] 2007. [Citado el: 6 de enero de 2011.] <http://www.postgresql.org>.

**Pressman, Roger s. 2002.** *Ingeniería del Software. Un enfoque práctico.* [ed.] Concepción Fernández. [trad.] Dpto de lenguajes y sistemas informáticos y escuela universitaria de informática de la universidad Pontificia de Salamanca. 6ta Edición en Español. Salamanca: MC Graw Hill, 2002. pág. 956. Vol. 1 y 2. ISBN: 970-10-5473-3.

**ReLoad. 2008.** ReLoad. [En línea] 2008. <http://www.reload.ac.uk/>.

**Sommerville, Ian. 2005.** *Ingeniería del Software.* Madrid : PEARSON EDUCACIÓN, SA., 2005. pags 5 y 124. ISBN:84-7829-074-5.

**Sencha. 2006.** sencha. [En línea] 2006. [Citado el: 9 de enero de 2011.] <http://www.sencha.com/products/js/>.

**TTnet, Red. 2005.** *La formación sin distancia. Estudio realizado por el Grupo de Trabajo de e-learning de la red TTnet.* 2005.

**Universidad Carlos III de Madrid. 2011.** metadatos-XMLRDF. [En línea] 2011. [Citado el: 6 de junio de 2011.] <http://www.metadatos-xmlrdf.com/metadatos>.

**visualparadigm. 2009.** visualparadigm. [En línea] 2009. [Citado el: 30 de diciembre de 2010.] <http://www.visual-paradigm.com/product/vpuml/provides/>.

**W3C. 2008.** W3C. *Extensible Markup Language (XML) 1.0 (Fifth Edition).* [En línea] 26 de noviembre de 2008. [Citado el: 10 de diciembre de 2010.] <http://www.w3.org/TR/2008/REC-xml-20081126/>

**Zend. 2011.** [En línea] 2011. <http://www.zend.com>