Universidad de las Ciencias Informáticas

Facultad 4

Implementación del módulo Metadatos para la herramienta de autor CRODA 2.0.

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

Autor: Ramiro González Almarales

Tutores: Ing. Dunia María Colomé Cedeño

Ing. Osvaldo Ernesto Stable Vilches

Ciudad de La Habana, Junio 2012

Por este medio declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) para que haga el uso que estime pertinente con este trabajo.

Para que así conste firmo la presente a los _____ días del mes de _____ del 2012.

_____

Ramiro González Almarales

Firma del autor

_____ _____

Ing. Dunia María Colomé Cedeño

Firma de la tutora

Ing. Osvaldo Ernesto Stable Vilches

Firma del tutor

Agradecimientos

A Jorge Iturria por guiarme en la implementación e integración de este trabajo con la herramienta CRODA, por apoyarme, por ser tan buen compañero de trabajo.

A mis tutores Dunia María y Osvaldo Estable por haberme dado la oportunidad de desarrollar este tema de tesis, por corregirme y guiarme.

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

Agradecer mis amigos:

Yosniel, Imanol, Hang(El capo), Yasmany, Renin, Leandro por estar presente en los momentos que mas los necesité.

A todos mis compañeros aquí en la Universidad.

Dedicatoria

Este trabajo lo dedico a mis padres Virgen y Ramiro 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 ingeniero.

A mis hermanos Yailix y Raimir por quererme como tal.

A mi familia por estar siempre apoyándome.

Y a mi novia Yaima Laugart por estar presente cuando más la necesité, en los momentos de tristeza y alegría.

Resumen:

Las herramientas de autor en la actualidad tienen gran importancia y utilidad dentro del e-learning (aprendizaje electrónico), estas permiten la creación de los objetos de aprendizaje, recursos que se crean con el fin de ser localizados y usados debido a 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 o legales, dependiendo del esquema de metadatos empleado. Además facilitan la realización de búsquedas de Objetos de Aprendizaje en los repositorios donde se almacenan.

En el presenta trabajo se pretende implementar un módulo para la gestión de metadatos en la herramienta de autor CRODA (**C**reando **O**bjetos **D**e **A**prendizaje) de tal manera que se fortalezca gradualmente el proceso de creación de Objetos y Unidades de Aprendizaje, favoreciendo la actividad del llenado de sus metadatos, que repercute, en la localización de los objetos dentro de un repositorio y contribuyendo a que su descripción se realice con mayor facilidad a través de un mismo componente.

En este trabajo se obtiene la fundamentación teórica necesaria para su realización, para ello se utiliza bibliografía actualizada de los temas relacionados, se crean los componentes necesarios con que debe contar el sistema, llegando así a las funcionalidades que requiere el módulo.

Palabras clave: Herramienta de autor, CRODA, metadato, e-learning, estándar.

Índice

Introducción

El vertiginoso desarrollo de las tecnologías de la información y las comunicaciones (TIC) ha provocado cambios en diversas áreas y procesos, sin quedar exento el proceso de formación. En este ámbito se pueden apreciar potenciales cambios en la manera en que el estudiante recibe y materializa los conocimientos y en la forma en que el profesor imparte o elabora las clases o cursos. Con la vinculación de las tecnologías y el proceso de enseñanza-aprendizaje surge el término "e-learning" que se traduce como aprendizaje electrónico e implica dentro de sí nuevos métodos, tecnologías y recursos que posibilitan el aprendizaje por cualquier medio electrónico, permitiendo así que las personas con acceso a las tecnologías tengan la oportunidad de aprender haciendo uso de recursos educativos, sin la presencia física permanente de un profesor.

El e-learning no sustituye el método tradicional de enseñanza, pero si lo fortalece en gran medida debido a que promueve una nueva concepción del proceso de enseñanza-aprendizaje donde el profesor representa un guía o tutor que puede incluso retroalimentarse del conocimiento de sus estudiantes originando una mayor interacción y participación del estudiante en las actividades propuestas. Además permite romper las barreras de tiempo y espacio. Con el desarrollo del aprendizaje electrónico surge la necesidad de intercambiar recursos entre plataformas para su uso y reutilización. Como respuesta a esta problemática surgen tecnologías como los Objetos de Aprendizaje (OA), diferenciados de otros recursos digitales por las características: durabilidad, accesibilidad, interoperabilidad y reusabilidad. En este trabajo se asume la definición de OA del autor Leonardo Rodríguez: "*Recurso didáctico digital estandarizado, descrito por metadatos, compuesto por uno o varios objetos de información, que responde a un único objetivo de aprendizaje y puede ser utilizado en diversas situaciones de enseñanza-aprendizaje*" (1), elaborado en la Universidad de las Ciencias Informáticas.

En la definición anteriormente mencionada sobre los OA se aborda que estos deben estar descritos con 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 esté descrito siguiendo un esquema de metadatos. El término metadatos se refiere a la información de la información (datos de los datos), o sea, "son información estructurada que describen, explican, localizan o de alguna manera facilitan la obtención, el uso o la administración de un recurso de información" (2).

A partir del anterior concepto de metadatos, se puede inferir, que los metadatos a través de funcionalidades de búsquedas que brindan los Repositorios de Objetos de Aprendizaje (ROA). Son un medio por el cual el profesor puede conocer las características del recurso sin necesidad de acceder al mismo, informándose si le es útil para sus objetivos. 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 consecuente su uso y reutilización no serán óptimos.

Entre las herramientas creadas para la gestión de los OA se desatacan, las Herramientas de Autor (HA) y los ROA para su creación y almacenamiento respectivamente. Los ROA garantizan el almacenamiento organizado del OA permitiendo su localización, recuperación, catalogación, reutilización, revisión y eliminación. Según Guzmán se dividen en dos tipos, los que almacenan los recursos con su respectiva descripción con metadatos y los que almacenan los metadatos y referencian el objeto de otro repositorio (3). Las HA por su parte permiten la creación de OA, según las características propias de cada una, proveen al profesor las funcionalidades necesarias para la realización de las diferentes actividades como son: la descripción, exportación e importación de los objetos; así como la creación de ejercicios de diferentes tipos, y la descripción de las unidades de aprendizaje para puntualizar las actividades que realizarán los alumnos y profesores posteriormente, indicando en qué momento y condiciones, o con qué OA y servicios se relacionarán. El término Unidad de Aprendizaje *"se refiere la descripción del procesos de enseñanza-aprendizaje unido a los recursos físicos que dicho proceso hará uso, incluyendo los objetos de aprendizaje, se puede empaquetar en una única entidad llamada Unit of Learning también conocida como UOL"* (4).

Para lograra la correcta interrelación entre las diferentes herramientas relacionadas con los objetos y unidades de aprendizaje se hace uso de estándares. Los estándares especifican una serie de normas establecidas por determinadas instituciones avaladas en el tema, que posibilitan organizar, localizar, recuperar e intercambiar recursos. En otras palabras garantizan la interoperabilidad con otros sistemas que se ajusten a los mismos estándares, aspecto muy importante que se debe tener en cuenta durante la creación y almacenamiento de los OA. En la actualidad existen estándares dentro del e-learning que permiten a los sistemas la comunicación y el intercambio de información de forma transparente, denominados, estándares de interoperabilidad, ejemplos de estos estándares son: Instructional Management Systems - Digital Repositories Interoperability (IMS-DRI), Simple Query Interface (SQI) y

Open Knowledge Initiative (O. K. I.). También se encuentran los estándares de contenidos, que para lograr la homogeneidad en el desarrollo de contenidos, proponen estructurarlos de manera tal que tenga ciertas características y componentes agregados que los hagan consistentes, para que puedan ser manipulados por los ROA, sin la necesidad de estar desarrollados en un mismo formato (5), por ejemplo, Sharable Content Object Reference Model (SCORM) y la especificación Content Packaging de Instructional Management System (IMS CP). Y los estándares para la catalogación de recursos como Learning Object Metadata (LOM) y Dublin Core, utilizados en el proceso de descripción de un OA o una UoL.

Teniendo en cuenta que el estándar LOM esta orientado al ámbito educativo y que por consecuente presenta toda una categoría dedicada al uso educativo de los recursos, resulta de interés para esta investigación. LOM esta compuesto por sesenta y cuatro elementos de información ofreciendo cierta riqueza semántica al recurso, pero a su vez puede ocasionar que la descripción se haga un poco monótona. La descripción de los recursos es una actividad muy importante para su reutilización sin embargo en ocasiones no se le presta la atención necesaria o se desconocen los valores correctos o incorrectos en cada uno de los metadatos por parte de los profesores. La selección correcta, la asignación adecuada del esquema de metadatos y sus valores, proporcionan en los contenidos las propiedades necesarias para potenciarlos como recursos reutilizables, asequibles y durables (5). Es necesario entonces que las HA, brinden funcionalidades que guíen y ayuden al profesor para una correcta gestión de los metadatos durante el proceso de completamiento.

Existen en el mundo varias HA como RELOAD y AUTORe que emplean diferentes estándares de catalogación, ejemplo: LOM o Dublin Core, para la descripción de los recursos en el proceso de creación. Otras herramientas como los ROA: RHODA, e-prints y JORUM emplean metadatos para la creación y realización de búsquedas de los OA, permitiendo simplemente añadir metadatos a los recursos creados, con una interfaz amigable y con elementos de ayuda en algunos casos, sin tener en cuenta aspectos que apoyen al profesor en la descripción de los OA, que hagan de esta una actividad motivadora e interesante para los docentes.

En la UCI, el Centro de Tecnologías para la Formación (FORTES) de la facultad 4, desarrolló la herramienta de autor web CRODA, que cuenta con aspectos importantes para el proceso de creación de un OA y su posterior reutilización, pues permite describirlo con metadatos rigiéndose por el estándar de

catalogación LOMv1.0, posibilitando una amplia caracterización para tales recursos, debido a la gran cantidad de elementos de información que este último brinda. Con la herramienta también se pretende en su versión 2.0; la creación no solo de OA sino también de UoL, con la posibilidad de catalogación con metadatos.

Sin embargo CRODA 1.0 presenta las siguientes deficiencias para la gestión de los metadatos:

- No permite definir elementos obligatorios según interés y características de las instituciones que lo utilicen para la creación de recursos.
- Pudieran existir un mayor número de elementos a autocompletar por parte del sistema en las instancias de metadatos, facilitando la actividad de catalogación para el autor.
- No cuenta con las funcionalidades para la exportación/importación y publicación de instancias de metadatos, imposibilitando su reutilización.
- No cuenta con las funcionalidades para gestionar los metadatos de las UOL que se crearán en la herramienta en su versión 2.0

A partir de lo anteriormente planteado se resume que en CRODA 1.0 no se tienen en cuenta aspectos que contribuyen a enriquecer la descripción de los OA y las UOL, de manera tal que esta actividad, de vital importancia para su posterior reutilización, se haga interesante y fácil para los docentes. El profesor necesita interfaces amigables y elementos de ayuda que lo guíen durante la inserción de los metadatos y se hace necesaria la optimización e incremento de las funcionalidades asociadas a la gestión de los metadatos de la herramienta.

Problema de investigación: ¿Cómo incorporar funcionalidades en la herramienta de autor CRODA que facilite la gestión de los metadatos en los OA y las unidades de aprendizaje con el propósito de favorecer a su reutilización?

Se precisa como **objeto de estudio** el proceso de gestión de metadatos.
El **objetivo general** de este trabajo es implementar un módulo para la gestión de metadatos en CRODA 2.0 a partir del esquema LOMv1.0.

El objetivo general se desglosa en los siguientes **objetivos específicos**:

- Elaborar el marco teórico de la investigación.
- Implementar las funcionalidades que conforman el módulo Metadatos en CRODA 2.0.
- Validar la solución propuesta.

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

Se plantea como **idea a defender** que la implementación de un módulo en CRODA 2.0 que permita definir aquellos elementos de información que se desea sean de llenado obligatorio, el autocompletamiento de los metadatos por parte del sistema, así como la exportación/importación y publicación de metadatos, favorecerá la descripción en CRODA de los OA y las UOL contribuyendo a su reutilización.

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

1. Selección del esquema de metadatos a emplear
2. Investigación sobre el proceso de gestión de metadatos en herramientas de autor existentes e investigaciones al respecto.
3. Desarrollo de los artefactos propuestos por la metodología seleccionada durante la implementación y prueba del módulo Metadatos.
4. Implementación de las funcionalidades que conforman el módulo Metadatos en CRODA 2.0.
5. Integración de las funcionalidades que conforman el módulo Metadatos en CRODA 2.0.
6. Comprobación de las funcionalidades que conforman el módulo Metadatos en CRODA 2.0.

Para dar cumplimiento a las tareas planteadas se hace necesario el empleo de los siguientes métodos teóricos. El **Histórico-Lógico** para estudiar el origen y evolución de los metadatos, los estándares y conceptos relacionados con los mismos, así como las herramientas que permiten describir los recursos con metadatos para evaluar posibles mejoras a incluir en la nuestra. El **Analítico-Sintético** para el análisis y la extracción de la información más relevante acerca de las metodologías, tecnologías y herramientas posibles a ser utilizadas en el desarrollo de este módulo, lo que permite seleccionar las características más acordes a nuestros objetivos. Además del método empírico **Observación** con el cual se toman las mejores prácticas de sistemas con la misma finalidad del que se pretende desarrollar así como sus vulnerabilidades para convertirlas en ventajas de este nuevo producto.

El presente trabajo se estructura en tres capítulos:

- **Capítulo 1:** Se abordan importantes temas relacionados con el e-learning. Se hace un estudio de los estándares LOM y SCORM, de los diferentes procesos para la creación de metadatos propuestos en disímiles proyectos e investigaciones actuales y se observa cómo se realiza en varias HA.

- **Capítulo 2:** Se detalla el proceso de implementación del módulo Metadatos. Se explican el modelo de implementación, los diagramas de paquetes así como los de componentes del sistema. Además se describen los modelos de datos, tanto el relacional como el XML (Extensible Markup Language). Incluye una explicación de las funcionalidades más críticas y complejas.

- **Capítulo 3:** Se reproduce el proceso de pruebas. Se desglosan las pruebas de caja negra a través de casos de prueba estructurados a partir de casos de uso. Se relacionan los errores detectados así como el proceso de depuración en tres iteraciones.

Capítulo 1. Fundamentación teórica

Introducción

En este capítulo se precisan los elementos teóricos necesarios para entender el funcionamiento del módulo Metadatos basado en el estándar LOMv1.0 a partir del estudio de los diferentes procesos para la creación de metadatos, propuestos en disímiles proyectos e investigaciones actuales, además, se observa cómo se realiza dicho proceso en algunas HA, aportando utilidades al módulo para su implementación. De igual manera se hace una revisión y selección de lenguajes de programación web, gestores de bases de datos y las metodologías de desarrollo de software propuestas para el cumplimiento de este trabajo.

1.1 El aprendizaje electrónico (e-learning)

En las sociedades del conocimiento las nuevas tecnologías de la información y las comunicaciones representan un papel fundamental. Como consecuencia de la aplicación de estas nuevas tecnologías en procesos de formación surge lo que se conoce como e-learning. Algunos autores lo definen como un *"conjunto de tecnologías, aplicaciones y servicios orientados a facilitar la enseñanza y el aprendizaje a través de Internet/Intranet, que facilitan el acceso a la información y la comunicación con otros participantes"* (6). Según Guzmán *"e-learning (electronic learning): Término que cubre un amplio grupo de aplicaciones y procesos, tales como aprendizaje basado en Web, aprendizaje basado en computadora, aulas virtuales y colaboración digital. Incluye entrega de contenidos vía Internet, intranet/extranet, audio y videograbaciones, transmisiones satelitales, TV interactiva, CD-ROM y más"* (3).

Mientras que otros prefieren una definición más abarcadora en la que se mencionan términos como plataformas tecnológicas y herramientas de comunicación síncrona y asíncrona en este caso se destaca el caso de Francisco García el cual define al e-learning como 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"* (7).

Como se puede observar en todos está presente el uso de las TIC para el proceso de enseñanza-aprendizaje, en el cual juega un papel importante la Internet/Intranet. Se pude decir además que el e-learning trae consigo una gran variedad de ventajas (8):

- 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.

1.1.1 Plataformas que interactúan en el entorno e-learning.

En la práctica, para llevar a cabo una formación basada en e-learning es necesario el uso de plataformas que permiten la comunicación e interacción entre los profesores, alumnos y contenidos. Principalmente existen 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 (3).
Las primeras son conocidas como Sistemas de Gestión de Aprendizaje, LMS por sus siglas en inglés, es un software basado en un servidor web que presenta módulos para los procesos administrativos y de seguimiento necesarios en sistemas de enseñanza-aprendizaje de forma simplificada. Estas plataformas permiten la realización de actividades tales como: seguimiento de cursos con evaluaciones, facilitan el aprendizaje colaborativo mediante el uso del chat, foros y video conferencias con los profesores y entre los propios estudiantes. Ejemplos de estas herramientas son Moodle[1], Blackboard[2] y Claroline[3].

Un Sistema de Administración de Contenidos de Aprendizaje o LCMS (Learning Content Management System) se define como un sistema basado en la web utilizado para crear, aprobar, publicar, administrar y

[1] Véase http://moodle.org.
[2] Véase http://www.blackboard.com.
[3] Véase http://www.claroline.net.

almacenar recursos educativos (como los OA) y cursos en línea. "*Los principales usuarios son los diseñadores instruccionales que utilizan los contenidos para armar los cursos, los profesores que utilizan los contenidos para complementar su material de clase e incluso los alumnos en algún momento pueden acceder a la herramienta para desarrollar sus tareas o completar sus conocimientos"* (3).

1.1.2 Herramientas de autor.

Existen distintos tipos de herramientas que apoyan al e-learning entre las que se encuentran los ROA y las herramientas de autor. *"Las herramientas de autor son aplicaciones que disminuyen el esfuerzo a realizar por los profesores, maestros, educadores, etc., ofreciéndoles indicios, guías, elementos predefinidos, ayudas y una interfaz amigable para crear materiales educativos y/o cursos en formato digital"* (9). *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. A continuación se presenta un estudio de algunas de las herramientas de autor más usadas.

Reload

Reload es una herramienta de autor gratuita y de código abierto, que posibilita 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 gratuitas y de códigos abiertos. También Reload tiene un empaquetador de contenidos mediante IMS Content Packaging 1.1.4, SCORM 1.2 y SCORM 2004 (10).

Con Reload es posible:

- Probar los contenidos empaquetados con el uso del reproductor de SCORM
- 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 formularios, vista de árbol y elementos de ayuda, sin embargo, aún solo 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

La herramienta de autor eXeLearning es de código abierto y se utiliza para la creación de recursos educativos. Esta herramienta tiene una interfaz amigable y permite la catalogación de metadatos con el estándar Dublin Core de una manera simple con elementos de ayuda y una interfaz basada en formularios. Es 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 o HTML, exporta contenido como páginas web rigiéndose por los estándares de contenido IMS y SCORM 1.2 (11). A pesar de estas ventajas, presenta deficiencias en cuanto a la gestión de metadatos, ya que permite el empaquetamiento con SCORM 1.2, el cual propone a LOM como estándar para la catalogación, sin embargo, en eXeLearning la descripción de recursos se hace con Dublin Core, por consecuencia se limita la descripción de la instancia de metadatos LOM.

Hot Potatoes:

La suite Hot Potatoes incluye seis aplicaciones que le permite crear varios tipos de ejercicios: de opción múltiple, de respuesta corta, crucigramas, frases confusas, ordena cronológico y llenar los espacios en blanco. Hot Potatoes es un programa gratuito pero no de código abierto. Permite el empaquetamiento con el estándar SOCORM 1.2. (12) En cuanto a la catalogación de los recursos Hot Potatoes permite añadir metadatos con el estándar Dublin Core, metadatos que posibilitan que los buscadores de la red puedan encontrarlo fácilmente. Al seleccionar la opción añadir metadatos se muestra un ventana en formato HTML con una ayuda que guía al autor.

Cuadernia

Se trata de una herramienta fácil y funcional que permite crear de forma dinámica eBooks o libros digitales en forma de cuadernos compuestos por contenidos multimedia y actividades educativas para aprender jugando de forma muy visual.

Se propone una interfaz muy sencilla de manejo, tanto para la creación de los cuadernos como para su visualización a través de Internet o desde casa. La apuesta es generar contenidos digitales de apoyo a la acción educativa, proporcionando un software divertido y ameno que ayudara a grandes y a pequeños a aprender jugando, con toda la potencia que nos ofrecen las nuevas tecnologías e Internet. Cuadernia se rige por el estándar de empaquetado SCORM para poder distribuirlos los recursos educativos y el estándar de catalogación LOM para potenciar las búsquedas y la reutilización de forma fácil de un determinado recurso a la hora de planificar una clase o una acción formativa concreta (13). A través de la herramienta Cuadernia Catalogación que presenta una interfaz amigable en forma de formulario pero que no presenta ayuda por cada elemento descriptivo del estándar LOM

Udutu

Udutu[4] es una herramienta de autor web que permite añadir contenidos en flash, audio y vídeo, así como documentos, imágenes, ejercicios de autoevaluación o sea la creación de cursos completamente multimedia 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 descripción por lo que se vería afectado su uso y reutilización.

Por las desventajas expuestas de las herramientas analizadas se llega a la conclusión que presenta dificultades para lograr la gestión de metadatos totalmente eficiente. Como se ha mencionado anteriormente poseen interfaces amigables basadas en formularios y elementos de ayuda en el llenado de los elementos de información. Y al ser estas características en la edición de metadatos, un factor de vital importancia, se pretenden tomar como guía para desarrollar en CRODA 2.0 una interfaz que permita la

[4] http://www.udutu.com

edición de metadatos con múltiples vistas que interactúen entre sí. Además brindar al usuario un sistema de ayuda para cada elemento de información

1.1.3 Objetos de aprendizaje.

Los OA surgieron producto del e-learning por la necesidad de compartir y reutilizar contenidos educativos. Debido a la diversidad de interpretaciones por los diferentes autores en diferentes contextos, el término OA no posee una única definición. La IEEE define a los OA como "*cualquier entidad, digital o no digital, que puede ser usada para el aprendizaje, la educación o la formación"* (14).

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 con información del OA, aunque es importante aclarar que no siempre es externa. *"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"* (15).

En este trabajo como se dijo anteriormente se toma como definición de OA. "*Recurso didáctico digital estandarizado, descrito por metadatos, compuesto por uno o varios objetos de información, que responde a un único objetivo de aprendizaje y puede ser utilizado en diversas situaciones de enseñanza-aprendizaje*"

Las definiciones anteriormente expuestas demuestran las diferencias y desacuerdos que existen referentes al término OA. Para algunos autores es un recurso completamente digital, mientras para otros no. Pero para todos es evidente un claro propósito educativo. Y algunos incluso hacen énfasis en la importancia que tiene la descripción con metadatos o su reutilización. Precisamente 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 lo hacen hoy una tecnología importante y necesaria dentro del proceso de enseñanza aprendizaje.

Principales componentes de un OA

Los OA están estructurados de diferentes formas. En una presentación la Dra. Lourdes Galeana divide al OA en tres partes:

- *Unidad de Información*: Contenidos multimedia individuales (texto, imágenes, audio, vídeo) en la que se tiene la posibilidad de generar contenido textual mediante el acceso a través de editores de texto.
- *Unidad de Contenido*: Define la ubicación en la que se encuentran albergados los contenidos, facilitando la generación de plantillas.
- *Unidad Didáctica*: Abarca cada uno de los elementos que permiten generar planteamientos de aprendizaje significativo, determinar criterios de evaluación, contenidos, recursos y actividades de enseñanza-aprendizaje.

La Universidad de las Ciencias Informáticas utiliza los componentes de los Objetos de Aprendizaje como se muestran en la (Figura 1): Objetos de información, descritos por metadatos y estructurados didácticamente de manera correcta (16).

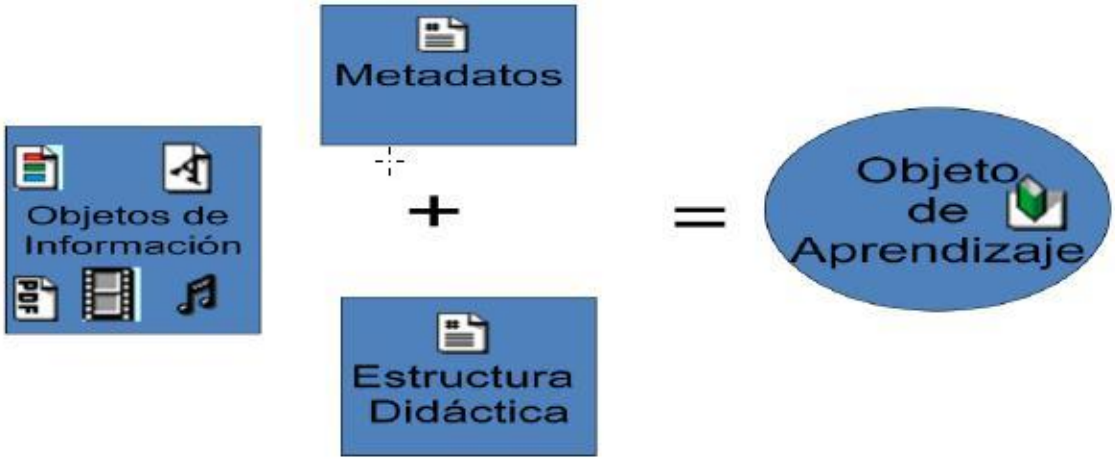

Figura 1: Componentes del Objeto de Aprendizaje utilizado en la UCI

Objetos de Información:

Está formado por activos de aprendizaje como vídeos, gráficos, animación, entre otros. Contiene toda la información que permita al usuario poder interactuar con el OA para dar cumplimiento a sus objetivos. Esta información sería de navegación simple como cuadros de búsqueda, barras de enlace y barras de herramientas, entre otras.

Metadatos:

Los que incluyen la información necesaria para catalogar el OA, tal como el título, autor, contenido, público objetivo, especificaciones técnicas y otras definidas según el estándar utilizado. Permitiendo así buscar rápidamente, organizar y actualizar los OA.

Estructura didáctica:

Es la organización del contenido educativo basada en una unidad de aprendizaje, definida pedagógicamente.

El autor de la presente investigación está de acuerdo con las partes de OA propuestas por la universidad, definida así en los proyectos CRODA y RHODA, permitiendo de esta manera una mejor organización en la información que se le brindará al usuario final.

1.1.4 Estándares en el e-learning.

La gestión de los OA es más eficiente si se utilizan esquemas estándares de metadatos. *"Un esquema estándar es un conjunto de elementos, que propone un grupo u organismo reconocido, para describir un determinado dominio o tipo de recursos"* (17). Al utilizar un estándar de metadatos es posible organizarlos, localizarlos, recuperarlos, procesarlos, evaluarlos, intercambiarlos y reutilizarlos, de manera tal que se garantice la interoperabilidad con otros sistemas que manejen esquemas de metadatos compatibles. Los estándares surgen por la necesidad de lograr la interoperabilidad entre las plataformas del e-learning, es decir lograr la comunicación e integración, aun siendo plataformas diferentes. Diseñar estándares multi-entornos, es un reto difícil que hace solo poco años comenzó a desarrollarse, pero que cuenta en la actualidad con la existencia de organizaciones y proyectos enfocados a ese fin, para aprovechar las ventajas que ofrecen las plataformas del e-learning integradas entre sí. A continuación se presenta una explicación de algunos de estos grupos y proyectos según Guzmán (3):

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[5].

Advanced Distribuited 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[6].

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

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[7].

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

El 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 o por sus siglas en inglés LTSC, que se encarga de desarrollar estándares técnicos, recomendaciones y guías para la tecnología educativa8.

[5] Disponible en http://www.imsproject.org
[6] Disponible en http://www.adlnet.org
[7] Disponible en http://www.ariadne-eu.org
[8] Disponible en http://www.ieee.org

1.1.4.1 Estándares para el empaquetamiento de recursos.

Los estándares de empaquetamiento son aquellos que permiten el intercambio de materiales entre las herramientas y sistemas, por ejemplo, una HA, una biblioteca digital de recursos educativos o un LMS, capaces de interpretar los paquetes, independientemente de la forma y el lugar de origen de dichos paquetes, deben soportar los mismos estándares. De la diversidad de estándares creados para el empaquetamiento por distintas organizaciones, se analizarán en este trabajo 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 Content Packaging (Instructional Managament System Content Packaging) es una variante de estándar para el empaquetamiento de contenidos promovida por IMS Global Learning Consortium. IMS Content Packaging 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 cuál página HTML mostrar primero, este estándar propone el uso del esquema LOM para la descripción de los recursos (18).

SCORM

El estándar Sharable Content Object Referente Model, surge cuando el Departamento de Defensa de Estados Unidos junto al proyecto ADL, con el fin de lograr una forma común de identificar, definir y comunicarse a todos los recursos involucrados en un sistema e-learning, desarrollan 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 (5). Para la descripción de los recursos con metadatos, SCORM propone al estándar LOM y define qué elementos de información son obligatorios en los metadatos del mismo, para marcar los componentes descritos. Con el fin de incorporarle a los recursos las cualidades necesarias para simplificar su uso y gestión. Lo que se busca mediante esta información complementaria es poder saber cuál es el contenido y el propósito de un OA sin tener que acceder a dicho contenido. Por tanto, los metadatos aportan información orientada a hacer más eficiente la búsqueda y utilización de los recursos (19).

1.1.4.2 Estándares para describir recursos.

La necesidad de localizar, clasificar y reutilizar 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[9], Canadian Core Learning Resource Metadata Application Profile (Cancore)[10], LOM-ES[11] y LOM[12]. Seguidamente se estudian LOM y Dublin Core.

Dublin Core

Esquema de metadatos elaborado por la DCMI (Dublin Core Metadata Initiative)13, enfocado al ámbito bibliotecario, no es tan amplio como LOM en cuanto a posesión de elementos de información, puesto que solo posee quince (Contributor, Coverage, Creator, Date, Description, Format, Identifier, Language, Publisher, Relation, Rights, Subject, Source, Title y Type), mientras que LOM cuenta con sesenta y cuatro. Aun así por su simplicidad Dublin Core sigue siendo uno de los estándares con mayor difusión a nivel

[9] Disponible en http://dublincore.org

[10] Disponible en http://www.cancore.ca

[11] Una adaptación de LOM creado y utilizado por la comunidad educativa de España.

[12] Para consultar los 64 elementos propuestos por LOM www.uvs.sld.cu/archivos/lomv1spanish.rar/

[13] 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.

mundial. Los elementos son opcionales, pueden repetirse y 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 desde el punto de vista educacional.

Learning Object Metadata (LOM)

El IEEE junto al LTSC, desarrolló en el año 2002 el estándar LOM 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 (14).

Los esquemas para la descripción de recursos, proponen un conjunto de elementos de información. En el caso de LOM, como se ha mencionado anteriormente posee sesenta y cuatro elementos de información, un número elevado 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 (14):

- 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 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

Muchas investigaciones afirman que los sesenta y cuatro elementos para los metadatos que proporciona LTSC 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 (20).

En el modelo de metadatos descrito en el CAM de SCORM 1.2, se definen elementos obligatorios para cada componente de un OA. Los elementos de obligatoriedad, son los que se deben describir obligatoriamente para cada componente de un OA, estos se definen para reducir 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 que los almacene. 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 (20). Resultan interesantes la preguntas ¿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 SCORM 1.2 propone a la categoría Uso Educativo como opcional y muchas veces es la de más interés para los profesores. Por tanto 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 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 (21). 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, en el repositorio RHODA que se emplea en la UCI para almacenar los OA creados en CRODA, uno de los criterios de búsquedas que se utiliza está relacionado con la categoría Uso Educativo, sin embargo esta no se considera 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.

1.1.5 Los metadatos y los procesos para su creación.

Los metadatos han tenido su surgimiento desde la antigüedad, cuando los bibliotecarios describían los libros manualmente. En la siguiente definición de metadatos se hace mención a términos de la actualidad tales como *servicios* y *aplicaciones* "*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 ello".* (22). En otras publicaciones se hace referencia a los beneficios que aportan los metadatos (23) :

- 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.

La informática como ciencia, tiene como objetivo 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 (17).

También el equipo perteneciente al proyecto JORUM, desarrolla un flujo de trabajo para la creación de metadatos en el que primeramente se autocompletan un grupo de elementos de información y 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 educacionales 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 (24).

Otros proyectos como ADL, no proponen exactamente un flujo para la creación de metadatos, en cambio, desarrollaron una guía para la creación de contenidos reutilizables con SCORM 2004 (25), 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, el cual 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.

La reutilización de metadatos es otra variante para la descripción de recursos, se plantea que hablar de reutilización en el contexto e-learning, 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 catalogación y a lenguajes como XML (26). La reutilización de los metadatos es una funcionalidad que puede ser empleada para facilitar la actividad del llenado de estos, con la observación de ejemplos del esquema LOM en codificación en XML.

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

1.2 Herramientas y tecnologías a utilizar.

Atendiendo a las políticas de desarrollo del centro FORTES en su actual migración al software libre. Se hace necesario el análisis de las herramientas empleadas en el desarrollo de CRODAv1.0. De manera tal que el módulo presentado puede ser incorporado a la herramienta de autor CRODA en su versión 2.0 y otras herramientas o sistemas similares que necesiten la descripción con metadatos.

1.2.1 Metodología 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" (27). Las metodologías de desarrollo de software, en la ingeniería del software actual, representan la base y la guía, pues dependiendo de las características propias de cada una de estas, abarcan todo un conjunto de actividades necesarias en un orden correcto para controlar un proceso de desarrollo de software desde inicio a fin.

Por la dificultad asociada al 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. En el mundo de la ingeniería informática, la creación de un software o parte de este, es un proceso riguroso. Donde la metodología juega un papel fundamental.

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 (28):
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 (27):
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

Es de gran importancia la creación de modelos que interrelacionen la vida real y el sistema a construir en la construcción de un software. Los modelos se componen de documentos que describen cosas, diagramas y otros modelos o artefactos. En este aspecto es importante mencionar UML 2.0, 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. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos. Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas (29).

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

Debido a que RUP basa su éxito en mantener una documentación ordenada a lo largo del desarrollo del software, lo que constituye una ventaja al pensar en el posterior mantenimiento del sistema. Teniendo en cuenta las particularidades del equipo de trabajo del proyecto, el cual está conformado mayoritariamente por estudiantes, que necesitan consultar una documentación como base, cuando el equipo se disuelva y sea necesario realizar un cambio o mejora. A pesar de estar recomendada para proyectos extensos y grandes grupos de trabajo, es la metodología más adecuada para dirigir el proceso de desarrollo del software. Debido a que define claramente actividades realizadas por roles generando a su paso artefactos que sustentan el proceso de desarrollo del producto. 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 trabajo.

1.2.2 Herramienta Case.

Una herramienta CASE es aquella que permite crear los artefactos necesarios siguiendo una metodología en la construcción de un software. De vital importancia para la metodología RUP, en las fases de Análisis y Diseño e Implementación, donde, se generan la mayor cantidad de artefactos, a los cuales, es muy fácil

realizarles los cambios que ocurren en el diseño de un software precisamente por ser modelados con estas herramientas, además el software se hace con mayor calidad y rapidez. A continuación se describen dos de las herramientas CASE más empleadas en el mundo.

Visual Paradigm 6.4 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 elaborar 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 2.0, 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 (30). Es una herramienta multi-plataforma como: Windows, Linux y Unix, además es muy fácil de usar. Teniendo en cuenta sus principales ventajas se decide la selección de Visual Paradigm como herramienta CASE en el presente trabajo.

1.2.3 Tecnologías de implementación.

Específicamente para la implementación del módulo presentado teniendo en cuenta las características de las tecnologías y herramientas necesarias, en este caso, framework de desarrollo, lenguajes y otras tecnologías relacionadas con el código a implementar.

Framework de desarrollo.

Los marcos de trabajo o frameworks de desarrollo surgieron con la necesidad de facilitar y optimizar la creación de aplicaciones web. Dentro de los frameworks más usados en la actualidad se encuentran Symfony. A continuación se describen sus principales características.

Symfony 1.4.17

Según Fabien Potencier 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 (31):

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y Unix estándares).
- Independiente del sistema gestor de bases de datos, soportados por PDO.
- 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 con 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 desarrollo del módulo.

ExtJs 3.4.0

ExtJs es una librería de JavaScript que permite el desarrollo de aplicaciones web interactivas empleando tecnologías como Ajax, 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 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 checkboxs, editor HTML, árbol de datos, pestañas, paneles divisibles en secciones, etc. (35).

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 todas las funcionalidades, componentes y eventos que presenta ExtJs y por haber sido empleada en la interfaz de usuario para CRODA 1.0, se propone entonces para la futura implementación del módulo Metadatos.

1.2.3.1 Lenguajes de implementación.

XML

XML 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"* (32). XML no posee etiquetas definidas desde un principio, por lo que el autor las define según las necesite. Se propone el uso de dicho lenguaje debido a que los metadatos bajo el estándar LOM se codifican con el mismo, codificación que se rige por un grupo de reglas que especifican 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 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 y otros medios soportados. Este lenguaje se propone para la construcción de las vistas del módulo presentado. (33)

CSS (Cascading Style Sheets) 2.0

CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito casi siempre en HTML. 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. (33)

JavaScript 1.5

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

"Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios" (34).

Dado que JavaScript es un lenguaje de programación del lado del cliente, ampliamente utilizado en el mundo del desarrollo web, por ser muy versátil y compatible con todos los navegadores modernos, es muy importante el empleo del mismo para la gestión de la interfaz cliente/servidor ya que posibilita la interacción de los usuarios con la aplicación. Además a pesar de ser un lenguaje sencillo, tiene en cuenta varios organismos internacionales de normalización.

PHP 5.3.5

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. Posee una amplia documentación en su página oficial, donde todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda. Es libre, por lo que se presenta como una alternativa de fácil acceso para todos. No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución. Además es el empleado actualmente por el equipo de CRODA.

1.2.3.2 Sistema gestor de bases 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 encuentra PostgreSQL para el almacenamiento relacional y eXist-db que permite el almacenamiento de archivos en formato XML.

PostgreSQL 8.4

PostgreSQL es un potente servidor de base de datos relacional orientado a objetos, 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 (36): 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, subselecciones, 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 1.4

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, es necesario en la implementación del módulo el empleo del gestor de base de datos 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 (37). 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.

1.2.4Servidor 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 en el mundo del software libre se encuentra Apache.

Apache 2.2.17

Apache es un servidor web HTTP robusto cuya implementación se realiza de forma colaborativa y precede de la licencia BSD. 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 y el nivel de respuesta. Además Apache es modular, de código abierto, multiplataforma,

popular (fácil conseguir ayuda/soporte) (38). 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 puede resumir que en CRODA 1.0 no se tienen en cuenta aspectos que contribuyen a enriquecer la descripción de los OA y las UOL. El profesor necesita elementos de ayuda que lo guíen, así como interfaces amigables durante la inserción de los metadatos. En el capítulo se expusieron los principales conceptos y tecnologías en el e-learning, como la tecnología OA y los estándares para catalogarlos. Además quedó seleccionado para la implementación de este trabajo el esquema de metadatos LOMv1.0 y los procedimientos a tomar en cuenta en su creación, conjuntamente con 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. También se seleccionaron, previo un análisis, las tecnologías y herramientas para el desarrollo de los componentes.

Capítulo 2. Implementación

Introducción

En este capítulo se describe la estructura del sistema y la manera en que se agrupan sus elementos fundamentales mediante el empleo de los diagramas de paquetes para su representación más general. Y para la descripción de las funcionalidades más complejas de manera más exquisita se utilizan los diagramas de componentes que evidencian las relaciones entre sus scripts y archivos php.

2.1 Modelo de implementación

El modelo de implementación describe cómo los elementos del modelo de diseño se implementan en términos de componentes, cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados y cómo dependen los componentes unos de otros (39).

2.2 Diagrama de paquetes

En la (*Figura 2*) se observa un diagrama de paquetes general que esboza los dos grandes conjuntos lógicos en los que se divide el sistema: la parte relacionada con la librería ExtJs 3.4.0 y la parte que se asocia al framework Symfony en su versión 1.4.

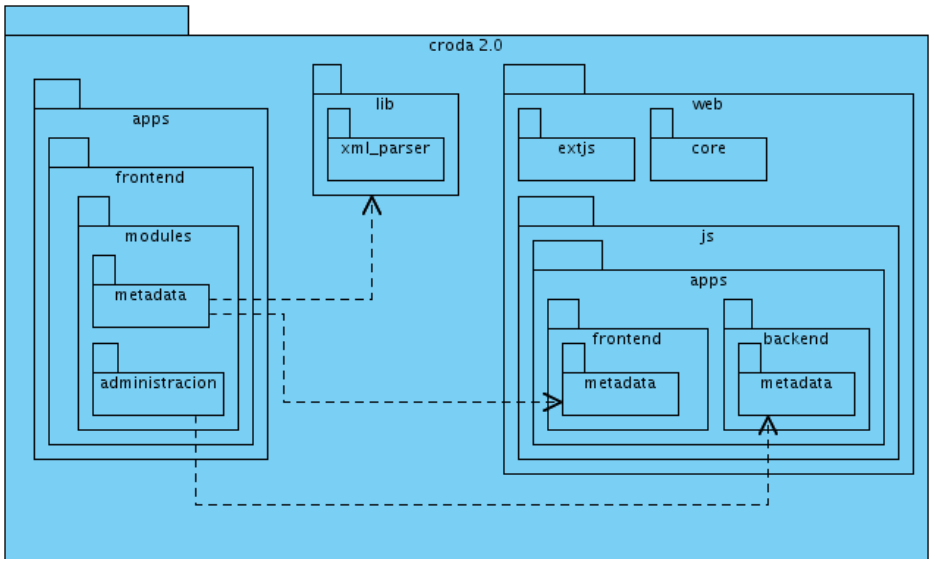

Figura 2: Diagrama General del Sistema

Donde los archivos referentes a los componentes de extendidos de ExtJs seguirán las estructura aplicada a los archivos de Symfony en sus versiones recientes siguiendo el estándar que se desea emplear en CRODA 2.0.

Para lograr dicho propósito fue necesario la creación de nuevos scripts que se integraron al resto de las dependencias de la librería siguiendo el patrón del framework. Estos archivos se agrupan en las carpetas apps, extjs y core, en esta última se almacenan los scripts core.js y module.js, encargados de hacer funcionar las librerías ExtJs como si se tratase del propio Symfony sobre sus diferentes módulos.

Esta forma de organización permitió tratar al conjunto de scripts referentes a un módulo de Symfony como módulos independientes dentro de ExtJs, dentro de la carpeta apps. Es por ello que el conjunto de funciones, scripts y componentes relacionados con el módulo Metadatos se encuentran dentro de esta localización en la carpeta metadata, siguiendo patrones de prefijos para denotar la funcionalidad de los mismos.

2.3 Diagramas de Componentes

El flujo de información en la aplicación web comienza en el *index.php* del framework Symfony. El mismo redirecciona sobre el *actions.class.php* del módulo Main (*Ver Figura 3*).

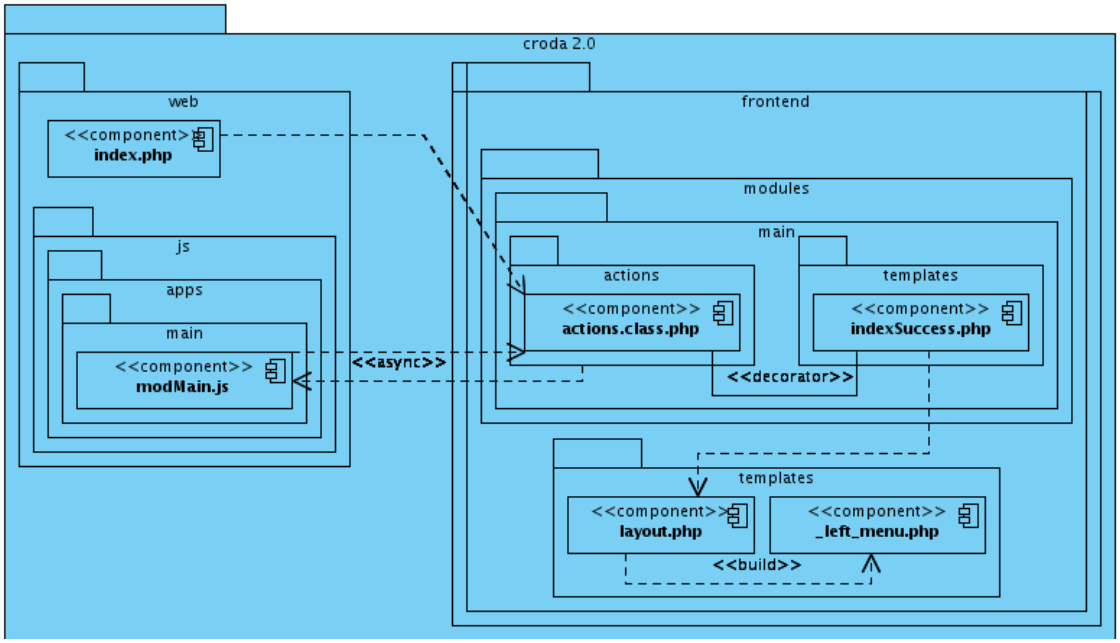

Figura 3: Diagrama de Paquetes. Integración de Symfony 1.4 con EXT JS 3.4.0

Este *actions* tiene asociado una plantilla (*template*) que a través de un esquema (*layout*) define los comportamientos comunes a todas las páginas del proyecto. Este procedimiento se denomina Decorador (31).

El *layout.php* se encarga de dar paso a un parcial (*partial*) denominado: *_left_menu.php.* Este parcial se ocupa del lateral izquierdo de la página de inicio y de hacer la petición al script *modMain.js* de la librería Ext JS, el cual maneja el área de los componentes visuales en dicha librería (*fronted*).

En el *modMain.js* se encuentra la función *buildTreeCreation* que al ser invocada envía una petición asincrónica *(async)* al *action* del módulo Main y recibe lo que debe graficar en el árbol de opciones que despliega. A partir del *script modMain* se interrelaciona con otros módulos, como learningDesign y content.

2.3.1 Integración del módulo Metadatos con learning- Design o Content

Para hacer uso del módulo Metadatos es necesario crear una instancia del componente *modMetadata.js* que con el control de otros componentes como *cmpFullView.js, cmpMetadataTree.js y cmpMetadataTabs.js* permite hacer una gestión amigable a los metadatos. Para la creación de una

instancia de dicho módulo son necesarios los parámetros **path** (ubicación de los metadatos en la BD eXist), el tipo de **recurso** que se describirá (OA, UoL), el **id** asociado a este recurso y el **objeto JSON** donde se guardará el modelo metadatos. Este último parámetro es opcional, se usa cuando el recurso aún no existe en la BD eXist *js (Ver* Figura 6*).*

Haciendo uso de estos parámetros el componente *modMetadata.js* se comunica de forma asincrónica con el componente actions.class.php donde se lleva a cabo la gestión de todas las funcionalidades disponibles en el módulo tales como modificar, guardar e importar/exportar instancias de metadatos.

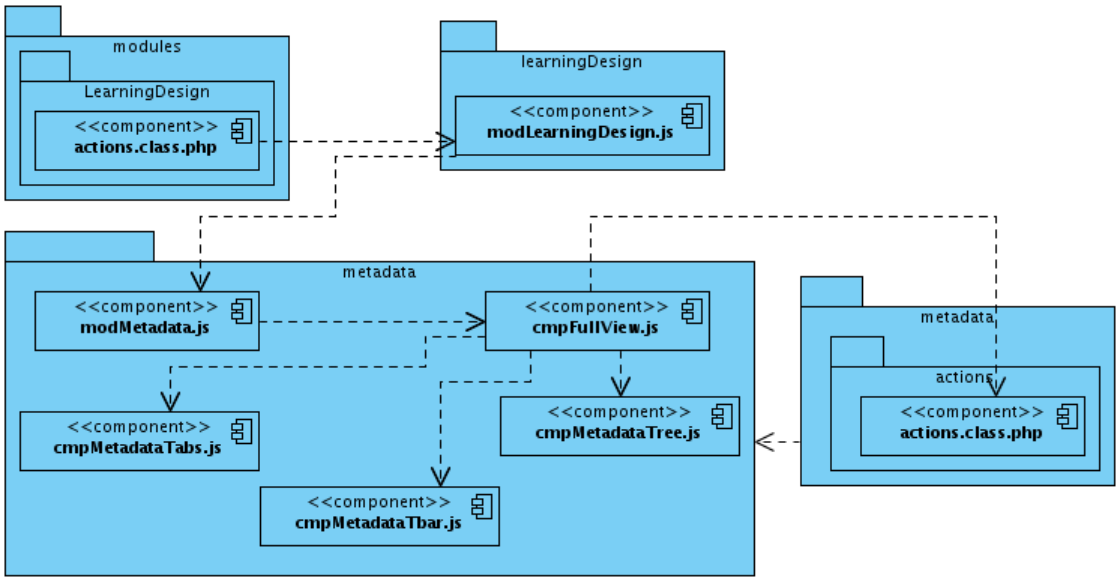

Figura 4: Diagrama Integración del módulo Metadatos con learningDesign o Content

Cuando se hace instancia del módulo Metadatos, este controla qué formulario mostrar en dependencia de las necesidades del usuario. La vista principal del módulo se crea a través de los siguientes scripts: *cmpFullView.js, cmpMetadataTree.js y cmpMetadataTabs.js.*

El componente cmpFullView.js es el que invoca el resto de los scripts e inicializa las diferentes estructuras para ir completando los metadatos mediante sus funciones, mostradas según los permisos asignados para cada usuario a través del componente cmpMetadataTbar.js *(Ver* Figura 5*).* Además se comunica con las

base de datos XML y PostgresSQL a través de la clase actions.class.php del módulo de Metadatos en Symfony.

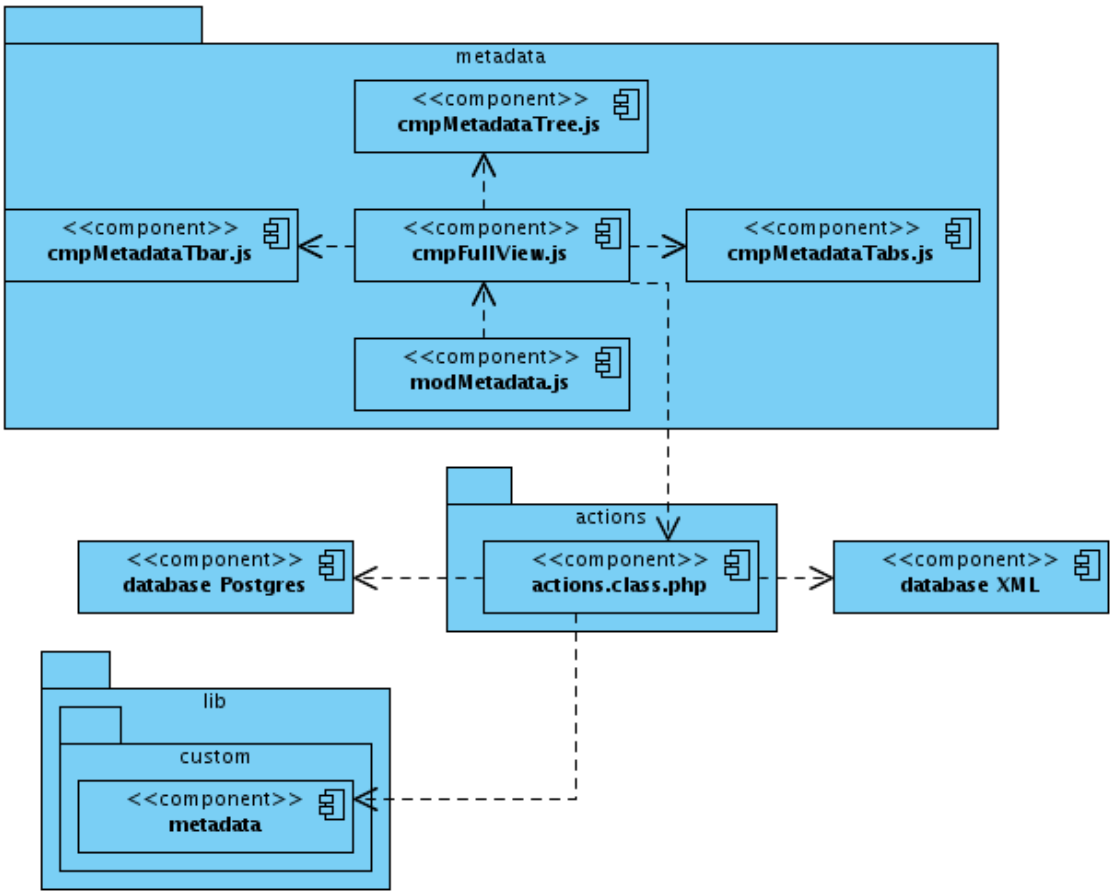

Figura 5: Diagrama de Componentes. Vista Principal.

Los componentes cmpMetadataTabs.js y cmpMetadataTree.js son los encargados de mostrar y gestionar la instancia de metadatos en cuestión en forma de árbol y formulario respectivamente garantizando sincronización entre ellos.

2.3.2 Funcionalidades generales

- **Convertir esquema XML a JSON**

En funcionalidades como el importar y editar instancias de metadatos se hace necesario convertir el **XML** que representa las instancias de metadatos a otro formato que la aplicación pueda interpretar con mayor facilidad a la hora de visualizar y gestionar en la interfaz web del módulo. Por tal polémica fue necesaria la creación de un componente que se encargara de esta tarea, de manera tal que fuera un proceso transparente para los componentes **JavaScript**. El componente puede ser utilizado desde cualquier módulo para llevar a cabo la conversión ya sea de **XML** a **JSON** o viceversa. A continuación se hace una presentación de la interrelación con el módulo Metadatos. (Ver *Figura 6*).

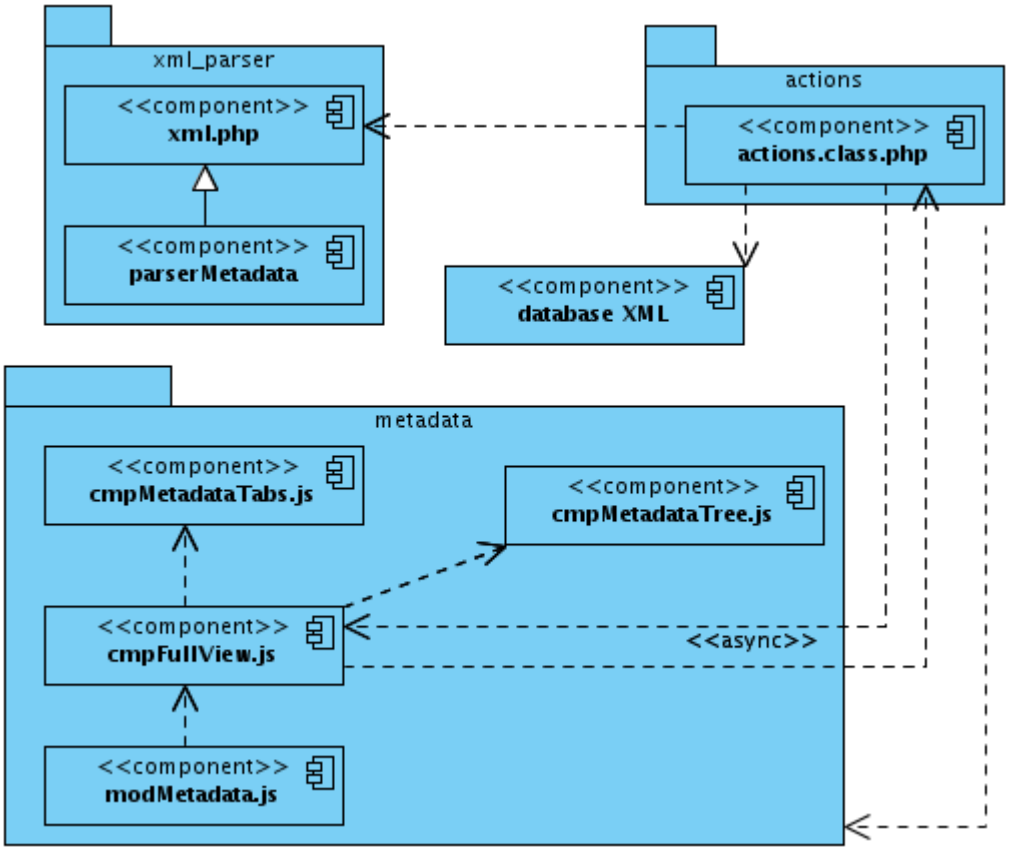

Figura 6: Relación entre el parserMetadata y el módulo Metadatos.

- **Exportar**

Partiendo de *cmpFullView.js* se hace una petición de tipo POST al *actions.class.php* del módulo Metadatos pasando por parámetro la instancia de metadatos en cuestión, allí se procesa a través del componente MetadataController.php donde se crea un archivo XML temporal listo para exportar y forzar la

descarga mediante PHP. Como resultado se obtiene una instancia de metadatos validada, lista para ser importada y por consecuente reutilizada tanto por la herramienta CRODA 2.0 como por cualquier otra herramienta utilizada para la gestión de los metadatos regida por el estándar LOMv1.0 (Ver *Figura 7*)

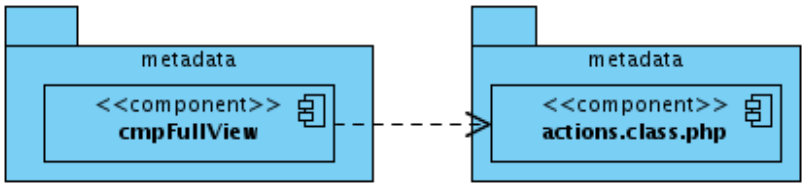

Figura 7: Diagrama de Componentes. Funcionalidad Exportar.

- **Importar**

El componente *cmpFullView.js* haciendo uso del componente *cmpMetadataImport.js* permite seleccionar un XML de metadatos bajo el estándar IEEE-LOM o IMSMD para el caso de los paquetes bajo SCORM 1.2. Luego se envían mediante el método POST al *actions.class.php* del módulo Metadatos que su vez hace uso del componente *MetadataController.php* donde se valida con el esquema XML correspondiente y de ser válido es transformado a formato JSON y enviado al componente de la vista *cmpFullView.js* que se actualiza con las nuevas propiedades y valores. (Ver *Figura 8*)

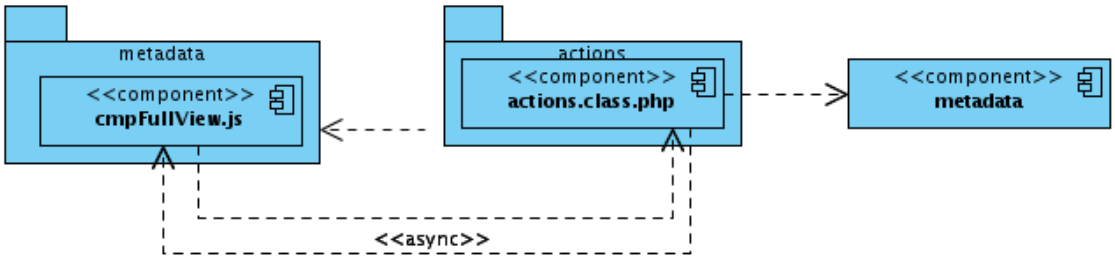

Figura 8: Diagrama de Componentes. Funcionalidad Importar.

- **Guardar**

Con el componente cmpFullView.js como base se ejecuta una petición POST con destino a la acción *SaveMetadata* al *actions.class.php* del módulo Metadatos el cual por su parte haciendo uso del componente *metadata* convierte el JSON entrante en XML y luego lo valida con el schema XML

correspondiente *(Ver Figura 9).* Luego si el proceso anterior fue satisfactorio se almacena en la base de datos XML dentro del archivo y recurso seleccionado a través del **path** pasado por parámetros.

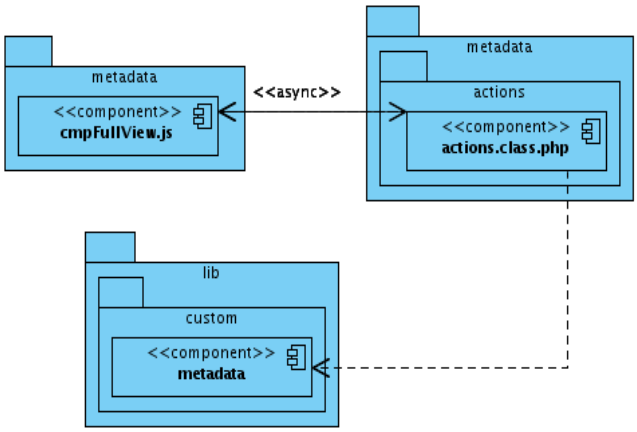

Figura 9: Diagrama de Componentes. Funcionalidad Guardar.

2.4 Modelo de datos

En CRODA los datos se almacenan en dos bases de datos diferentes. Una de ellas es la base de datos libre postgreSQL y la otra es una base de datos nativa (XML).

2.4.1 Modelo de base de datos relacional

La base de datos postgreSQL de CRODA cuenta con cuarenta y dos tablas de las cuales, el módulo de Metadatos emplea tres: *tb_pif, tb_metadata y sf_guard_user (Ver Figura 10).*

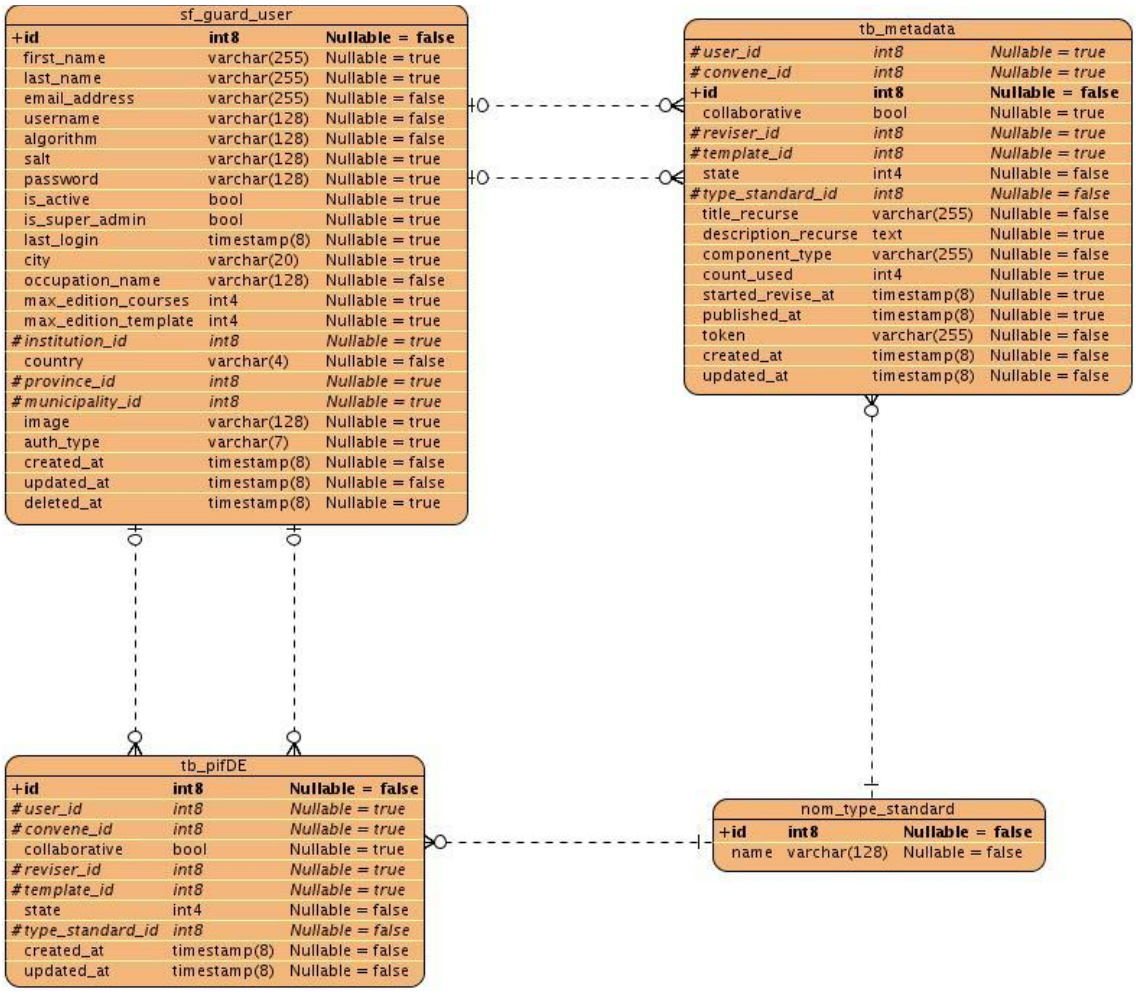

Figura 10: Base de datos postgreSQL. Tablas.

2.4.2 Descripción de las tablas de la base de datos postgreSQL

Nombre:	sf_Guard_User.
Descripción:	Tabla referente a los usuarios de la aplicación, la cual contiene los datos generales y específicos de los mismos, nombre de usuario, contraseña, nombre y apellidos, rol que ocupará en la aplicación (creador, generador), el tipo de acceso a la herramienta, a través de LDAP o manual, así como el identificador del propio usuario.

Atributo:	Tipo:	Descripción:
Id	Integer.	Identificador del usuario.
first_name	varchar	Nombre real del usuario.
last_name	varchar	Apellido del usuario.
email_address	varchar	Dirección de correo del usuario.
username	varchar	Nombre del usuario en la herramienta
algorithm	varchar	Algoritmo usado para encriptar las contraseñas de los usuarios
salt	varchar	Texto aleatorio concatenado al algoritmo de encriptación.
password	varchar	Contraseña del usuario.
Is_active	bool	Si está activo el usuario en la herramienta.
Is_super_admin	bool	Si es súper administrador de la herramienta.
last_login	timestamp	Último día que tuvo acceso a la herramienta.
city	varchar	Cuidad de origen.
occupation_name	varchar	Identificador del rol.
max_edition_courses	int.	Máximo de cursos en edición.
max_edition_template	int.	Máximo de plantillas en edición.
institution_id	Integer.	Identificador de la institución.
country	Integer.	País.
province_id	Integer.	Identificador de la provincia.
municipality_id	Integer.	Identificador del municipio.
Image	Varchar.	Imagen.

Tabla 1: Tabla sf_Guard_User de la BD PostgresSQL.

Nombre:	tb_pif.
Descripción:	Tabla referente a los OA generados en la aplicación, la cual contiene los datos generales y específicos de los mismos,

	nombre del OA, autor, fecha de creado, fecha de culminación, estado del OA, así como el identificador del mismo.	
Atributo:	**Tipo:**	**Descripción:**
Id.	Integer.	Identificador.
user_id.	Integer.	Identificador del usuario.
reviser_id.	Integer.	Identificador del revisor.
template_id.	Integer.	Identificador de la plantilla.
state.	Integer.	Estado.
type_standard_id.	Integer.	Identificador del tipo de estándar.
created_at.	Timestamp.	Fecha de creación.
Updated_at.	Timestamp.	Fecha de actualización.

Tabla 2: Tabla tb_pif de la BD PostgresSQL.

Nombre:	tb_metadata	
Descripción:	Tabla referente a los metadatos generados en la aplicación, que contiene los datos generales y específicos: título y descripción del OA o la UOL al que se asocian, el tipo del componente que describen dentro del recurso al que pertenecen, la fecha en que se publicó la instancia de metadatos y la cantidad de veces que fueron usados.	
Atributo:	**Tipo:**	**Descripción:**
Id	Integer.	Identificador.
user_id	Integer.	Identificador del usuario.
title_recurse	Varchar.	Título del OA o la UoL.
description_recurse	Varchar.	Descripción del OA o la UoL al que pertenece.

Comoponent_type	Varchar	Tipo de recurso al que pertenece OA o UoL.
token	Varchar	Token único para cada instancia de metadatos.
published_at	Timestamp.	Fecha de publicación.
count_used	Integer.	Cantidad de veces usado.

Tabla 3: Tabla tb_metadata de la BD PostgresSQL

2.4.3 Modelo de datos XML

La diferencia fundamental entre una base de datos nativa XML y una relacional es que la primera define un modelo lógico en comparación con el relacional de la última.

Los XML que se encuentran estructurados pero los datos específicos aún no han sido almacenados en ellos son denominados “manifiesto” en estos sistemas. Constituyen la declaración de la estructura lógica que guiará el almacenamiento de información etiquetada en ellos.

Los datos XML son almacenados en colecciones para su correcto indexado y ganar en organización. La colección más general que contiene a las demás se denomina *croda (Ver Figura 11)*. En ella se almacenan las *UoLs* y p*ifs* que se correlacionan con sus respectivas tablas de la base de datos postgreSQL. Dentro de *pifs* se almacena el manifiesto de cada OA. Dentro de *uol* se almacena el manifiesto de la especificación IMS – LD y en metadata el XML asociado a los metadatos publicados y en revisión. Cada manifiesto se construye según la siguiente estructura: ***imsmanifest***_ *<id del OA o la UOL>.xml*.

Figura 11: Esquema de las colecciones de la base de datos nativa Exist DB.

2.5 Diagrama de despliegue

En el desarrollo del módulo Metadatos como se dijo anteriormente se tienen dos tipos de base de datos correlacionadas. Una de tipo relacional y la otra nativa para el almacenamiento de los metadatos en los manifiestos XML creados por la aplicación CRODA 2.0. La base de datos relacional postgresSQL, por su parte es la encargada de mantener el control de los recursos creados. (Ver *Figura 12*)

En el módulo Metadatos el usuario accede al sistema a través de una PC cliente que establece una conexión HTTP con el servidor web Apache. El servidor web se comunica con el servidor de base de datos postgresSQL mediante el protocolo ADO del recuso creado, al mismo tiempo abre una conexión con el servidor nativo de XML (Exist DB) aprovechando el protocolo SOAP para almacenar los manifiestos de dichos recursos.

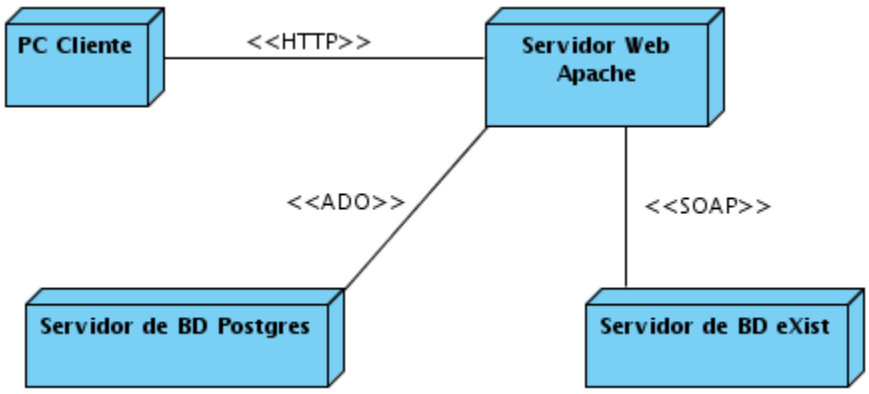

Figura 12: Diagrama de despliegue del módulo de Metadatos.

Conclusiones

Con el desarrollo de capítulo se logró integrar de forma satisfactoria el módulo Metadatos desarrollado utilizando la librería JavaScript ExtJS al sistema de funcionamiento de CRODA alojado en el framework Symfony. Siendo necesario un componente php que permitirá la conversión de forma transparente para JavaScript los XML en JSON. Se generaron además los diferentes artefactos y modelos necesarios para describir el comportamiento de la herramienta y su gestión de las bases de datos, tanto postgreSQL como XML.

Capítulo 3: Validación de la solución propuesta

Introducción

Las pruebas son de vital importancia en el desarrollo de software debido a que propician la obtención de buenos resultados. Se realizan con el objetivo de revisar que el software tenga el nivel de calidad requerido. Permiten detectar errores durante su funcionamiento, probando así la entrada y salida correcta de datos, tomando las especificaciones como referencia de comportamiento de la aplicación (40). Este proceso debe comenzar en la fase de requerimientos y terminar con la finalización de la aplicación. En el proceso de pruebas se definen varios métodos, técnicas y tipos de pruebas, las cuales se abordarán a continuación.

3.1 Métodos de validación

Actualmente existen métodos con diferentes enfoques que posibilitan la validación de software. Para la presente investigación se utilizan las pruebas de seguridad y de funcionalidad

3.1.1 Pruebas de seguridad

Las pruebas de seguridad se utilizaron con el objetivo de identificar fallas en los mecanismos de protección establecidos, basándose en las acciones autorizadas por los distintos roles de usuario.

En la realización de las pruebas de seguridad se asignaron a los probadores los roles de diseñador Instruccional y revisor diseño instruccional (bibliotecario de contenidos). Intentaron acceder a funcionalidades que no contaban con los privilegios necesarios y algunos lograron obtener información del sistema a través de la URL. Después de detectar este error se procedió a erradicarlo, comprobando los permisos establecidos para realizar las acciones

3.1.2 Pruebas de funcionalidad

La realización de pruebas es un elemento crítico para garantizar la calidad del software. Es una actividad en la cual un sistema o componente es ejecutado bajo condiciones o requerimientos específicos.

Las pruebas se aplican en las diferentes fases del desarrollo del software, con el objetivo limitar gradualmente los posibles errores, a medida que el desarrollo avanza. En este acápite se resaltan los dos grupos en que se dividen: las pruebas de caja blanca y las pruebas de caja negra.

Las pruebas de Caja Blanca: Se nombran de esta forma porque prueban la parte interna del software, específicamente el código fuente. Se comprueban los caminos lógicos del sistema generando casos de prueba que ejerciten las estructuras condicionales y los bucles. Existen varios métodos que analizan diferentes partes del software y se complementan entre sí para garantizar la calidad del sistema.

Las Pruebas de Caja Negra: Se basan en los requerimientos funcionales del sistema y se llevan a cabo desde el exterior de la aplicación. Estas pruebas permiten medir en que grado se cumplen los requerimientos solicitados por el cliente y se aplican sobre la interfaz del software observando las respuestas del sistema antes determinadas acciones. Son realizadas mediante casos de prueba con el objetivo principal de demostrar que las funcionalidades son correctas.

En la validación de la solución propuesta se utilizaron las pruebas de caja negra, donde se aplicó el método de partición equivalente, que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de pruebas. Este método está orientado a los requisitos funcionales del software y permite la creación de juegos de datos para probar a profundidad todas las funcionalidades del software e identificar los diseños de casos de prueba (41).

3.2 Prueba de Caja Negra

Las pruebas de **caja negra** consisten en ir probando uno a uno los diferentes módulos que constituyen una aplicación. Se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. O sea, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce el resultado esperado, así como que la integridad de la información externa se mantiene. Esta prueba examina algunos aspectos del modelo, fundamentalmente del sistema, sin tener mucho en cuenta la estructura interna del software. Se centran principalmente en los requisitos funcionales del software. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos.

3.2.1 Casos de Prueba

Diseño de casos de prueba basado en CU Establecer metadatos obligatorios para las UoL

Descripción general

El caso de uso se inicia cuando el bibliotecario de contenido decide establecer un conjunto de metadatos como obligatorios para las UoL 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 las UoL creadas en CRODA.

Condiciones de ejecución

El bibliotecario de contenidos debe estar autenticado en el sistema.

Secciones a probar en el Caso de Uso

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Establecer metadatos obligatorios para las UOL.	EC 1.1: Establecer metadatos obligatorios para las UOL	El sistema muestra todos los elementos del esquema LOM y permite al bibliotecario escoger los que desee establecer como obligatorios. Luego muestra además la opción Establecer.

Descripción de variable

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Elemento de LOM	campo de selección	Sí	En este campo solamente se seleccionan aquellos campos que se desea sean de llenado obligatorio.

Matriz de datos

SC 1 Establecer metadatos obligatorios para las unidades de aprendizaje

Escenario	Elemento de LOM	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
EC 1.1: Establecer metadatos obligatorios para las UOL.	V	El sistema establece los metadatos obliga- torios para la unidad de aprendizaje de forma correctamente.		Para establecer metadatos obligatorios para las UOL se: -El usuario accede al menú lateral izquierdo a la opción Metadatos. -Luego se da clic en la opción establecer metadatos obligatorios para las UOL -Después se da clic en el botón aceptar y se establecen de forma correcta metadatos obligatorios para las UOL.
	I			

Diseño de casos de prueba basado en CU Establecer metadatos obligatorios para los OA creados con SCORM 2004

Descripción general

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.

Condiciones de Ejecución.

El bibliotecario de contenidos debe estar autenticado en el sistema.

Secciones a probar en el Caso de Uso.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Establecer metadatos obligatorios para OA creados con SCORM 2004	EC 1.1: Establecer metadatos obligatorios para OA creados con SCORM 2004	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. Luego se muestra la opción Establecer.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Elemento de LOM	campo de selección	Sí	En este campo solamente se seleccionan aquellos campos que se desea sean de llenado obligatorio.

Descripción de variable

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Elemento de LOM	campo de selección	Sí	En este campo solamente se seleccionan aquellos campos que se desea sean de llenado obligatorio.

Matriz de datos

SC 1 Establecer metadatos obligatorios para OA creados con SCORM 2004.

Escenario	Elemento de LOM	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
EC 1.1: Establecer metadatos obligatorios para OA creados con	V	El sistema establece los metadatos obligatorios para OA creados con SCORM 2004		Para establecer metadatos obligatorios para OA creados con SCORM 2004 se:

Escenario	Elemento de LOM	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
SCORM 2004	I			-Accede al menú lateral izquierdo a la opción Metadato. -Luego se da clic en la opción Metadatos Establecer metadatos obligatorios para OA creados con SCORM 2004. -Después sale la ventana con nombre Establecer metadatos obligatorios para OA creados con SCORM 2004 mostrando todo los datos referentes para su correcto

Diseño de casos de prueba basado en CU Establecer metadatos obligatorios para los OA creados con SCORM 1.2

Descripción General

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.

Condiciones de Ejecución

El bibliotecario de contenidos debe estar autenticado en el sistema.

Secciones a probar en el Caso de Uso

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: "Establecer metadatos obligatorios (SCORM 1.2).¨.	*EC 1,1: "Establecer metadatos obligatorios (SCORM 1.2).".*	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.

Descripción de variable

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Agregación de Contenido	*ChecBox*	No	Este campo permite seleccionar los contenidos que desee agregar.
2	Organización de Contenido	*ChecBox*	No	Este campo permite seleccionar los contenidos a organizar.
3	Assets	*ChecBox*	No	En este campo se seleccionan los Assets
4	SCOs	*ChecBox*	No	En este campo se seleccionan los SCOs.

Matriz de datos

SC Establecer metadatos obligatorios (SCORM 1.2)

ID del escenario	Escenario	Agregación de Contenido	Organización de Contenido	Assets	SCOs	Respuesta del sistema	Resultado de la prueba
[EC 1,1]	"Establecer Metadatos"	V	V	V	V	Se espera que se seleccionen correctamente los datos	
		I	I	I	I	No se espera que se seleccionen correctamente los datos	
		I	V	V	V	No se espera que se establezca el metadato, ya que no se agregaron los contenidos.	
		V	I	V	V	No se espera que autocomplete el metadato, ya que no se selecciona los contenidos a organizar.	

		V	V	I	V	No se espera que autocomplete el metadato, ya que no se seleccionan los Assets.	
		V	V	V	I	No se espera que autocomplete el metadato, ya que no se seleccionan los SCOs.	

Diseño de casos de prueba basado en CU Autocompletar metadatos

Descripción general

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.

Condiciones de ejecución

El bibliotecario de contenidos debe estar autenticado en el sistema.

Secciones a probar en el caso de uso

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad

SC 1: "Autocompletar Metadatos¨.	*EC 1,1: "Autocompletar Metadatos".*	El sistema muestra los siguientes metadatos, ofreciéndole al actor la posibilidad de establecer valores para ellos: -**Idioma** (Categoría General): idioma del objeto de aprendizaje. -**Ámbito** (Categoría General) para escoger el país o países donde podrá aplicarse el OA. -**Idioma** (Categoría Meta-metadatos) para establecer el idioma con el que serán descritos los OA. -**Derechos de autor u otras restricciones** (Categoría Derechos) con valores: SI, NO, para establecer si el OA tiene o no derechos de autor u otras restricciones. (Ver sección 1) - **Costo** (Categoría Derechos) con los valores: SI, NO, para establecer si el OA tendrá costo o no.
		-**Localización** (Categoría Técnica): tendrá por defecto la dirección del repositorio: http://roa.uci.cu. Esta podrá ser cambiada. El sistema permite además, establecer los metadatos seleccionados.
SC 2: "Establecer Metadatos"	*EC 2,1: "Establecer Metadatos".*	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.

Descripción de variable

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Idioma	*ChecBox*	No	En este campo se permite escoger el idioma del objeto de aprendizaje.
2	Ámbito	*ChecBox*	No	En este campo se permite introducir el ámbito para escoger el país o países donde podrá aplicarse el OA.
3	Localización	*Campo de texto*	No	Este campo tendrá por defecto la dirección del repositorio: http://roa.uci.cu. Esta podrá ser cambiada.
4	Derechos de autor u otras restricciones	*ChecBox*	No	Este campo es para establecer si el OA tiene o no derechos de autor u otras restricciones.
5	Costo	*ChecBox*	No	Este campo es para establecer si el OA tendrá costo o no.

Matriz de datos

SC Autocompletar metadatos

ID del escenario	Escenario	Idioma	Ámbito	Localización	Derechos de autor u otras restricciones	Costo	Respuesta del sistema	Resultado de la prueba
[EC 1,1]	“Autocompletar”	V	V	V	V	V	Se espera que se entre correctamente los datos	
		I	I	I	I	I	No se espera que se entre correctamente los datos	
		I	V	V	V	V	No se espera que autocomplete el metadato, ya que no se selecciona el idioma.	
		V	I	V	V	V	No se espera que autocomplete el metadato, ya que no se selecciona el ámbito.	

		V	V	I	V	V	No se espera que autocomplete el metadato, ya que no se entra la localización.	
		V	V	V	I	V	No se espera que autocomplete el metadato, ya que nos e selecciona si tiene derecho de autor o no.	
		V	V	V	V	I	No se espera que autocomplete el metadato, ya que nos e selecciona si tiene costo o no.	

Diseño de casos de prueba basado en CU Buscar metadatos

Descripción general

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.

Condiciones de ejecución

El actor debe encontrarse autenticado en el sistema y en la edición de los metadatos.

Secciones a probar en el caso de uso

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Buscar metadatos.	EC 1.1: Buscar metadatos.	El sistema brinda la posibilidad de escoger el tipo de metadatos que desea buscar mediante las opciones: -Metadatos de Unidad de Aprendizaje. -Metadatos de Objetos de Aprendizaje. 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.

Descripción de variables

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Tipos de metadatos a buscar.	campo de selección	No	En este campo solamente se selecciona el tipo de metadato a buscar (OA, UOL).

2	Elemento de LOM.	campo de selección	Sí	En este campo se selecciona aquel elemento perteneciente a LOM por el cual se quiere hacer la búsqueda.
3	Valor	campo de texto	Sí	En este campo se introducen el valor del elemento asociado por el cual se desea realizar la búsqueda.

Matriz de datos

SC1 Buscar metadatos

Escenario	Tipos de metadatos a buscar.	Elementos de LOM.	Valor	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
EC 1.1: Buscar metadatos.	V	V	I	El sistema busca los		Para buscar metadatos se: -Accede al menú lateral izquierdo a la opción Metadato. -Luego se da clic en la opción Buscar Metadatos.
	V	I	V	El sistema busca los metadatos		
	V	V	I	El sistema busca los		
	V	V	V	El sistema busca los		

Escenario	Tipos de metadatos a buscar.	Elementos de LOM.	Valor	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
	I	I	I	El sistema no hace la búsqueda de los metadatos porque falta el tipo de metadatos		-Después sale la ventana con nombre Edición de Metadatos mostrando todos los datos referentes para su correcto establecimiento.

3.3 Clasificación de las no conformidades

La descripción de las clasificaciones de las **no conformidades** ayudará a que el proceso de pruebas se realice en un tiempo reducido y planificado, sin atentar contra el cronograma de desarrollo y los compromisos con el cliente. Se podrá realizar una evaluación más profunda del software a revisar, se tendrán mejores estadísticas de cuáles son las no conformidades más comunes en los diferentes tipos de software y se le podrá dar un mejor tratamiento a la no conformidad en el proceso de liberación del software (42).

Según lo planteado en la Norma UNE-EN ISO 9001:2000 una no conformidad. “Es el incumplimiento de algunos de los requisitos” (42).

Se trata de una desviación entre lo que hay escrito (lo que hemos dicho que vamos a hacer) y lo que ha ocurrido (lo que hemos hecho). Este fallo queda registrado en un informe y se establecen las acciones preventivas y correctivas necesarias para arreglar lo que no funcione y evitar que vuelva a ocurrir. (43).

Las mismas se clasifican de acuerdo al nivel de importancia en:

1. **Las No Conformidades Significativas**: Son aquellas que afectan la calidad del producto o servicio de manera visible, impidiendo o no el cumplimiento de algún requisito.
2. **Las No Conformidades No Significativas**: Son aquellas que resultan menos visibles, que no atentan el cumplimiento de algún requisito.
3. **Las Recomendaciones**: Son aquellas que quedan en función de la apreciación del probador para oportunidades de mejoras del producto o servicio.

Las no conformidades se clasifican además en función de sus características de acuerdo al tipo de artefacto como:

1. **No Conformidades de Documentación**:
   - Formato.
   - Error técnico.
   - Otros errores.
   - Correspondencia con otra documentación.
2. **No Conformidades de Aplicación**:
   - Validación.
   - Opciones que no funcionan.
   - Errores de interfaz.
   - Funcionalidad.
   - Excepciones.
   - Correspondencia de lo implementado con lo documentado.
3. **No Conformidades comunes (para ambos artefactos)**:
   - Ortografía.

- Redacción.
- Errores de idioma.

Resultados de las pruebas de caja negra.

Los resultados obtenidos durante las iteraciones fueron los descritos en la Tabla 4. En la misma se pueden apreciar la cantidad de no conformidades, las que fueron solucionadas (cerradas) y las que por algún motivo justificado no pudieron ser resueltas.

Iteración	No conformidades	Cerradas	No proceden
1ra	12	12	0
2da	7	7	0
3ra	2	2	0

Tabla 4: No conformidades.

Como se puede apreciar a partir de las pruebas realizadas a todos los casos de uso se obtuvieron 12 no conformidades en la primera iteración de las cuales 3 resultaron significativas, 4 no significativas y el resto recomendaciones. Para la segunda iteración se disminuyó a 4 no significativas y 3 recomendaciones. Y en la tercera y última iteración 2 recomendaciones.

La mayoría de las no conformidades detectadas se relacionaban con los mensajes al usuario después de realizar las acciones, la validación de las instancias de metadatos con el esquema asociado, la posterior edición de los metadatos y el autocompletamiento de los elementos por el sistema. Con el objetivo de erradicar las no conformidades se utilizaron expresiones regulares y se redefinieron determinadas funcionalidades.

Conclusiones

Al concluir el presente capítulo se logró probar de forma satisfactoria el módulo Metadatos. Durante la ejecución de las pruebas de Caja Negra mediante los casos de prueba se identificaron y corrigieron una serie de errores que fueron pasados por alto durante la etapa de implementación de este sistema. Este

procedimiento mejoró la calidad del producto final y garantizó que las funcionalidades definidas inicialmente estuviesen implementadas correctamente.

Conclusiones generales

Al término de la presente investigación para la implementación del módulo Metadatos para la gestión de los metadatos en los recursos que se crean en CRODA, se concluye que:

El esquema de metadatos LOM, se encuentra entre las principales soluciones para la descripción de recursos didácticos.

Se obtuvo un editor de metadatos integrado a la herramienta de autor CRODA, que facilitará la descripción de los recursos, incidiendo positivamente en el aumento de su reutilización.

Los métodos de validación definidos permitieron identificar y erradicar posibles errores del sistema, evitando insatisfacciones por parte de los usuarios

Recomendaciones

Para futuras versiones del módulo Metadatos se recomienda:

Mejorar el mecanismo para la gestión de los metadatos a autocompletar, permitiendo añadir y eliminar los elementos que se considera más usados en la institución que la utiliza.

Adaptar el módulo de manera tal que acepte otros estándares de catalogación además de LOM, permitiendo importar instancias de metadatos elaboradas en HA que utilizan otro estándar para la clasificación.

Bibliografía

1. **Rodríguez González, Leonardo.** *Análisis y Diseño de la versión 3.0 de RHODA.* Habana : s.n., 2011.

2. **Brand, Amy, Daly, Frank y Meyers, Barbara.** Metadata Demystified: A Guide for Publishers. [En línea] julio de 2003. [Citado el: 30 de mayo de 2012.] ISBN.

3. **López Guzmán, Clara.** *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.

4. **Hernández, Eduardo.** Unidades de aprendizaje, una propuesta de complemento a los objetos de aprendizaje. [En línea] enero de 2006. [Citado el: 15 de mayo de 2012.] Unidades de aprendizaje, una propuesta de complemento a los objetos de aprendizaje.

5. **López Guzmán, Clara y García Peñalvo, Francisco José.** *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 2005".

6. **TTnet.** *La Formación Sin Distancia.Estudio realizado por el Grupo de Trabajo de "e-learning" de la red TTnet.* Madrid : s.n., 2006.

7. *Estado actual de los sistemas e-learning.* **Peñalvo, Francisco José García.** 2, Salamanca : Ediciones Universales de Salamanca, 2005, Teoría de la educación: Educación y Cultura en la sociedad de la información., Vol. 6. ISSN 1138-9737.

8. **Ayuntamiento de Cartagena.** welearn.cartagena. [En línea] 2011. [Citado el: 9 de enero de 2012.] http://welearn.cartagena.es/moodle/mod/book/view.php?id=509&chapterid=11.

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

10. **Reload.** RELOAD. *Reusable eLearning Object Authoring & Delivery.* [En línea] 23 de Julio de 2008. [Citado el: 10 de Enero de 2012.] http://www.reload.ac.uk/.

11. **eXeLearning.** eXe. [En línea] 2008. [Citado el: 13 de enero de 2012.] eXe eXeLearning.

12. **Hot Potatoes.** Hot Potatoes. [En línea] 2011. [Citado el: 1 de junio de 2012.] http://web.uvic.ca/hrd/halfbaked/.

13. **Cuadernia.** Cuadernia. [En línea] 30 de abril de 2010. [Citado el: 1 de junio de 2012.] http://cuadernia.educa.jccm.es/.

14. **LTSC.** *Draft Standard for Learning Object Metadata.* New York : IEEE – SA Standard, 2002.

15. *Association for Educational Communications & Technology.Toward an instructional design model based on learning objects.* **Chiappe Laverde, Andrés, Segobia Cifuentes, Yasbley y Rincón Rodríguez, Helda Yadira.** 6, Condinarmuca : Learning and Performance Support Laboratory, University of Georgia, Athens, 2007, Educational Technology Research and Development, Vol. 55. 10.1007/s11423-007-9059-0.

16. **Gil Mateos, Jorge E.** *Estrategia de Gestión de Recursos Educativos Abiertos en forma de Objetos de Aprendizaje en la Universidad de La Habana.* Universidad de La Habana.La Habana : s.n., 2010.

17. **Berlanga, Adriana J., López, Clara y Morales, Erla.** *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.

18. **IMS.** IMS Global Learning Consortium From Innovation To Impact! [En línea] 2011. [Citado el: 11 de 1 de 2012.] http://www.imsglobal.org/content/packaging.

19. **Fernández Manjón, Baltasar, y otros.** *USO DE ESTÁNDARES APLICADOS A TIC EN EDUCACIÓN.* 2007.

20. **ADL.** *Modelo de Referencia de Objetos de Contenido Compartido.Modelo de Agregación de Contenidos.Versión 1.3.2 (Borrador).* Alexandria : s.n., 2006.

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

22. **Grupo de Catalogadores de Información Geográfica.** Geoportal sobre Metadatos de Información Geográfica. [En línea] 2010. [Citado el: 11 de Enero de 2012.] http://metadatos.latingeo.net.

23. **Universidad Carlos III de Madrid.** Metadatos Recuperación y Acceso a la Información. [En línea] 2010. [Citado el: 11 de Enero de 2011.] http://www.metadatos-xmlrdf.com/metadatos.

24. **Jorum y Baird, Kenny.** Automated Metadata. *Jorum.* [En línea] 26 de Julio de 2006. [Citado el: 12 de Enero de 2012.] http://www.jorum.ac.uk/docs/pdf/.

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

26. **López Guzmán, Clara y García, Francisco.** *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., 2008. pág. 10.

27. **Sommerville, Ian.** *Ingeniería del Software.* [trad.] María Isabel Alfonso Galipienso, y otros. Madrid : PEARSON EDUCACIÓN, SA., 2005. pág. 5 y 124. ISBN:84-7829-074-5.

28. **Martínez, Alejandro y Martínez, Raúl.** *Guía a Rational Unified Process.* 2007.

29. **Object Management Group.** Unified Modeling Language. [En línea] 2011. [Citado el: 25 de enero de 2012.] www.uml.org.

30. **Visual Paradigm.** Visual Paradigm. [En línea] 2009. [Citado el: 16 de 12 de 2011.] http://www.visual-paradigm.com/product/vpuml/provides/.

31. **Potencier, Fabien y Zaninotto, François.** *Symfony La guía definitiva.* [trad.] Miguel Sanchez, Luciano A. Andrade , Martín Palacio Pentucci Javier Eguíluz Pérez. s.l. : Apress, 2008. ISBN-13: 978-1590597866.

32. **Frederick, Shea, y otros.** *Learning Ext JS 3.2.* 2010.

33. **W3C.** W3C. *Extensible Markup Language (XML) 1.0 (Fifth Edition).* [En línea] 26 de noviembre de 2008. [Citado el: 10 de enero de 2012.] http://www.w3.org/TR/2008/REC-xml-20081126/.

34. **desarrolloweb.** [En línea] 2011. [Citado el: 11 de enero de 2012.] http://www.desarrolloweb.com/html.

35. **Eguíluz Pérez, Javier.** *Introducción a JavaScript.* 2009.

36. **PostgresSQL.** PostgresSQL. [En línea] 2007. [Citado el: 16 de diciembre de 2011.] http://www.postgresql.org/.

37. **Exist.** eXist. [En línea] 2006. [Citado el: 14 de diciembre de 2011.] http://exist.sourceforge.net/.

38. **Apache Software Fundation.** Apache. [En línea] 2011. [Citado el: 5 de enero de 2012.] http://httpd.apache.org/.

39. **Acuña, Karenny Brito.** Por qué utilizar RUP para desarrollar aplicaciones web. 2009. [En línea] 2009. http://www.eumed.net/libros/2009c/584/Por%20que%20utilizar%20RUP%20para%20desarrollar%20aplicaciones%20web.htm.

40. **Fernández, Luis, Lara, Pedro J. y Gutiérrez, Celia.** Generación de casos de prueba a partir de especificaciones UML. [En línea] 2003. [Citado el: 24 de 04 de 2012.] http://www.econonline.com.ar/metsi/wp-content/uploads/2010/10/Casos-de-Prueba-vs-Casos-de-Uso.pdf.

41. **Pressman.** Ingeniería del software. Un enfoque práctico. [En línea] 2002. [Citado el: 29 de mayo de 2012.] http://www.buscalibros.cl/ingenieria-software-un-enfoque-pressman-cp_185947.htm. ISBN 970-60-5473-3.

42. **Mares Amézquita, Saúl.** Gestión de No Conformidades. [En línea] 2007. [Citado el: 10 de 12 de 2011.] http://www.mex.ops-oms.org/contenido/tuberculosis/cdtaller/presentaciones/M%C3%B3dulo%208%20.

43. **Fernández Pereda, Héctor.** Norma ISO 9001:2000 Sistema de Gestión de la Calidad y Requisitos. [En línea] [Citado el: 10 de mayo de 2010.] http://www.buscarportal.com/articulos/iso_9001_2000_gestion_calidad.html..

44. **Calderón, Amaro, Valverde Rebaza, Sarah Dámaris y Carlos, Jorge.** *Metodologías Ágiles.* 2007.

45. **IBM.** Rational Rose Enterprise. [En línea] 2009. [Citado el: 12 de enero de 2012.] http://www-142.ibm.com/software/products/es/es/enterprise.

46. **Zend.** [En línea] 2011. [Citado el: 5 de enero de 2012.] http://www.zend.com.

47. **Oracle Corporation.** Oracle. [En línea] 2011. [Citado el: 10 de diciembre de 2011.] http://www.oracle.com.

48. **Cibernetia.** cibernetia. [En línea] 2011. [Citado el: 16 de diciembre de 2011.] http://www.cibernetia.com/manuales/instalacion_servidor_web/3_otros_servidores_web.php.

Glosario de términos

Reutilización: capacidad de los OA para integrarse con otros formando una sola entidad. En términos de reutilización de metadatos se refiere a la capacidad de reutilizar los metadatos de un recurso para describir otro.

Accesibilidad: capacidad que hace identificable y ubicable cuando se necesite al contenido de un recurso, para los requerimientos formativos necesarios, que se conozca su adecuación a los objetivos sin necesidad de obtener el propio contenido o pagar derechos por él, mediante la provisión de información suficiente.

TIC: Tecnologías de la Información y las Comunicaciones.

Open Source: código abierto. Manera de nombrar también a las aplicaciones desarrolladas bajo el amparo de un producto con licencia GPL.

Interoperabilidad: capacidad de las plataformas para comunicarse e integrarse de una forma efectiva aun siendo diferentes.

Ajax: acrónimo de Asynchronous JavaScript And XML, es una técnica de desarrollo web para crear aplicaciones interactivas.

DHTML: El HTML Dinámico o DHTML (del inglés Dynamic HTML) designa el conjunto de técnicas que permiten crear sitios web interactivos utilizando una combinación de lenguaje HTML estático, un lenguaje interpretado en el lado del cliente(como JavaScript), el lenguaje de hojas de estilo en cascada (CSS) y la jerarquía de objetos de un DOM.

DOM: Document Object Model, una interfaz de programación de aplicaciones (API) que proporciona un conjunto estándar de objetos que permiten acceder y modificar el contenido, estructura y estilo de los documentos HTML y XML.