



UNIVERSIDAD DE LAS CIENCIAS INFORMATICAS

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

**EDITOR DE METADATOS BASADO EN EL ESQUEMA LOM
PARA XALIX.**

Facultad 4

Autor:

Dayneli Saavedra Téllez.

Tutor:

Ing. Osvaldo Ernesto Estable.

La Habana, junio 2014.

Declaración de autoría

Declaro que soy la única autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firma la presente a los ____ días del mes de _____ del año 2014.

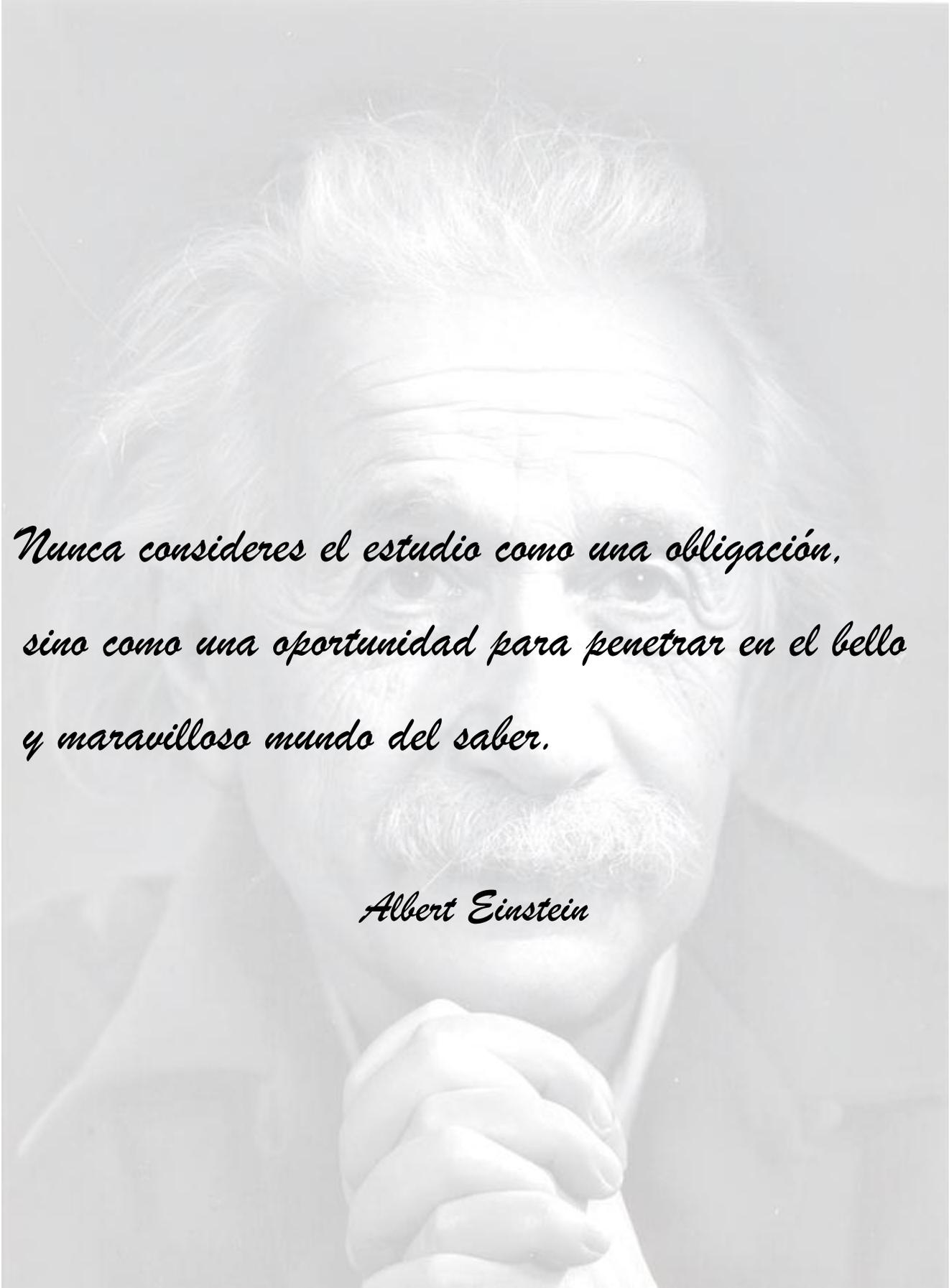
Autor:

Dayneli Saavedra Téllez

Tutor

Ing. Osvaldo Ernesto Estable





*Nunca consideres el estudio como una obligación,
sino como una oportunidad para penetrar en el bello
y maravilloso mundo del saber.*

Albert Einstein

Agradecimientos

A mi tutor Osvaldo Ernesto Estable por haberme dado la oportunidad de desarrollar este tema de tesis, por corregirme, guiarme y apoyarme no solo como tutor sino también como un buen amigo.

Al profesor Roberto López Desagües por corregirme en esta investigación.

A los profesores Orestes y Rosalba por brindarme su ayuda en un determinado momento.

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

A mis tres amigas Rosalia, Aimara y Gretel por apoyarme y soportarme siempre.

A Willy por su apoyo incondicional.

A Rosy por escucharme siempre.

A todos mis amigos, Yosbel, Gonzalo, Abel, Celso, Pérez, Darián y familia, por ayudarme a la instancia en la escuela y apoyarme cuando más lo necesitaba.

A mi gran amiga del alma Rossangel por sus consejos, apoyo incondicional, dedicación y por cargar conmigo en cualquier circunstancia.

A mi familia, especialmente a mis tías queridas Neyci y Elizabeth.

A mi hermanita querida por ser mi motor impulsor.

En especial a mi madre que es la causante de todo lo que he logrado en la vida, mi ídolo y la estrella que me alumbra.



Dedicatoria

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

A mi hermana Daynet por quererme tal y como soy y por darme fuerzas para seguir adelante.

A mi familia por estar siempre apoyándome.

A Willy por estar siempre presente en todo momento.

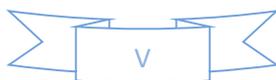
Resumen

En la actualidad se ha desarrollado de manera notable la tecnología vinculada con la ciencia, este desarrollo ha influido en la esfera educativa donde existe la marcada presencia de las Tecnologías de la Información y las Comunicaciones, que con su avance dieron paso al aprendizaje electrónico (e-learning) para recortar la brecha distancia. Con el desarrollo del e-learning surge la necesidad de crear recursos educativos que apoyen dicho aprendizaje, permitiendo a los estudiantes y a la sociedad en general, adquirir conocimientos de manera más cómoda. Estos recursos son adaptables a diferentes contextos y plataformas, por lo que surgen los Objetos de Aprendizaje (OA).

La Universidad de las Ciencias Informáticas cuenta con el Departamento de Producción de Herramientas Educativas del centro de Tecnologías para la Formación (FORTES), el cual se especializa en la creación de OA y en su área destinada al desarrollo de aplicaciones con fines formativos ha decidido integrar tres de sus productos: la Plataforma Educativa (ZERA), el Repositorio de OA (RHODA) y la Herramienta de Autor (CRODA).

Con el objetivo de apoyar dicha integración se decidió crear un marco de trabajo común (Xalix), en el cual se acoplaron las principales funcionalidades de cada una de estas aplicaciones en tecnologías comunes. Debido a la importancia que tienen los metadatos para la descripción de los recursos educativos, se decidió incorporar a Xalix un editor que facilitara a los usuarios la caracterización de los mismos.

El editor de metadatos presente hoy en CRODA no está desarrollado con tecnología compatible en su totalidad con la utilizada en el nuevo marco de trabajo, por tanto no se podía integrar este módulo de forma directa a Xalix, definiéndose como propuesta de solución, desarrollar un componente integrable en Xalix que permita la edición de metadatos basado en el esquema LOM.



Índice de contenido

Introducción.....	1
Capítulo 1: Análisis de conceptos, herramientas y tecnologías.....	6
Introducción.....	6
1.1 El aprendizaje electrónico (e-learning).....	6
1.1.1 Plataformas que interactúan en el entorno e-learning	6
1.1.2 Estándares en el entorno e-learning	11
1.1.3 Los metadatos en los OA.....	15
1.2 Herramientas y tecnologías a utilizar para el desarrollo del componente	17
Capítulo 2: Características y modelación de la propuesta de solución.....	26
Introducción.....	26
2.1 Modelo de dominio o modelo conceptual.....	26
2.2 Requerimientos del sistema	29
2.3 Identificación de actores del sistema.....	31
2.4 Identificación de casos de uso del sistema.....	31
2.5 Diagramas de colaboración	36
2.6 Modelo de Diseño	40
2.6.1 Patrones del Diseño	40
2.6.2 Patrones arquitectónicos.....	43
Capítulo 3: Implementación y validación de la solución propuesta.....	45
Introducción.....	45
3.1 Diagrama de componentes	45
3.2 Métodos de validación.....	47
3.2.1 Pruebas de integración.....	47
3.2.2 Pruebas de funcionalidad	47
3.2.3 Técnica de ladov	62
Conclusiones generales	66
Bibliografía.....	68
Glosario de términos.....	71
Anexos.....	74
Anexo 1. Descripción de los casos de uso identificados en el sistema	74
Anexo 2. Diagramas de colaboración de los casos de uso definidos.....	84

Índice de tablas

<i>Tabla 1: Identificación de los actores del sistema.</i>	31
<i>Tabla 2: Descripción del caso de uso Gestionar metadatos.</i>	32
<i>Tabla 3: DCP de la SC Editar metadatos.</i>	48
<i>Tabla 4: DCP de la SC Insertar metadatos.</i>	49
<i>Tabla 5: Descripción de la variable Categoría General.</i>	49
<i>Tabla 6: Descripción de la variable Categoría Ciclo de Vida.</i>	50
<i>Tabla 7: Descripción de la variable Categoría Meta-Metadatos.</i>	51
<i>Tabla 8: Descripción de la variable Categoría Técnica.</i>	52
<i>Tabla 9: Descripción de la variable Categoría Uso educativo.</i>	53
<i>Tabla 10: DCP de la SC Exportar metadatos.</i>	54
<i>Tabla 11: DCP de la SC Importar metadatos.</i>	54
<i>Tabla 12: Descripción de variable.</i>	55
<i>Tabla 13: Matriz de datos de la SC1 Importar metadatos.</i>	55
<i>Tabla 14: DCP de la SC Publicar metadatos.</i>	55
<i>Tabla 15: DCP de la SC Eliminar metadatos publicados.</i>	56
<i>Tabla 16: DCP de la SC Reutilizar metadatos.</i>	57
<i>Tabla 17: DCP de la SC Buscar metadatos.</i>	58
<i>Tabla 18: Descripción de la variable.</i>	58
<i>Tabla 19: Matriz de datos de la SC Buscar metadatos.</i>	59
<i>Tabla 20: Resultado de las pruebas de caja negra.</i>	61
<i>Tabla 21: Cuadro Lógico de ladov.</i>	62
<i>Tabla 22: Niveles de satisfacción.</i>	63
<i>Tabla 23: Resultado de las escala de satisfacción.</i>	64
<i>Tabla 24: Variables de la fórmula del ISG.</i>	64

Índice de ilustraciones

<i>Ilustración 1: Aspectos que involucra un OA.</i>	10
<i>Ilustración 2: Componentes de un OA.</i>	16
<i>Ilustración 3: Modelo de dominio.</i>	27
<i>Ilustración 4: Diagrama de casos de uso del sistema.</i>	32
<i>Ilustración 5: Estereotipo que representa la CI.</i>	36
<i>Ilustración 6: Estereotipo que representa la CE.</i>	37
<i>Ilustración 7: Estereotipo que representa la CC.</i>	37
<i>Ilustración 8: Diagrama de colaboración del caso de uso Gestionar metadatos.</i>	38
<i>Ilustración 9: Diagrama de colaboración del escenario Crear metadatos.</i>	38
<i>Ilustración 10: Diagrama de colaboración del escenario Modificar metadatos.</i>	39
<i>Ilustración 11: Diagrama de colaboración del escenario Eliminar metadatos.</i>	39
<i>Ilustración 12: Diagrama de clases de diseño del sistema.</i>	44
<i>Ilustración 13: Diagrama de Componentes.</i>	46
<i>Ilustración 14: Resultado de las pruebas por iteraciones.</i>	61

Ilustración 15: Fórmula de ISG. 64
Ilustración 16: Nivel de satisfacción..... 64

Introducción

El desarrollo de la ciencia y la tecnología en la actualidad ha provocado cambios en diversas esferas resaltando la relacionada con el proceso educativo. Los cambios en esta área están definidos en la manera en que los estudiantes reciben los contenidos que el profesor elabora a través de diferentes medios, tanto clases como cursos, permitiendo de esta forma demostrar los conocimientos adquiridos.

Las posibilidades de desarrollo de modalidades educativas que de alguna manera han permitido que en el ámbito educativo surjan propuestas o modelos como materiales didácticos que apoyen los procesos de enseñanza y aprendizaje, han incrementado. Esto, junto al avance de las Tecnologías de la Información y las Comunicaciones (TIC), demanda cambios en las metodologías y estrategias educativas, para responder con ello a las necesidades de conocimiento de la sociedad (1).

Con la vinculación de las tecnologías y el proceso de enseñanza-aprendizaje ha surgido el denominado e-learning (aprendizaje electrónico), el cual incluye un conjunto de estándares tecnológicos de Internet, métodos y recursos educativos que posibilitan la educación a distancia, es decir, el aprendizaje sin la necesidad de la presencia física de un profesor. El e-learning no supera el método tradicional de enseñanza, pero sí ayuda en gran medida, ya que gracias a las TIC, los estudiantes pueden comunicarse e interactuar con sus compañeros de clase y profesores, brindándoles comodidades en sus horarios de estudio y proporcionando mayor participación del estudiante en las actividades propuestas.

Los recursos educativos constituyen una de las formas de apoyar al quehacer diario de los docentes, dentro de los que se encuentran apuntes, esquemas didácticos y libros, pero dentro del marco de la educación a distancia se hace necesario el uso de recursos educativos que proporcionen una vía de facilitación de aprendizaje, para lo cual se han realizado diversas acciones como puede ser la creación de Objetos de Aprendizaje (OA) (2).

Los OA posibilitan la compartición de contenidos entre instituciones; dichos materiales educativos han sido diseñados y creados en unidades pequeñas, con el propósito de maximizar el uso de un mismo material en distintas situaciones educativas, pudiendo ser actualizados sin modificar el curso o programa completo. Los objetos de aprendizaje en el ámbito educativo se introducen sin considerar necesariamente a las TIC, sin embargo, es a partir de éstas cuando cobra fuerza la idea de tener unidades de aprendizaje interoperables, con la capacidad de integrarse en estructuras y plataformas diferentes, reutilizables, durables y actualizables (1).

Los OA pueden ser almacenados digitalmente, distribuidos, reutilizados y recuperados. Según el Comité de Estándares de Tecnologías de aprendizaje (LTSC – Learning Technology Standards Committee por sus

siglas en inglés), "Los objetos de aprendizaje se definen como cualquier entidad, digital o no digital, que puede ser utilizada, reutilizada o referenciada durante el aprendizaje apoyado en la tecnología" (1).

Cuando se habla de objetos de aprendizaje es natural también hablar de "repositorios" de objetos de aprendizaje. Estas colecciones de recursos digitales constan de dos partes: los contenidos (objetos digitales) y la meta, información asociada a los contenidos denominada "Metadato", que permiten la catalogación digital de la información que contiene un OA y su reutilización en diversos contextos (1).

Todos los OA son descritos a través de metadatos, los cuales se refieren a la información de la información (datos de los datos). Los metadatos son un conjunto de atributos o elementos necesarios para describir un recurso. A través de los metadatos se tiene un primer acercamiento con el objeto, conociendo rápidamente sus principales características. La selección correcta del esquema de metadatos y la asignación adecuada de sus valores, dan a los contenidos las propiedades necesarias para potenciarlos como recursos reutilizables, flexibles y durables (3).

Para gestionar los OA, varias han sido las herramientas desarrolladas, las cuales permiten su creación y almacenamiento, destacándose los Repositorios de Objetos de Aprendizaje (ROA) y las Herramientas de Autor (HA).

Los ROA son creados para el almacenamiento organizado de OA, permitiendo así su localización, recuperación, catalogación, reutilización, revisión y eliminación. Es imposible pensar en objetos de aprendizaje sin buscarles un lugar donde albergarlos. Existen dos tipos de ROA, primeramente los que almacenan recursos con sus metadatos, donde el objeto y su descripción se encuentran en un mismo sitio y los que sólo almacenan metadatos y el objeto se referencia en otro repositorio generalmente (4).

Las HA facilitan la creación de OA, dependiendo de las características propias de cada una, permiten a los profesores trabajar con diferentes funcionalidades que se hacen necesarias para la creación de actividades variadas como describir, importar y exportar objetos creados, además de crear y describir los diseños instruccionales para puntualizar actividades que posteriormente las realizarán tanto alumnos como profesores.

Durante la creación y almacenamiento de un OA es importante tener en cuenta el uso de estándares que son los que garantizan la interoperabilidad de los recursos con diversas organizaciones. Los mismos son como patrones, modelos o guías que las instituciones adoptan o crean para realizar diferentes procesos.

Para el desarrollo del editor de metadatos resulta interesante la utilización del estándar LOM debido a su aplicación en el ámbito educativo y a sus beneficios en la descripción de los recursos. LOM contiene un

grupo mínimo de elementos para la administración, ubicación y evaluación de los objetos de aprendizaje, está 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, por lo que es necesario prestarle la debida atención a la asignación adecuada de los valores y la descripción de los metadatos en la elección de su esquema, el cual es el punto de partida hacia un recurso reutilizable, duradero y flexible (3).

En la Universidad de las Ciencias Informáticas (UCI) se encuentra el Departamento de Producción de Herramientas Educativas del Centro de Tecnologías para la Formación (FORTES) de la Facultad 4, el cual en su área destinada al desarrollo de aplicaciones con fines formativos, ha decidido integrar tres de sus productos: la herramienta de autor CRODA, el repositorio de OA RHODA y la Plataforma Educativa ZERA. Todos deberán estar acoplados mediante componentes a un único marco de trabajo (Xalix).

Para dicho desarrollo se necesita incorporar cada una de las funcionalidades de estas tres aplicaciones en tecnologías comunes. En el caso específico de la herramienta de autor CRODA, hoy cuenta con importantes funcionalidades como la creación individual y colaborativa de OA basados en SCORM como estándar y diseños de aprendizaje basados en la especificación IMS-LD, haciéndose necesario la descripción de metadatos basados en el esquema LOM para lo cual la herramienta cuenta con un editor.

El editor de metadatos presente hoy en CRODA no está desarrollado con tecnología compatible en su generalidad con la utilizada en el marco de trabajo y el servidor utilizado como soporte para el modelo de datos genera bajo rendimiento, por lo que contiene insuficiencias que no le permiten integrar este módulo de forma directa al marco de trabajo Xalix.

Lo anterior expuesto denota la presencia de un problema a resolver, por lo que surge la siguiente **interrogante**: ¿Cómo integrar las funcionalidades del editor de metadatos de CRODA en el marco de trabajo Xalix?

Por lo que se define como **objetivo general**: Desarrollar un componente integrable al marco de trabajo Xalix, que permita incorporar las funcionalidades del editor de metadatos de CRODA.

Para guiar el desarrollo del software se plantean las siguientes **preguntas científicas**:

- ¿Cuáles son las características de la especificación LOM?
- ¿Cuáles son las tecnologías necesarias para el desarrollo de un componente que permita la edición de los metadatos en el marco de trabajo Xalix?

- ¿Cómo determinar las funcionalidades que debe brindar un componente para la edición de metadatos en el marco de trabajo Xalix?
- ¿Cómo crear los componentes y funcionalidades del editor de metadatos en el marco de trabajo Xalix?
- ¿Cómo integrar la propuesta de solución desarrollada en el marco de trabajo Xalix?

Para dar cumplimiento al objetivo general se plantean los siguientes **objetivos específicos**:

- Analizar tecnologías que contribuyan con la edición de metadatos basados en el esquema LOM.
- Analizar el funcionamiento y desarrollo del editor de metadatos en CRODA v2.0.
- Desarrollar el editor de metadatos para el marco de trabajo Xalix.
- Comprobar el funcionamiento del editor de metadatos.

El **objeto de estudio** de este trabajo se enmarca en el proceso de integración en marcos de trabajo dentro del desarrollo de software, teniendo como **campo de acción** un componente integrable en el marco de trabajo Xalix, que permita la edición de metadatos basados en el esquema LOM.

Para dar cumplimiento a los objetivos específicos planteados, se conforman las siguientes **tareas de investigación**:

- Análisis de los conceptos y tecnologías más utilizadas para conformar la base teórica metodológica de la investigación.
- Análisis del funcionamiento y desarrollo del editor de metadatos en CRODA v2.0.
- Elaboración del diseño de un componente que permita la edición de metadatos basados en el esquema LOM.
- Desarrollo del editor de metadatos.
- Comprobación de la integración y funcionamiento del editor de metadatos en Xalix.

Para dar cumplimiento a las tareas planteadas se hace necesario el empleo de los siguientes **métodos científicos**. El **Analítico-Sintético** se aplica en la presente investigación durante el análisis de las funcionalidades y componentes presentes hoy en CRODA que permiten el trabajo con los metadatos, el modelo de información generado por la especificación LOM para la creación de cuestionarios y

tecnologías a utilizar, que permitan el desarrollo de un componente que contribuya con la descripción de los recursos con fines formativos, presentes en el marco de trabajo. El **Histórico-Lógico** para estudiar el origen y evolución de la especificación LOM, como el desarrollo del trabajo con los metadatos en los sistemas que sean integrados con nuevas tecnologías en el marco de trabajo Xalix. El método **Modelación** para el esbozo de los diferentes diagramas y modelos generados durante el proceso de desarrollo de un componente, que contribuya con la descripción de los recursos con fines formativos presentes en el marco de trabajo Xalix.

El presente trabajo se estructura en tres capítulos:

- **Capítulo 1:** Se plantea la fundamentación teórica de la investigación, destacando conceptos relacionados a los recursos educativos y al aprendizaje electrónico. Se describen las tecnologías y metodología de desarrollo de software, que serán usadas durante la implementación del sistema.
- **Capítulo 2:** Se realiza el análisis y diseño de la propuesta de solución, se describen los requisitos funcionales y no funcionales que posteriormente serán utilizados como punto de partida en la implementación del editor de metadatos. Se reflejan las prácticas de programación, patrones arquitectónicos, patrones de diseño, garantizando que el sistema cumpla con los requerimientos deseados.
- **Capítulo 3:** En este capítulo se describen todos los procesos relacionados a la implementación y pruebas, además de la estrategia trazada para desarrollar las mismas y la técnica para saber la satisfacción del cliente.

Capítulo 1: Análisis de conceptos, herramientas y tecnologías

Introducción

En este capítulo se realiza la fundamentación teórica de esta investigación, abordando temas de relevancia para la misma como los conceptos y términos asociados al dominio del problema. Se realiza un estudio de las diferentes herramientas que son útiles para la implementación del editor de metadatos. Se realiza la selección de la metodología a utilizar en el ciclo de vida del software. De igual manera se hace una revisión y elección de lenguajes de programación web propuestos para el desarrollo del sistema.

1.1 El aprendizaje electrónico (e-learning)

El Aprendizaje electrónico (e-learning) puede ser definido como una modalidad de aprendizaje dentro de las herramientas o aplicaciones de hipertexto como soporte (Correo electrónico, web, Chat, etc.) y los contenidos y/o unidades de aprendizaje en línea (5).

El “aprendizaje electrónico” se produce a través de los estándares tecnológicos de Internet y los materiales descargables, que son un componente importante de apoyo para dicho aprendizaje (9).

Desde el punto de vista conceptual e-learning es un término al cual se le han asignado múltiples definiciones: formación en línea, cursos en línea (on-line), formación virtual y formación a distancia. Pero en general se traduce como aprendizaje producido a través de un medio tecnológico-digital. Gracias a las TIC, los estudiantes "en línea" pueden comunicarse y colaborar con sus compañeros "de clase" y docentes (profesores, tutores, etc.), de forma síncrona o asíncrona, sin limitaciones de tiempo y espacio. Dentro de la Educación a distancia, es una de las opciones más frecuentemente utilizada para atender la necesidad de educación continua o permanente. Dadas sus características y el soporte tecnológico que la respalda, la educación virtual da la oportunidad de que el estudiante elija sus horarios de estudio, convirtiéndose así en una muy buena opción para aquellas personas que trabajen y quieran estudiar en sus momentos libres, posibilitando no solo aprender conceptos nuevos, sino también afianzar conocimientos y habilidades, aumentado así la motivación de los estudiantes por diferentes temas (5).

1.1.1 Plataformas que interactúan en el entorno e-learning

En la actualidad, en el ámbito de la práctica para llevar a cabo un programa de formación basado en e-learning, además del uso de Internet como medio de comunicación, se requiere de herramientas o sistemas de software que permitan la comunicación e interacción entre profesores, alumnos y contenidos. Estos sistemas consisten principalmente en plataformas de aprendizaje, cuya función es dar seguimiento

a los cursos, los cuales han quedado limitados para aplicaciones más complejas en las que se busca, entre otras, la construcción, integración y comunicación de contenidos a fin de compartir y reutilizar lo que otros han realizado en el mismo campo. El objetivo es crear entornos interoperables en los que se integren otros componentes a un entorno e-learning como son, además de una plataforma de aprendizaje, administradores de contenidos, repositorios de objetos de aprendizaje y herramientas de autor (3).

Las plataformas de aprendizaje o LMS (Learning Management System por sus siglas en inglés), son aplicaciones Web que proveen módulos para las tareas administrativas y de seguimiento que se requieren en un proceso de enseñanza, simplificando su control. Los módulos administrativos permiten, la configuración de cursos, matricular alumnos, registrar profesores y asignarle dichos cursos a un alumno. Los módulos de seguimiento llevan informes de progreso y calificaciones. Estos sistemas son los más populares y la mayoría de las organizaciones que tienen un sistema e-learning, cuentan nada más con este tipo de aplicación (3).

Un LMS registra usuarios, organiza catálogos de cursos, almacena datos de los usuarios y provee informes para la gestión. Dichos sistemas incluyen herramientas de comunicación al servicio de los participantes en los cursos. Las mejoras en usabilidad y accesibilidad permiten salvar la brecha digital y extender las posibilidades de formación a mayor número de personas, superando la barrera tecnológica (5).

Los sistemas administradores de contenidos o LCMS por sus siglas en inglés, tienen su origen en los sistemas de gestión de contenido o CMS (Content Management System), cuyo objetivo es simplificar la creación y administración de los contenidos en línea utilizados principalmente en portales con publicaciones periódicas. En la mayoría de los casos los CMS separan los contenidos de su presentación o estilo en pantalla y facilitan un mecanismo de trabajo para la gestión de una publicación Web. Los LCMS siguen el concepto básico de los CMS, pero enfocados al ámbito educativo, administrando y concentrando únicamente recursos de aprendizaje y no todo tipo de información. En esencia, un LCMS es un sistema basado en Web que se utiliza para crear, aprobar, publicar, administrar y almacenar recursos educativos (3).

Los repositorios son unidades de almacenamiento que concentran los recursos educativos de forma ordenada para facilitar su localización y reutilización. Usualmente tanto los LMS como los LCMS tienen repositorios pequeños, que almacenan contenidos para el uso exclusivo de cada uno de ellos, pero la tendencia actual es contar con los llamados Repositorios de Objetos de Aprendizaje, que son capaces de exportar contenidos que fácilmente pueden incorporarse tanto a los LMS como a los LCMS (3).

Las herramientas de autor son aplicaciones de software que sirven para la construcción o elaboración de contenidos (3).

En el desarrollo del componente se interactuó específicamente con la herramienta de autor CRODA, estudiando sus funcionalidades para lograr la integración del editor de metadatos al marco de trabajo Xalix.

CRODA, herramienta de autor para la creación de OA

El presente trabajo propone un entorno para la edición de metadatos en el marco de trabajo Xalix a partir de la herramienta de autor CRODA, en el cual se integran herramientas tecnológicas -metodológicas con las potencialidades de la web 2.0.

El término Objeto de Aprendizaje fue introducido por Wayne Hodgins en 1992. A partir de esa fecha han sido muchos los autores que han definido el concepto; de hecho, la falta de consenso en su definición ha llevado a la utilización de múltiples términos y sinónimos: objetos de aprendizaje reutilizables, objeto de conocimiento reutilizable y cápsula de conocimiento. David Willey, en el año 2001 propone la siguiente definición de OA: “cualquier recurso digital que puede ser usado como soporte para el aprendizaje” (1).

Para la gestión de OA existen diversas herramientas entre las que se encuentran los repositorios de objetos de aprendizaje (ROA) que posibilitan su almacenamiento, los sistemas para la gestión del aprendizaje (LMS) que facilitan su distribución y las herramientas de autor, cuyo valor radica en permitir la creación de los OA. Estas últimas constituyen un software que permite a una persona en específico crear un programa instruccional sin elementos de programación explícitos, simplemente especificando el contenido instruccional y la lógica del aprendizaje.

Existen varias herramientas de autor que han sido desarrolladas entre las que se destacan RELOAD, Uduu y AUTORE. Estas permiten de manera general la creación de recursos educativos reutilizables, interoperables, accesibles y duraderos. A pesar de sus ventajas y facilidades para la creación de OA, las mismas no posibilitan el intercambio de conocimientos entre docentes, que favorezca el incremento de la calidad de dichos OA.

El término “ web 2.0” fue acuñado por O’ Reilly Media en el año 2004 y se pretende que las aplicaciones relacionadas con esta web intenten ser más dinámicas y se caractericen como “comunidades sociales”, donde el mayor énfasis se da a la contribución y participación de los usuarios. En aplicaciones relacionadas con la web 2.0, las personas que interactúan con la misma desarrollan una reputación en base a la cantidad y calidad de sus contribuciones, se comparten documentos en los que varias personas

pueden trabajar al mismo tiempo, se utilizan interfaces dinámicas y atractivas que se acercan a las aplicaciones de escritorio, se comparte información, en ocasiones en tiempo real, por medio de interfaces de programación y comunicación que permite el desarrollo rápido de nuevas aplicaciones y la participación de la comunidad en el etiquetamiento, clasificación y toma de decisiones (6).

La web 2.0 utiliza herramientas y servicios web que se complementan como blogs, wikis y otras herramientas sociales, para apoyar la creación de comunidades de aprendizaje. En el Centro de Tecnologías para la Formación (FORTES) de la Universidad de las Ciencias Informáticas (UCI) se encuentra la herramienta de autor llamada CRODA que posibilita la creación de OA a partir de un conjunto de plantillas predefinidas que guían al profesor en la estructuración de su objeto. Además permite la descripción de los OA a partir del esquema de metadatos LOM, lo cual facilita una amplia catalogación. CRODA además exporta los objetos utilizando el estándar SCORM lo que contribuye a su interoperabilidad. Con vistas a obtener los beneficios que proporciona la creación colaborativa de OA, conjunto a la creación del equipo, se genera un área de debate para el OA en cuestión (chat) (6).

Objetos de aprendizaje

Abordar una definición de Objeto de Aprendizaje es una tarea un tanto complicada, ya que existen variados conceptos con respecto al término, gracias a la evolución ha tenido que adaptarse a las nuevas necesidades educativas y tecnológicas. Como ejemplos de aprendizajes apoyados por la tecnología se incluyen: los sistemas de entrenamiento basados en computadoras, los ambientes de aprendizaje interactivos, los sistemas inteligentes de instrucción apoyada por computadoras, los sistemas de aprendizaje a distancia, los ambientes de aprendizaje colaborativo, los contenidos multimedia, el contenido instruccional, los objetivos de aprendizaje, el software instruccional y las herramientas de software, así como las personas, organizaciones o eventos referenciados durante el aprendizaje apoyado por la tecnología (1).

Finalmente es de gran ayuda que los profesores reciban una capacitación referente al uso de los OA para que de esta forma puedan ser incorporados a acciones educativas que tengan como objetivo que los estudiantes adquieran habilidades de aprendizaje y contribuyan a su propio desarrollo.

Los objetos de aprendizaje es un tema que puede situarse para su análisis en diferentes escalas para identificar sus posibilidades e implicaciones. A continuación se muestra un esquema con algunos aspectos que involucra los OA:

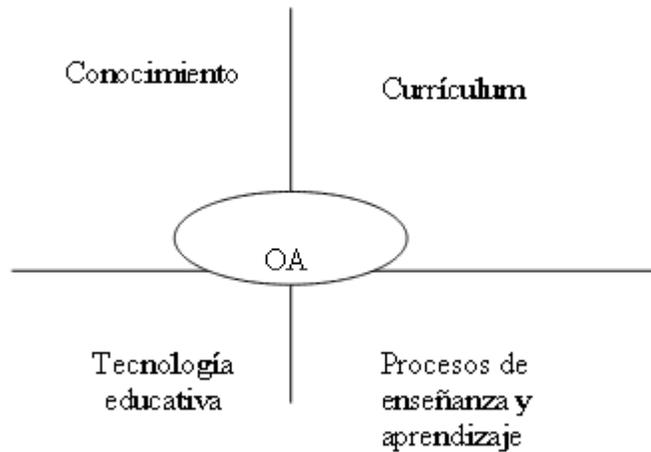


Ilustración 1: Aspectos que involucra un OA.

Conocimiento

Las Instituciones Educativas deben lograr sistemas eficientes para la producción y distribución del conocimiento y podría verse en los objetos de aprendizaje una estrategia de distribución que permita que los productos llegaran a más usuarios obteniéndose mayores beneficios por el conocimiento usado al interior de las organizaciones. También sería de gran beneficio no la búsqueda de eficiencia por la economía de esfuerzos en torno a la producción, distribución y uso de recursos para el aprendizaje, sino la puesta en circulación de insumos para la vinculación de comunidades interesadas en problemáticas afines; acercamiento de modos de ver problemas y formas de intervención; desarrollo de cadenas de producción en las que un objeto provoque derivaciones elaboradas por múltiples usuarios, es decir, que se puedan difundir y transmitir las propias opiniones de diferentes personas u organismos con respecto a temas específicos y de esta manera ampliar tanto en conocimiento como en desarrollo, ya que son dos términos que van juntos en esta temática (7).

Currículum

Pasando al segundo cuadrante del esquema inicial, el diseño educativo por objetos de aprendizaje puede constituir una oportunidad para romper los esquemas disciplinarios cerrados que se han manifestado en los planes por asignaturas. Es cierto que una forma de activar la producción de objetos para un acervo nacional es la digitalización de los contenidos de asignaturas. Una estrategia que promueva un acervo de

esta naturaleza poco contribuiría a la innovación curricular que se ha querido promover en las instituciones desde hace por lo menos dos décadas (7).

Procesos de enseñanza y de aprendizaje

La vinculación entre investigación y docencia podría verse favorecida si la producción de objetos de aprendizaje se orienta a la difusión de resultados de investigación, visiones sobre problemáticas, instrumentos para producir más información sobre problemas y campos de problemas (7).

Tecnología educativa

Los objetos de aprendizaje suponen enriquecimiento de las representaciones que se tengan sobre cualquier concepto u objeto real por integración de lenguajes múltiples. De igual forma permiten que los componentes con los que se ha construido un objeto puedan ser reorganizados para dar por resultado nuevos objetos. Al producir digitalmente recursos para el aprendizaje se utilizan diversos componentes: fotografías, videos, textos, gráficas, animaciones, dibujos y ejercicios (7).

Todos estos elementos contribuyen de una u otra manera a la difusión de la información y al aprendizaje de una forma más interactiva y dinámica.

1.1.2 Estándares en el entorno e-learning

El objetivo de los estándares es simplificar el trabajo de los docentes y no servir como limitador de la función o de la creatividad del educador. Una de las principales funciones de los mismos es facilitar la durabilidad y la reutilización, además del intercambio de contenidos entre diferentes plataformas y sistemas. En la tecnología y en la industria los estándares son elementos fundamentales ya que permiten el desarrollo de tecnologías aceptadas y compartidas por todos, que demuestran una madurez de dichos campos (8).

IMS-LD

Un Diseño de Aprendizaje (Learning Design por sus siglas en inglés) es definido en la especificación IMS como "una descripción de un método que permite a los alumnos alcanzar ciertos objetivos de aprendizaje por medio del desarrollo de ciertas actividades en un determinado orden, en el contexto de un cierto ambiente de aprendizaje" (9).

También se le define como la aplicación de un modelo pedagógico para un objetivo de aprendizaje, un grupo objetivo y un contexto determinado. En realidad, IMS Learning Design permite que a través de una

especificación aceptada públicamente, se exprese un proceso de enseñanza-aprendizaje, lo que equivale a describir las actividades y los recursos de ellas, que serán utilizadas y desarrolladas por personas (alumnos y tutores), para permitir que los alumnos alcancen un objetivo educacional esperado. Según esta especificación, un aprendizaje se diseña representando la siguiente idea: una persona adquiere un rol en el proceso de enseñanza-aprendizaje, usualmente este rol será de alumno o miembro del equipo docente, como un tutor, mentor. Dentro del rol, las personas desarrollarán actividades para lograr ciertas salidas o productos. Las actividades bajo esta especificación podrán ser de aprendizaje o de soporte y serán desarrolladas dentro de un ambiente que dependerá de los servicios y los objetos de aprendizajes necesarios o apropiados para que los roles desarrollen dichas actividades (9).

Soluciones ofrecidas por IMS LD

Sin mecanismos acordados y compatibles que describan las estrategias de enseñanza, los creadores de materiales educativos y la organización de éstos, continúan experimentando dificultades innecesarias en:

- Documentar las estrategias utilizadas al interior de estos materiales.
- Apego a procedimientos preestablecidos para asegurar la consistencia de la documentación.
- Asegurar que la calidad de la enseñanza se cumpla a través y entre organizaciones.
- Reutilización de elementos de materiales existentes de enseñanza.
- Reutilización e interoperabilidad no sólo de los recursos, sino de un diseño de aprendizaje.

IMS LD provee un nivel de abstracción en el proceso, ofreciendo piezas genéricas para diferentes enfoques pedagógicos. Utilizando el lenguaje, los diseñadores pueden conversar en términos más de pedagogía que de tecnología, haciendo por lo tanto, explícitas las alternativas pedagógicas sujetas a revisión, inspección, crítica y comparación (9).

Estándares para el empaquetamiento de recursos.

Los estándares de empaquetamiento son aquellos que permiten el intercambio de materiales entre las herramientas y el sistema, como 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.

En la actualidad son muchos los estándares existentes creados por distintas organizaciones para el empaquetamiento, dentro de los que se encuentra el estándar SCORM.

SCORM

Define un software que describe el modelo de agregación de contenidos, las relaciones que se establecen entre los componentes de los cursos, los modelos de datos y los protocolos de comunicación, de manera que puedan compartirse entre diferentes LMS los objetos definidos en uno de estos. Los elementos más característicos del modelo son:

- **SCO** (Sharable Courseware Object): Curso o componente de un curso que cumple con los requisitos de interoperabilidad, durabilidad y que dispone de la información suficiente para poder ser reutilizado y accesible. Un SCO es la mínima unidad intercambiable entre sistemas compatibles con SCORM y consiste en un objeto de aprendizaje que incluye un módulo software que le permite comunicarse con el entorno de ejecución proporcionado por el LMS.
- **Assets**: Recursos o elementos básicos, como ficheros de texto, audio, video, etc. Estos recursos básicos se agrupan en los SCOs.
- **Entorno de ejecución** (Runtime Environment, RTE): Propone un entorno estándar en el que se puede presentar un objeto de aprendizaje (en este caso un SCO) que es capaz de intercambiar datos con el LMS (8).

Estándares para la descripción de recursos

Debido a la falta de información acerca de diferentes recursos digitales, varias organizaciones se han involucrado en procesos de elaboración de esquemas de metadatos con el objetivo de localizar, clasificar y reutilizar dichos recursos para establecer un conjunto de acciones y para describir OA. Dentro de estos esquemas se encuentra LOM que será analizado a continuación.

IEEE Learning Object Metadata (LOM)

Los metadatos proporcionan características, propiedades e información sobre los objetos de aprendizaje que permiten describirlos, de forma que se simplifica su uso y gestión. Gracias a esta información complementaria, se puede saber cuál es el contenido y el propósito de un OA sin tener que acceder al mismo. Por tanto, los metadatos aportan información orientada a hacer más eficiente la búsqueda y utilización de los recursos. Los metadatos se pueden aplicar tanto a OA concretos como a cursos completos o a partes del curso.

Actualmente LOM es el estándar de e-learning formalmente aprobado que goza de mayor aceptación. El objetivo de LOM es la creación de descripciones estructuradas de recursos educativos. Su modelo de datos especifica qué aspectos de un objeto de aprendizaje deberían ser descritos y qué vocabularios se pueden utilizar en dicha descripción (8).

Los metadatos en la adopción IMS del estándar LOM están agrupados en categorías de metadatos. De forma más concreta, LOM distingue 9 categorías de metadatos diferentes:

1. Categoría general: Los metadatos en esta categoría representan información general sobre el material educativo que describe el mismo como un todo.
2. Categoría lifecycle (ciclo de vida): Esta categoría agrupa metadatos referidos a la historia y estado actual del proceso de producción y mantenimiento del material educativo por parte de los autores.
3. Categoría metametadata (meta-metadatos): Esta categoría agrupa información relativa a los metadatos en sí (de ahí su nombre).
4. Categoría technical (técnica): Categoría que agrupa metadatos relativos a las características y requisitos técnicos del material en sí.
5. Categoría educational (educativa): Categoría que agrupa metadatos relativos a los usos educativos del material.
6. Categoría rights (derechos): Categoría que agrupa metadatos relativos a los derechos de propiedad e intelectuales del material.
7. Categoría relation (relación): Categoría de metadatos utilizados para establecer relaciones entre el material y otros materiales.
8. Categoría annotation (anotación): Anotaciones y comentarios sobre el material educativo.
9. Categoría classification (clasificación): Metadatos para la clasificación del material en taxonomías (8).

Obligatoriedad de elementos en LOM

Muchas investigaciones afirman que los sesenta y cuatro elementos para los metadatos que proporciona LTSC (Comité de Estándares para Tecnología del Aprendizaje) son 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 (10).

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 (10).

SCORM 1.2 propone 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 edición 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 (11).

Las Herramientas de Autor como CRODA deben brindar a las instituciones que la emplean 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.

1.1.3 Los metadatos en los OA

Los metadatos permiten potenciar a los OA como recursos educativos recuperables, localizables, intercambiables y reutilizables. Su importancia radica en que a través de ellos se pueden conocer las principales características de un objeto y de esta forma conocer más acerca del mismo. Los metadatos son especialmente útiles en los recursos no textuales y en los que el contenido no puede ser indexado por sistemas automáticos como por ejemplo, los elementos multimedia o de audio. Diferentes sectores y organismos internacionales se encuentran desarrollando diversos esquemas y estándares para formalizar el uso de metadatos y establecer un conjunto de reglas semánticas, sintácticas y de contenido, que pretenden describir homogéneamente OA o recursos de información. Existen por lo tanto, diferentes propuestas al respecto. Con el uso del esquema adecuado, si los metadatos son semántica y estructuralmente correctos, el beneficio es doble. Por un lado se facilita la reutilización de los OA y por otro se posibilita la reutilización de los propios metadatos logrando que el mismo OA cumpla con diferentes esquemas de metadatos dirigidos a diferentes contextos o aplicaciones (12).

Los metadatos no son útiles solamente para describir y localizar determinado recurso, sino que también pueden contener información con fines administrativos y estructurales.

Caplan define los tipos de metadatos como:

Metadatos descriptivos: Su propósito es identificar cómo un recurso puede distinguirse de otro, descubrir cómo se encuentra un recurso y seleccionar recursos que cubran necesidades particulares. Este

tipo de metadato sirve también para formar colecciones de recursos similares y para realizar funciones de evaluación, relación (con otros recursos) y usabilidad (12).

Metadatos administrativos: Su fin es anotar información que facilite la gestión de los recursos, lo que incluye información sobre cuándo y cómo fue creado el recurso, quién es el responsable del acceso o de la actualización del contenido e información técnica, así como la versión de software o hardware necesarios para ejecutar dicho recurso (12).

Metadatos estructurales: Su objetivo es identificar cada una de las partes que componen el recurso, definiendo la estructura que le da forma. Por ejemplo, un libro que contiene capítulos y páginas puede etiquetarse con metadatos para identificar cada parte y la relación que guardan entre ellas. De manera similar, un curso compuesto por diferentes lecciones o actividades de aprendizaje puede etiquetarse con metadatos para identificar la estructura del contenido. Este tipo de metadatos es utilizado principalmente para el procesamiento automático de software de presentación o estilos (12).

Dentro de un entorno de aprendizaje en donde intervienen las tecnologías, si un recurso digital no tiene metadatos no puede considerarse OA ya que, en la práctica, los sistemas no pueden interpretarlo (o reconocerlo) y hacerlo realmente reutilizable. 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. Al utilizar un estándar de metadatos es posible organizarlos, localizarlos, recuperarlos, procesarlos, evaluarlos, intercambiarlos y reutilizarlos, es decir, garantizar su interoperabilidad con otros sistemas que manejen metadatos compatibles (12).

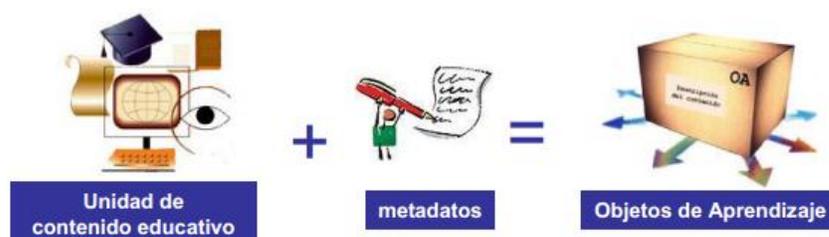


Ilustración 2: Componentes de un OA.

El uso de metadatos es un requerimiento funcional para los OA. Por ello, es sumamente importante seguir un proceso correcto para que el contenido de los metadatos sea consistente, pertinente y fácilmente interpretable para los sistemas y para el sector al que esté dirigido. El ciclo de vida para los metadatos

consiste en cuatro fases: seleccionar el esquema de marcado que se utilizará, llenar los metadatos acorde al esquema seleccionado, recuperar los metadatos y por último, (re)utilizar los metadatos (12).

1.2 Herramientas y tecnologías a utilizar para el desarrollo del componente

El centro FORTES se especializa en desarrollar tecnologías que permitan ofrecer servicios y productos para la implementación de soluciones de formación, aplicando las Tecnologías de la Información y las Comunicaciones a todo tipo de instituciones con modelos de formación y condiciones tecnológicas diferentes. En el módulo a implementar se seleccionarán un conjunto de tecnologías y herramientas atendiendo a sus características y a las nuevas políticas de desarrollo del centro FORTES en su actual migración al software libre. El módulo presentado debe tener la posibilidad de ser incorporado al marco de trabajo Xalix.

Metodología de desarrollo de software

El surgimiento de fundamentos de la metodología tiene inicios a partir de la fuerte necesidad de llevar un determinado proyecto a su meta deseada, estos fundamentos se adaptaron al desarrollo de software. La nueva etapa de adaptación contenía el desarrollo dividido en etapas de manera secuencial que permitía el mejoramiento de la necesidad existente en el campo del software.

Entre las principales metodologías tradicionales se encuentra RUP, que centra su atención en llevar una documentación exhaustiva de todo el proyecto (13).

RUP Proceso racional unificado (Rational Unified Process por sus siglas en inglés)

Proceso formal: Provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Tiene como meta principal asegurar la producción de software de alta calidad que satisfaga los requerimientos de los usuarios finales (respetando cronograma y presupuesto). Fue desarrollado por Rational Software y está integrado con toda la suite Rational de herramientas. Puede ser adaptado y extendido para satisfacer las necesidades de cualquier institución u organización que lo adopte. Está centrado en la arquitectura, se guía por casos de uso y utiliza UML como lenguaje de notación (13).

RUP está constituido por cuatro fases en su ciclo de vida:

- Fase de concepción

Tiene como propósito definir y acordar el alcance del proyecto con los patrocinadores, identificar los riesgos potenciales y producir el plan de las fases y el de iteraciones.

- Fase de elaboración

Se seleccionan los casos de uso que permiten definir la arquitectura base del sistema y que serán desarrollados en esta fase, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar.

- Fase de construcción

El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clasificar los requerimientos pendientes, administrar los cambios de acuerdo a las evaluaciones realizadas por los usuarios y se realizan las mejoras al proyecto.

- Fase de transición

El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto (14).

Esta metodología presenta un conjunto de ventajas que son las que facilitan el trabajo con la misma:

- Evaluación en cada fase que permite cambios de objetivos.
- Funciona bien en proyectos de innovación.
- Es sencillo, ya que sigue los pasos necesarios a la hora de desarrollar el software.
- Seguimiento detallado en cada una de las fases.

Tanto como ventajas también presenta un conjunto de desventajas:

- La evaluación de riesgos es compleja.
- Excesiva flexibilidad para algunos proyectos.
- Pone al cliente en ocasiones, en situaciones incómodas.
- El cliente deberá ser capaz de describir y entender a un gran nivel de detalle para poder acordar un alcance del proyecto con él (13).

Para el desarrollo del módulo se escogió esta metodología debido a que RUP es la utilizada en CRODA por política del proyecto, además provee una organización detallada con respecto a la información para su posterior utilización en el mantenimiento del sistema. Define claramente actividades realizadas por roles,

generando a su paso artefactos que sustentan el proceso de desarrollo del producto, además de dichas facilidades, utiliza UML como lenguaje de modelado del sistema, el cual es fácil de utilizar y es conocido para su implementación.

Herramienta Case

Una Herramienta CASE es aquella que permite la creación de los artefactos necesarios, siguiendo una metodología en la construcción de un software. En la utilización de la metodología RUP, se hace necesario el uso de la misma en las fases de Análisis, 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. Permite mayor calidad y rapidez del software.

Visual Paradigm 8.0 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, casos de uso y diagramas de actividades. 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, 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 (15).

Es una herramienta multi-plataforma y muy fácil de usar. Teniendo en cuenta las ventajas que ofrece la misma, se decide su utilización como Herramienta CASE en el presente trabajo.

Marco de trabajo Xalix

En el Centro FORTES se ha iniciado la formalización de un marco de trabajo en cada Línea de Productos de Software (LPS), con el propósito de disminuir la diversidad tecnológica de las soluciones y facilitarle el trabajo al cliente, permitiéndole utilizar distintas funcionalidades tanto de RHODA, ZERA como de CRODA, en un mismo marco de trabajo denominado Xalix.

Xalix cuenta con las siguientes características: **(16)**

Tecnologías

Framework de desarrollo: Symfony v2.3.7.

Lenguaje de programación para el servidor: PHP 5.4.

Lenguaje para el cliente: HTML5 (debe hacer uso de las nuevas etiquetas).

Gestor de base de datos: PostgreSQL.

Librería de CSS: Bootstrap v3.0.0 (compilado para el marco de referencia).

Librería de JavaScript: jQuery v1.10.2.

Componentes nativos de Xalix (en desarrollo): Gestión de autenticación, Gestión de servicios web, Temas y Panel de administración.

Para lograr un ágil y cómodo proceso de integración se requiere que el editor de metadatos se elabore sobre las bases de Xalix, es decir, que se implemente con tecnologías compatibles a las desarrolladas en el marco de trabajo.

Lenguajes de implementación

Actualmente existen diferentes lenguajes de programación para desarrollar en la Web, estos han ido surgiendo debido a las necesidades de las plataformas. Con las diferentes transformaciones las tecnologías han ido desarrollándose y han surgido nuevos problemas a solucionar. Esto ha dado lugar al desarrollo de lenguajes de programación para la Web dinámica que permitan interactuar con los usuarios.

PHP 5.4

Para el desarrollo del sistema se utiliza como lenguaje de programación por parte del servidor PHP ya que es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. Lo que distingue a PHP de algo como JavaScript del lado del cliente, es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El servidor web puede ser incluso configurado para que procese todos los ficheros HTML con PHP. Es extremadamente simple para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores profesionales, también es un lenguaje libre e incluye gran cantidad de funciones.

Aunque el desarrollo de PHP está centrado en programación de scripts del lado del servidor, se puede utilizar para otras funcionalidades, gracias a esta característica se pueden recopilar datos de formularios, generar páginas con contenidos dinámicos o enviar y recibir cookies. Aunque PHP puede hacer mucho más.

Ofrece un conjunto de ventajas:

- PHP puede usarse en todos los principales sistemas operativos, incluyendo Linux, muchas variantes de Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X y RISC.
- Con PHP se tiene la libertad de elegir el sistema operativo y el servidor web. Además, se tiene la posibilidad de utilizar programación por procedimientos, programación orientada a objetos (POO) o una mezcla de ambas.
- Con PHP no se está limitado a generar HTML. Entre las capacidades de PHP se incluyen la creación de imágenes, ficheros PDF e incluso películas Flash generadas sobre la marcha. También se puede generar fácilmente cualquier tipo de texto, como XHTML y cualquier otro tipo de fichero XML. PHP puede autogenerar estos ficheros y guardarlos en el sistema de ficheros en vez de imprimirlos en pantalla, creando una caché en el lado del servidor para contenido dinámico.

Una de las características más potentes y destacables de PHP es su soporte para un amplio abanico de bases de datos. PHP también cuenta con soporte para comunicarse con otros servicios como correo y directorio electrónico, supervisión de funcionamiento de la red, transferencia de noticias en red, usando protocolos tales como LDAP, IMAP, SNMP, NNTP, POP3 y HTTP (en Windows). PHP posee soporte para la instalación de objetos Java y para usarlos de forma transparente como sus propios objetos (17).

JavaScript 1.5

JavaScript es el lenguaje de programación por parte del cliente que se utiliza en el desarrollo de la aplicación. Es un lenguaje que se usa 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 (18).

Dado que JavaScript es un lenguaje de programación del lado del cliente, compatible con los navegadores modernos como Internet Explorer y Mozilla Firefox, 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. El código se integra junto a HTML brindando diferentes efectos para la interacción con los usuarios.

HTML 5 (HyperText Markup Language)

Es el lenguaje marcado de hipertexto que predomina 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 definen la sintaxis y colocan instrucciones especiales que no muestra el navegador, aunque sí indica cómo desplegar el contenido del documento. Este lenguaje se propone para la construcción de las vistas del módulo presentado (19).

CSS 3.0 (Cascading Style Sheets)

CSS es un lenguaje de estilo en cascada usado para definir la presentación de un documento estructurado, escrito casi siempre en HTML. Permite la separación de la estructura de un documento de su presentación. Algunas ventajas que ofrece 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 (19).

Lenguaje de modelado

UML

Lenguaje Unificado de Modelado o Unified Modeling Language en inglés, es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el Grupo de Gestión de Objetos u Object Management Group (OMG por sus siglas en inglés). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML. Ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del mismo, aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y

construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como RUP), pero no especifica en sí mismo qué metodología o proceso usar. UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, solo se diagrama la realidad de una utilización en un requerimiento, mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, sin embargo, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos (20).

Entorno de Desarrollo Integrado (IDE)

Un entorno de desarrollo integrado consiste básicamente en un software que previamente ha sido instalado en la máquina del cliente y cuyo principal objetivo es el desarrollo de otro software, permite mantener proyectos informáticos, los cuales se pueden implementar en diferentes lenguajes de programación, así como realizar una serie de operaciones básicas sobre ellos.

NetBeans 7.1

NetBeans IDE es un reconocido entorno de desarrollo integrado disponible para Windows, Mac, Linux y Solaris. El proyecto de NetBeans está formado por un IDE de código abierto y una plataforma de aplicación que permite a los desarrolladores crear con rapidez aplicaciones Web, empresariales, de escritorio y móviles, utilizando la plataforma Java, así como PHP, JavaScript y Ajax .El proyecto de NetBeans está apoyado por una comunidad de desarrolladores dinámica y ofrece documentación y recursos de formación exhaustivos, así como una selección diversa de complementos de terceros, además de que es gratuito y sin restricción de uso (21).

Frameworks y librerías

Un Framework puede definirse como una solución completa que contemplan herramientas de apoyo para el desarrollo y/o implementación de una aplicación. Además representa una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación a la que se le puede añadir elementos para construir una aplicación correcta (22).

Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones. Hoy en día la utilización de un framework es indispensable a la hora de realizar algún tipo de software o aplicación ya que este brinda al desarrollador ciertas ventajas que facilitan el trabajo a realizar.

Symfony 2.3.7

Symfony es un framework PHP que facilita el desarrollo de aplicaciones Web, es considerado el mejor framework para crear estos tipos de aplicaciones y ha sido probado en alguno de los sitios Web más grandes del mundo como por ejemplo Yahoo! Answers, Dailymotion, Delicious, además es el framework más documentado del mundo, ya que cuenta con miles de páginas de documentación distribuidas en varios libros gratuitos y decenas de tutoriales (23).

Es fácil de instalar y configurar en sistemas Windows y Linux. Funciona con todas las bases de datos comunes (MySQL, PostgreSQL, SQLite, Oracle, MS SQL Server). Compatible solamente con PHP 5 desde hace años, para asegurar el mayor rendimiento y acceso a las características más avanzadas de PHP. Traducido a más de 40 idiomas y fácilmente traducible a cualquier otro idioma.

Symfony2 cuenta con un sistema de bundles, que técnicamente, es un directorio que contiene todo tipo de archivos dentro de una estructura jerarquizada de directorios. Estos suelen contener clases PHP y archivos web (JavaScript, CSS e imágenes). Además pueden ser utilizados tantas veces como divisiones lógicas tenga la aplicación.

JQuery 1.10.2

Es un framework para el lenguaje JavaScript el cual simplificará la vía para programar en este lenguaje. JQuery es un producto con una aceptación por parte del programador muy buena y un grado de penetración en el mercado muy amplio, lo que hace suponer que es una de las mejores opciones. Además, es un producto estable, bien documentado y con un gran equipo de desarrolladores a cargo de la mejora y actualización del framework, todas las ventajas que presenta el framework sin duda son de agradecer ya que se obtienen de manera gratuita, porque tiene licencia para uso en cualquier tipo de plataforma, personal o comercial. Implementa una serie de clases (de programación orientada a objetos) que permiten programar sin preocupaciones del navegador que está utilizando el usuario, ya que funcionan de forma exacta en todas las plataformas más habituales (24).

Esta librería de JavaScript ofrece una infraestructura con la que se tendrá mucha mayor facilidad para la creación de aplicaciones complejas del lado del cliente. Se obtendrá ayuda en la creación de interfaces de usuario y efectos dinámicos. Cuando se programe JavaScript con jQuery se tendrá disposición de una interfaz para programación que permita hacer diferentes funcionalidades con el navegador y que funcionen para todos los usuarios visitantes (24).

Capítulo 1: Análisis de conceptos, herramientas y tecnologías

Simplemente se debe conocer las librerías del framework y programar utilizando las clases, sus propiedades y métodos para la consecución de los objetivos (24).

Capítulo 2: Características y modelación de la propuesta de solución

Introducción

Este capítulo constituye la propuesta de solución para lograr la edición de metadatos basado en el esquema LOM en el marco de trabajo Xalix. Para la descripción de la solución se desarrolla el modelo de dominio, donde se detallan las clases y relaciones que lo conforman. Se realiza la especificación de los requerimientos funcionales y no funcionales que debe presentar el sistema, así como el modelo de diseño del mismo. Además, se identifican y describen los casos de uso del sistema y sus actores. De manera general el capítulo recoge el flujo de análisis y diseño del sistema a desarrollar.

2.1 Modelo de dominio o modelo conceptual

Un modelo de negocio se realiza con la finalidad de describir cada proceso del negocio, especificando sus datos, actividades (o tareas), roles (o agentes) y reglas de negocio. Para conseguir sus objetivos, una empresa organiza su actividad por medio de un conjunto de procesos de negocio. Cada uno de ellos se caracteriza por una colección de datos que son producidos y manipulados mediante un conjunto de tareas, en las que ciertos agentes (trabajadores o departamentos) participan de acuerdo a un flujo de trabajo determinado. Además, estos procesos se hallan sujetos a un conjunto de reglas de negocio, que determinan la estructura de la información y las políticas de la empresa (25).

En el estudio realizado del problema planteado para este trabajo, no fue posible identificar procesos del negocio, ya que está centrado en tecnologías informáticas, pues el objetivo del trabajo es la creación de un editor de metadatos para el marco de trabajo Xalix que permita incorporar las funcionalidades del editor de metadatos presentes hoy en CRODA, con el propósito de lograr la descripción de recursos educativos, haciéndose difícil determinar actores y procesos de negocio. Por lo que se hace necesaria la realización del modelo de dominio.

El modelo de dominio o modelo conceptual es una representación visual estática del entorno real donde se desarrollará el sistema.

El modelo del dominio muestra (a los modeladores) clases conceptuales significativas en un dominio de problema; es el artefacto más importante que se crea durante el análisis orientado a objetos (26).

Diagrama de clases del modelo de dominio

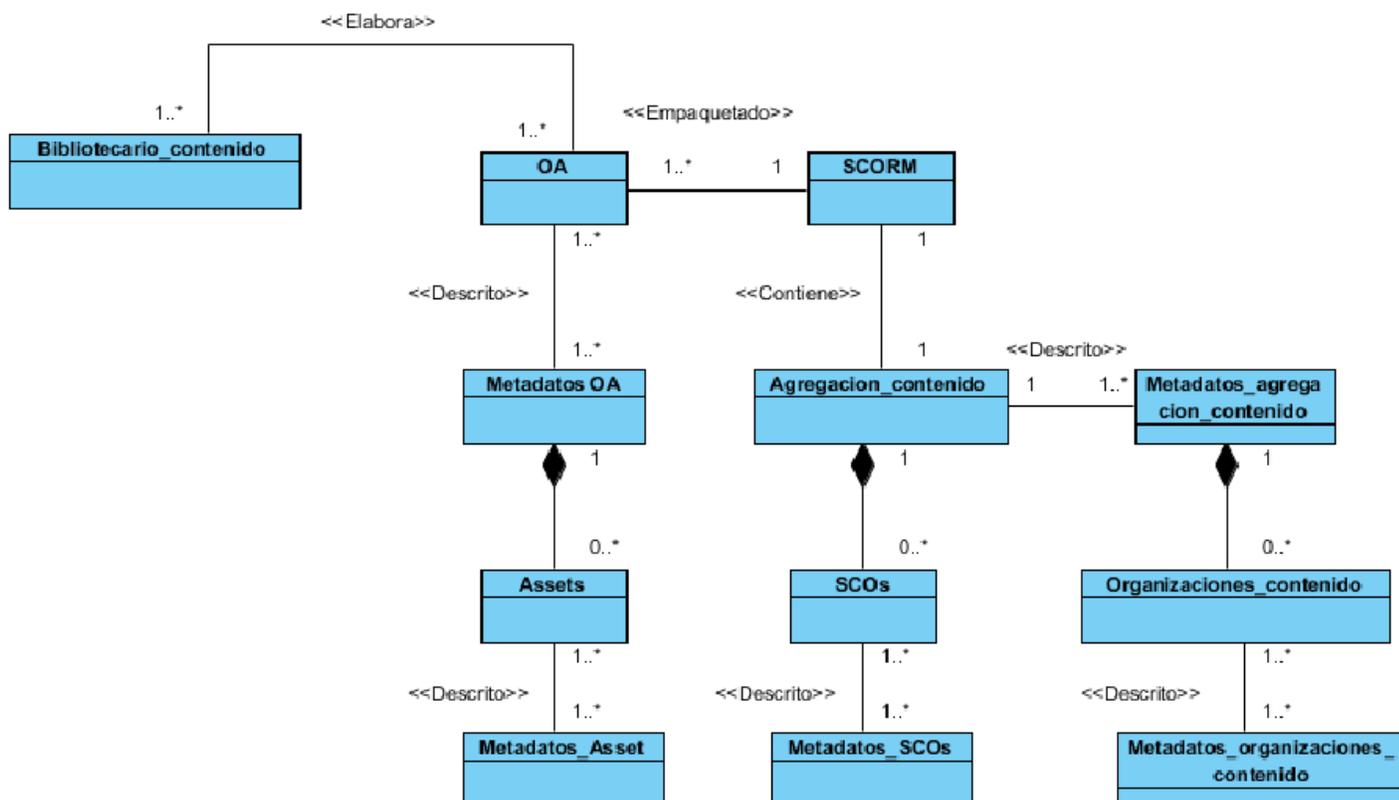


Ilustración 3: Modelo de dominio.

Conceptos de las clases del modelo de dominio

Bibliotecario de contenido: Es la persona encargada de crear los OA en la herramienta de autor CRODA.

OA: Cualquier recurso digital que puede ser usado como soporte para el aprendizaje (2). Pueden ser localizados a través de los metadatos que los describen.

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

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

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

Assets: Recursos o elementos básicos, como ficheros de texto, audio, video, etc. Estos recursos básicos se agrupan en los SCOs (8).

Metadatos de Assets: Conjunto de elementos necesarios que describen a los assets independientemente de su contexto.

SCOs (Sharable content object): Curso o componente de un curso que cumple con los requisitos de interoperabilidad, durabilidad y que dispone de la información suficiente para poder ser reutilizado y accesible (8).

Metadatos de SCOs: Conjunto de elementos de información necesarios que describen a un SCO independientemente de su contexto.

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

Metadatos de organización: Conjunto de elementos de información necesarios que describen una organización independientemente de su contexto.

Descripción del diagrama de clases del modelo de dominio

El bibliotecario de contenido es un rol autenticado en CRODA que permite elaborar uno o más OA empaquetados siguiendo el estándar SCORM, para posibilitar el intercambio de los contenidos entre sistemas. Este estándar contiene en su modelo de contenidos un conjunto de componentes, la agregación de contenido compuesta por Asset, SCOs y organizaciones de contenido, cada uno de estos componentes se describen individualmente con metadatos con el objetivo de facilitar su búsqueda y utilización.

Como se puede apreciar en el diagrama, la descripción del OA a partir de los metadatos es realizada por el bibliotecario de contenido, por lo cual las personas que asumen este rol, tienen como actividad principal la creación del OA, lo cual implica tener tiempo y conocimiento pedagógico y la actividad de describir estos, la cual requiere de la intervención de expertos en la catalogación, para obtener una mejor descripción. En CRODA no existe un rol que se encargue específicamente de la catalogación de los recursos, que conjuntamente con los autores asegure el desarrollo y gestión de metadatos precisos y efectivos.

2.2 Requerimientos del sistema

La ingeniería de requerimientos es un proceso trabajoso, extenso, el cual precisa de un análisis intenso del sistema a desarrollar, pues este provee un cambio en el entorno y las relaciones entre las personas implicadas en el mismo, haciéndose necesario la identificación de dichas personas, considerando sus necesidades y asegurando que entiendan todo lo relacionado al proceso.

Requisitos funcionales del sistema

Los requerimientos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que este debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares (27).

R1 Insertar metadatos en forma de formulario: el sistema debe permitir al usuario insertar los elementos de información en forma de formulario.

R2 Insertar metadatos en forma de árbol: el sistema debe permitir al usuario insertar elementos de información en forma de árbol.

R3 Permitir interactividad entre las vistas de árbol y formulario: el sistema debe ser capaz de reflejar en la vista de árbol todo lo que el autor modifique en la vista de formularios y viceversa.

R4 Guardar cambios efectuados en los metadatos: el sistema debe brindar la posibilidad al usuario de guardar los cambios efectuados en los metadatos.

R5 Deshacer cambios efectuados en los metadatos: el sistema debe brindar la posibilidad al usuario de deshacer los cambios efectuados en los metadatos, borrando los cambios introducidos por el usuario que no hayan sido guardados.

R6 Mostrar todos los metadatos: el sistema debe mostrar al usuario todos los elementos del esquema LOM.

R7 Mostrar ayuda para especificar los elementos de información: el sistema debe ser capaz de mostrar al usuario una ayuda sobre los elementos de información que brinda el esquema de metadatos LOM. Al lado de cada elemento de información aparecerá un ícono, cada vez que el usuario de un clic sobre el mismo, el sistema debe mostrar el significado del elemento y un pequeño ejemplo.

R8 Exportar archivo XML de los metadatos: el sistema debe brindar al usuario la posibilidad de exportar los metadatos en un archivo XML, a una dirección especificada.

R9 Importar archivo XML de los metadatos: el sistema debe brindar al usuario la posibilidad de importar los metadatos expresados en un archivo XML, desde una dirección especificada.

R10 Publicar metadatos: el sistema debe posibilitar al usuario publicar una instancia de metadatos, para

que pueda ser visualizada y utilizada por otros usuarios.

R11 Listar metadatos publicados: el sistema debe posibilitar al usuario listar todas las publicaciones de metadatos que han sido realizadas.

R12 Visualizar metadatos publicados: el sistema debe permitir al usuario visualizar cada instancia de metadatos que ha sido publicada, mostrando cada elemento de información con su valor correspondiente.

R13 Reutilizar metadatos: el sistema debe permitir al usuario reutilizar cada instancia de metadatos que ha sido publicada, cargándola en la instancia que está siendo editada por el usuario.

R14 Buscar metadatos: el sistema debe permitir al usuario realizar una búsqueda de metadatos y/o los elementos de información que el usuario desee encontrar.

R15 Eliminar metadatos publicados: el sistema debe permitir al usuario eliminar la instancia de metadatos que elija una vez que esté publicada.

Requisitos no funcionales del sistema

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

- Apariencia o interfaz externa

La interfaz no debe contener gran cantidad de imágenes ni otros medios tales como videos que limiten la rápida interactividad con la navegación. El diseño de interfaz de este módulo debe ajustarse al de CRODA 2.0, en general ha de tener un estilo formal y con una navegación sencilla y sugerente, además de mostrar de manera legible y concisa la información y los textos.

-Soporte

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

-Usabilidad

El módulo podrá ser usado por cualquier persona que posea un nivel básico de conocimientos de computación. El mayor peso recae en el rol bibliotecario de contenidos ya que debe poseer conocimientos específicos de los elementos del esquema de metadatos LOM y sobre el estándar SCORM para

desarrollar sus tareas en la herramienta. Debe presentar una interfaz especial, amigable, fácil de trabajar para usuarios no expertos en informática, ya que los creadores de OA mayormente son profesores y muchos no tienen un buen conocimiento computacional.

2.3 Identificación de actores del sistema

Un actor es un ente real que se comunica con la aplicación o producto y que es externo al sistema. La siguiente tabla muestra la relación de los actores que interactúan en el módulo.

Tabla 1: Identificación de los actores del sistema.

Actor	Descripción
Bibliotecario de contenidos	Es el rol encargado de describir los recursos educativos cuando solicite la edición de los metadatos.

2.4 Identificación de casos de uso del sistema

Un caso de uso es un escenario que describe cómo el software va a ser utilizado en determinada situación.

Los casos de uso deben definir los requisitos funcionales y operativos del sistema, diseñando un escenario de uso acordado por el usuario final y el equipo de desarrollo. También deben proporcionar una descripción clara y sin ambigüedades de cómo el usuario final interactúa con el sistema y viceversa, además de proporcionar una base para la validación de las pruebas (28).

Los casos de uso identificados para el módulo presentado son los siguientes:

1. Gestionar metadatos.
2. Exportar archivo XML.
3. Importar archivo XML.
4. Publicar metadatos.
5. Eliminar metadatos publicados.
6. Reutilizar metadatos.
7. Buscar metadatos.

Diagrama de los casos de uso identificados en el sistema

Luego de identificar los casos de uso y actores del sistema, se realiza una representación gráfica de los procesos (casos de uso) y su interacción con el actor identificado. Los casos de uso se encuentran en correspondencia con el actor del sistema por lo que deben encontrarse comunicados con dicho actor.

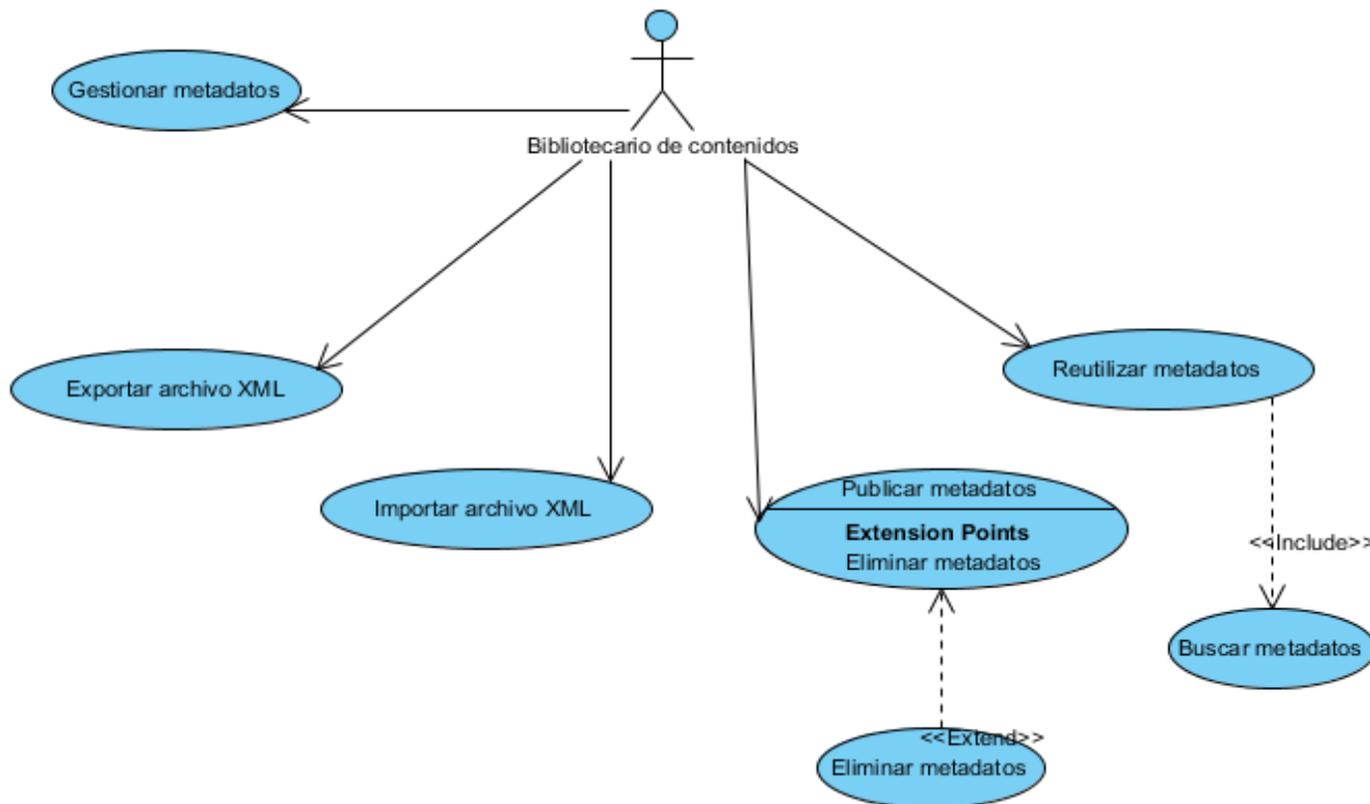


Ilustración 4: Diagrama de casos de uso del sistema.

Descripción textual de los casos de uso identificados

La descripción de los casos de uso (CU) permite documentar en un lenguaje cliente/usuario una lista de pasos que debe seguir el actor para interactuar con el sistema.

A continuación se muestra la descripción del caso de uso Gestionar metadatos, los restantes se pueden encontrar en el **Anexo 1**.

Tabla 2: Descripción del caso de uso Gestionar metadatos.

CU-1	Gestionar metadatos.
Actor	Bibliotecario de contenidos (generalización de Autor).

Capítulo 2: Características y modelación de la propuesta de solución

Resumen	El caso de uso se inicia cuando el actor decide insertar todos los metadatos, modificar y deshacer los cambios efectuados, consultar la ayuda y finaliza cuando realiza estas acciones.	
Referencias	R1, R2, R3, R4, R5, R6, R7.	
Prioridad	Crítico.	
Precondiciones	El actor debe encontrarse en la generación o edición de un OA. El actor debe encontrarse en la edición de los metadatos.	
Poscondiciones	Todos los metadatos fueron insertados correctamente, se deshicieron o guardaron los cambios efectuados y/o se consultó el sistema de ayuda.	
Flujo Normal		
Acción del actor	Respuesta del sistema	
1. El actor accede a la edición de los metadatos.	<p>2. El sistema genera una nueva ventana, mostrando los metadatos establecidos por LOM y por el bibliotecario de contenidos, en una vista en forma de árbol y una vista en forma de formulario mostrando íconos diferentes en cada caso y muestra un ícono de ayuda al lado de cada elemento.</p> <p>Además, el sistema muestra al usuario los siguientes elementos de información completados automáticamente solamente para el recurso educativo en su totalidad (es decir, la Agregación de Contenido o el paquete).</p> <p>Categoría General:</p> <ul style="list-style-type: none"> -Estructura: Jerárquica. -Identificador: IDE-MD5 (time ()). -Entrada: número aleatorio que se genera continuamente. 	

Categoría ciclo de Vida:

-Tipo: Autor y Creador.

-Fecha

-Estado: Borrador.

-Versión: 2.0

-Entidad: correo electrónico del generador y el creador.

Categoría Meta-Metadatos:

-contribución (Tipo): Si se solicita una edición al bibliotecario de contenido, en el elemento Tipo debe aparecer el rol Bibliotecario de contenido.

-Esquema de metadatos: LOM V2.0

-Fecha

-Identificador

-Entrada

Categoría Técnica:

-Formato: para el Objeto será Aplicación ZIP.

-Requisitos (Tipo): Sistema operativo.

-Nombre: Multi-SO.

-Tamaño

Categoría Uso educativo:

-Destinatario: por defecto tendrá el valor Aprendiz.

También el sistema permitirá al usuario:

-Insertar todos los metadatos (Ver sección 1).

-Mostrar sistema de ayuda. (Ver sección 2).

Capítulo 2: Características y modelación de la propuesta de solución

3. El actor inserta los valores de los metadatos, en la vista de árbol o en la vista de formulario.	
4. El actor accede a la opción Guardar.	5. El sistema guarda los valores introducidos por el usuario y muestra los mismos en ambas vistas y finaliza el caso de uso.
Flujo Alterno	
Acción del actor	Respuesta del sistema
4. El actor accede a la opción Deshacer.	5. El sistema deshace los cambios que no han sido guardados por el usuario y muestra los mismos en ambas vistas y finaliza el caso de uso.
Sección 1: Insertar todos los metadatos.	
Acción del actor	Respuesta del sistema
3. El actor accede a la opción Mostrar todos.	4. El sistema muestra en ambas vistas todos los metadatos del esquema LOM, diferenciando los establecidos por el bibliotecario de contenidos.
5. El actor inserta los valores de los metadatos que desee y accede a la opción Guardar.	6. El sistema guarda los valores introducidos por el usuario y finaliza el caso de uso.
Flujo Alterno	
Acción del actor	Respuesta del sistema
5. El actor deja en blanco algún elemento y accede a la opción Guardar.	6. El sistema muestra el mensaje de aviso: "Debe completar todos los elementos".
Sección 2: Consultar sistema de ayuda.	
Acción del actor	Respuesta del sistema
3. El actor accede al ícono de ayuda al lado de cualquier elemento de información.	4. El sistema muestra la explicación y un breve ejemplo del elemento de información.
Prototipo de interfaz de usuario	

The screenshot shows a web-based form for managing LOM (Learning Object Metadata) data. The interface is organized into a sidebar and a main content area. The sidebar on the left contains a tree view of metadata categories: General, Ciclo de Vida, Meta-metadatos, Técnica, Uso educativo, Derechos, Relación, Anotación, and Clasificación. The main content area is titled 'General' and contains several input fields with question marks: Identificador, Catálogo, Entrada, Título, Idioma (dropdown), Descripción (text area), Palabras claves, Ámbito (dropdown), Estructura (dropdown), and Nivel de Agregación (dropdown). There are also buttons for 'Guardar' and 'Deshacer' at the top left.

2.5 Diagramas de colaboración

Los diagramas de colaboración (DC) son los artefactos generados correspondientes al flujo de trabajo análisis y diseño, permiten mostrar las interacciones entre los objetos del análisis, creando enlaces y añadiendo mensajes entre ellos.

En los DC se utilizan un conjunto de estereotipos estandarizados en UML que permiten describir el sistema (29).

Clases de Interfaz (CI): Se utilizan para modelar la interacción entre el sistema y sus actores. Esta interacción a menudo implica recibir y presentar información y peticiones de y hacia los usuarios y los sistemas externos. Las CI representan abstracciones de ventanas, formularios, paneles, interfaces de comunicaciones, de impresoras, sensores y terminales.

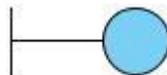


Ilustración 5: Estereotipo que representa la CI.

Clases de Entidad (CE): Se utilizan para modelar información que posee larga vida y que es a menudo persistente. Modelan la información y el comportamiento asociado de algún fenómeno o concepto, como una persona, un objeto o suceso del mundo real. Suelen mostrar una estructura de datos lógica y contribuyen a comprender de qué información depende el sistema.

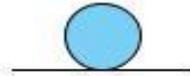


Ilustración 6: Estereotipo que representa la CE.

Clases de Control (CC): Representan coordinación, secuencia, transacciones y control de otros objetos y se usan con frecuencia para encapsular el control de un caso de uso en concreto. Los aspectos dinámicos del sistema se modelan con CC debido a que ella maneja y coordina las acciones y delega acciones a otros objetos, es decir, objetos de interfaz y de entidad.



Ilustración 7: Estereotipo que representa la CC.

A continuación se representa una muestra de los diagramas de colaboración correspondientes a la propuesta de solución, el resto de los diagramas se encuentran en el **Anexo 2**.

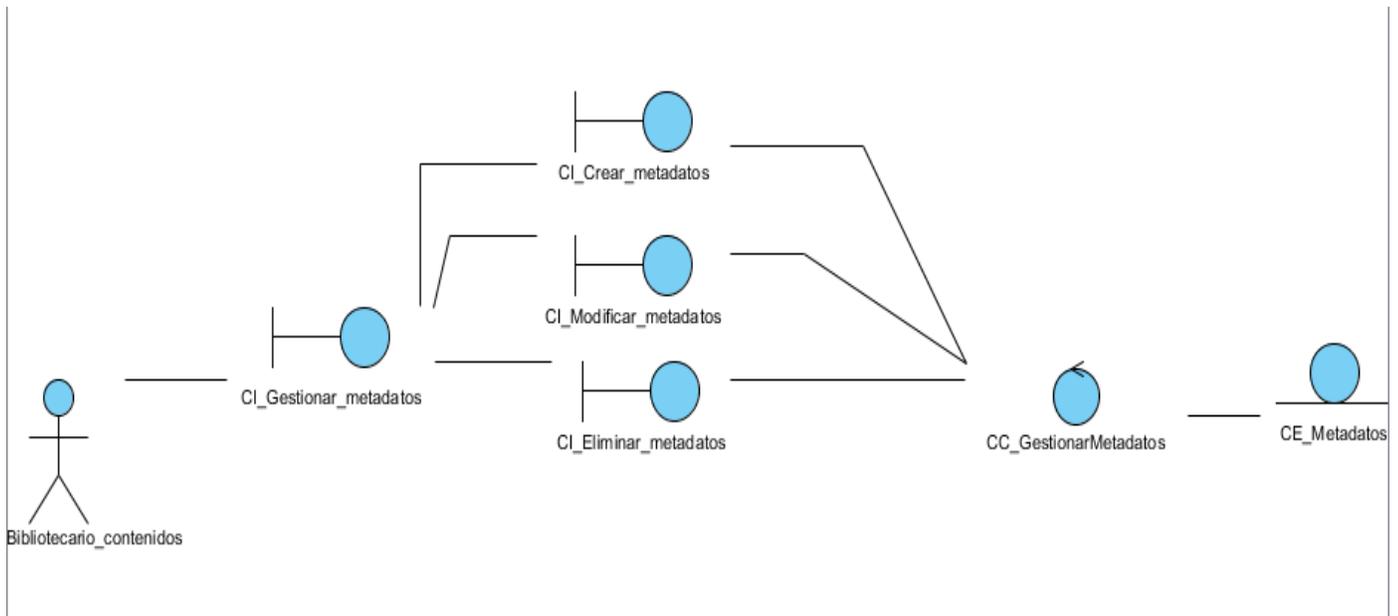


Ilustración 8: Diagrama de colaboración del caso de uso Gestionar metadatos.

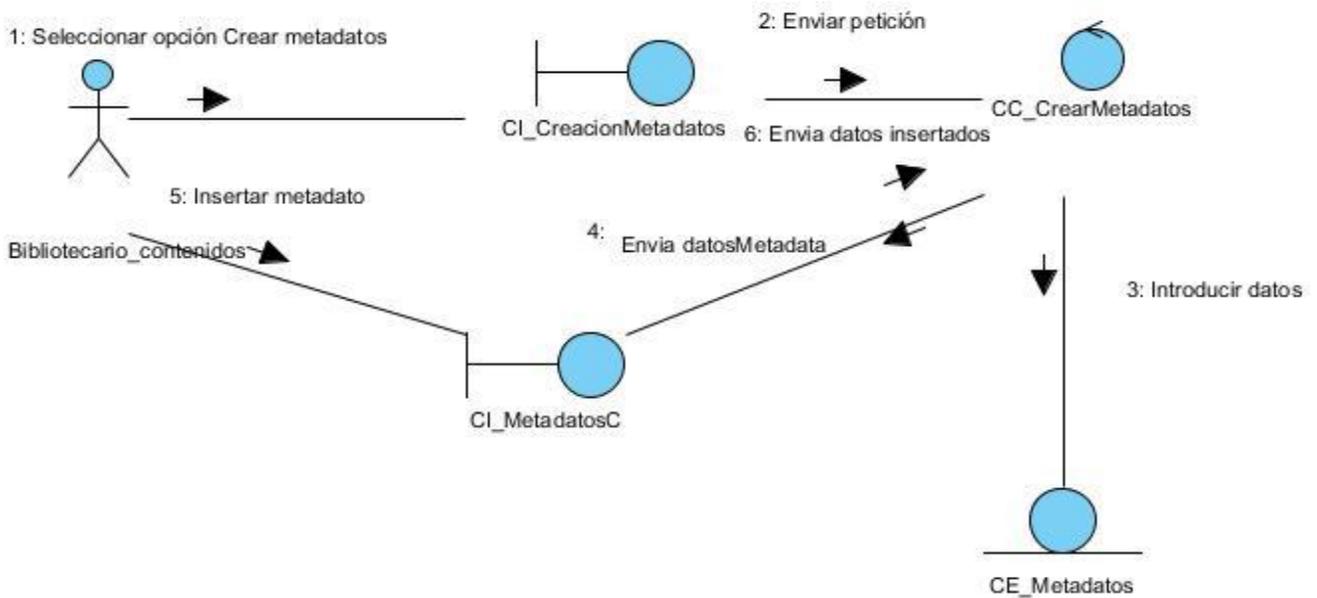


Ilustración 9: Diagrama de colaboración del escenario Crear metadatos.

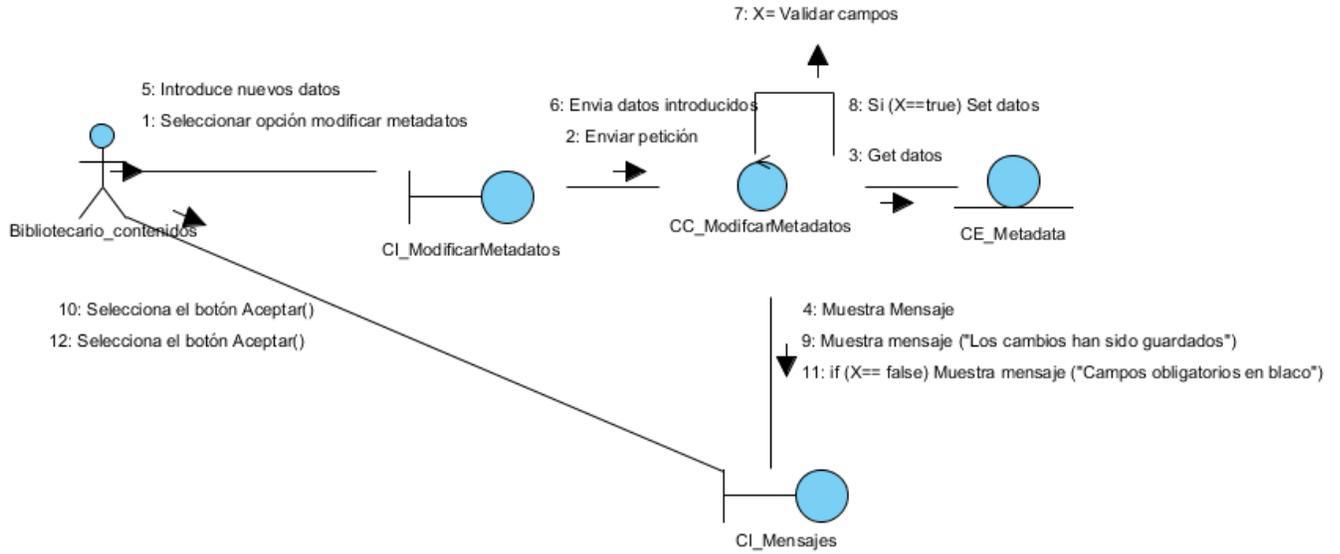


Ilustración 10: Diagrama de colaboración del escenario Modificar metadatos.

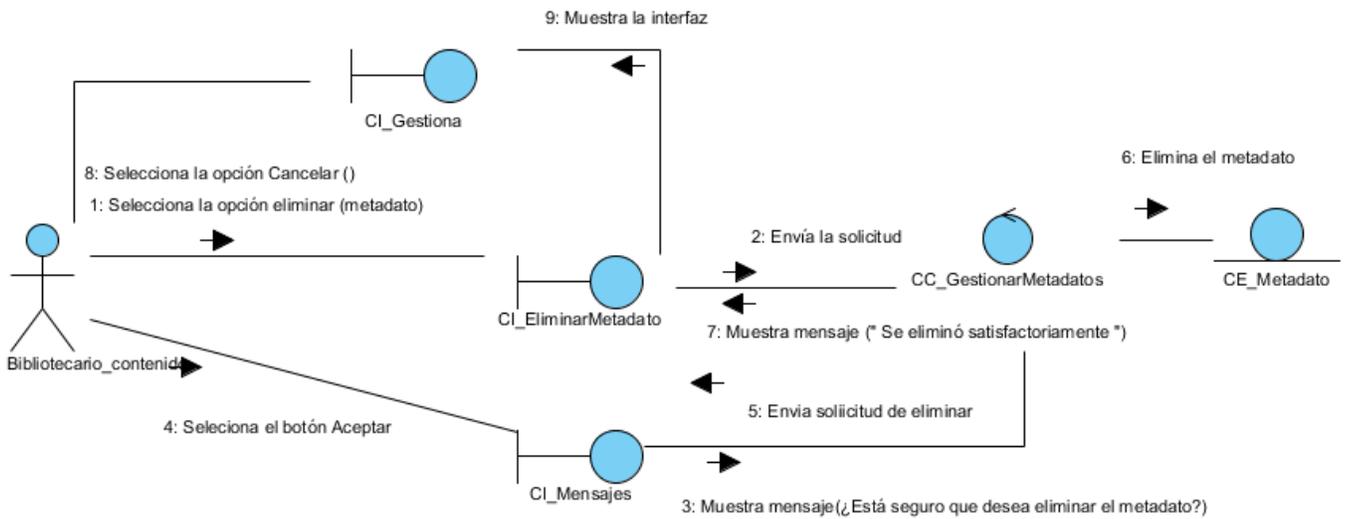


Ilustración 11: Diagrama de colaboración del escenario Eliminar metadatos.

2.6 Modelo de Diseño

El modelo del diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación (29).

Con el modelo de diseño es posible comprender de una forma más precisa los aspectos relacionados a los requisitos funcionales y no funcionales, lo que constituye un punto de entrada en la implementación del sistema.

2.6.1 Patrones del Diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Proporcionan una estructura conocida por todos los programadores, de manera que la forma de trabajar no resulte distinta entre los mismos, así la incorporación de un nuevo programador, no requerirá conocimiento de lo realizado anteriormente por otro. Permiten tener una estructura de código común a todos los proyectos que implemente una funcionalidad genérica. La utilización de patrones de diseño, permite ahorrar grandes cantidades de tiempo en la construcción de software.

Symfony es un framework que sigue la mayoría de los patrones de diseño para la web dentro de los que se encuentran los patrones **GOF** y **GRASP**.

Los patrones **GOF** se dividen en tres categorías, **los patrones de creación** que permiten la inicialización y configuración de objetos, **los patrones estructurales** que se encargan de separar la interfaz de la implementación y **patrones de comportamiento** que describen la comunicación entre objetos o clases (30).

Los patrones **GOF** utilizados en la aplicación se definen a continuación:

Patrones de creación (31)

Builder (Constructor). Separa la construcción de un objeto complejo de su representación, de forma que el mismo proceso de construcción pueda crear diferentes representaciones. Una clase superior Builder se aplica en la creación de clases menores cuando sea necesario crear y agregar instancias de clases, cuando sea necesario contener múltiples instancias de clases y cuando la clase superior dispone de los datos necesarios para la clase a instanciar. Reduce el acoplamiento. Permite variar la representación interna de estructuras complejas, respetando la interfaz común de la clase Builder.

Evidencia en la propuesta de solución: Clase DefaultController.php.

Patrones estructurales (31)

Proxy (Apoderado). Proporciona un sustituto o representante de otro objeto para controlar el acceso a éste. Puede mejorar la eficiencia al controlar la instanciación de recursos. Aumenta la seguridad. Los clientes se desentienden de la ubicación de los componentes accedidos.

Evidencia en la propuesta de solución: Clase sfAbstractLom.php.

Patrones de comportamiento (31)

Mediator (Mediador). Define un objeto que encapsula cómo interactúa un conjunto de objetos. Promueve un bajo acoplamiento al evitar que los objetos se refieran unos a otros explícitamente y permite variar la interacción entre ellos de forma independiente. Se aplica cuando se tienen varias clases relacionadas entre sí y existen muchas dependencias entre ellas, cuando las clases son difíciles de reutilizar porque existen muchas relaciones de interdependencia que los hacen inoperativos por sí solos.

Evidencia en la propuesta de solución: Clase sfAbstractLom.php con la utilización de la clase abstract_lom.yml.

Los Patrones de Asignación de Responsabilidades (**GRASP**) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (32).

En el desarrollo de la aplicación se utilizaron los siguientes tipos de patrones GRASP:

1. Experto: Es un patrón que se usa más que cualquier otro a asignar responsabilidades, es un principio básico que suele utilizarse en el diseño orientado a objetos. Con él no se pretende designar una idea oscura ni extraña; expresa simplemente la "intuición" de que los objetos hacen cosas relacionadas con la información que poseen. Ofrece una analogía con el mundo real. Se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. Esto soporta un bajo acoplamiento, lo que favorece al hecho de tener sistemas más robustos y de fácil mantenimiento. El comportamiento se distribuye entre las clases que cuentan con la información requerida, alentando con ello definiciones de clase sencillas y más cohesivas, que son más fáciles de comprender y de mantener. Así se brinda soporte a una alta cohesión (32).

Evidencia en la propuesta de solución: Clase DefaultController.php se especializa exclusivamente en modelar el negocio de la edición de metadatos basados en el esquema LOM.

2. Creador: Guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento. Al escogerlo como creador, se da soporte al bajo acoplamiento, lo cual supone menos dependencias respecto al mantenimiento y mejores oportunidades de reutilización. Es probable que el acoplamiento no aumente, pues la clase creada tiende a ser visible a la clase creador, debido a las asociaciones actuales que conllevan a elegirla como el parámetro adecuado (32).

Evidencia en la propuesta de solución: Clase DefaultController.php con la creación de instancias de sfAbstractLom.php.

3. Controlador: Mayor potencial de los componentes reutilizables. Garantiza que la empresa o los procesos de dominio sean manejados por la capa de los objetos del dominio y no por la de la interfaz. Un diseño de interfaz como controlador reduce la posibilidad de reutilizar la lógica de los procesos del dominio en aplicaciones futuras, por estar ligada a una interfaz determinada que rara vez puede utilizarse en otras aplicaciones. Permite reflexionar sobre el estado del caso de uso (32).

Evidencia en la propuesta de solución: Clase DefaultController.php controla el negocio de la edición de metadatos basados en el esquema LOM dentro del componente desarrollado.

4. Alta cohesión: Presenta semejanzas con el mundo real. Mejora la claridad y la facilidad con que se entiende el diseño. Se simplifican el mantenimiento y las mejoras en funcionalidad. A menudo se genera un bajo acoplamiento. La ventaja de una gran funcionalidad es que soporta una mayor capacidad de reutilización, porque una clase muy cohesiva puede destinarse a un propósito muy específico (32).

Evidencia en la propuesta de solución: Clase DefaultController.php y la relación con sfAbstractLom.php y DOMDocument.php.

5. Bajo acoplamiento: Soporta el diseño de clases más independientes, que reducen el impacto de los cambios, son más reutilizables y acrecientan la oportunidad de una mayor productividad. No puede considerarse en forma independiente de otros patrones como Experto o Alta Cohesión, sino que más bien ha de incluirse como uno de los principios del diseño que influyen en la decisión de asignar responsabilidades. No se afectan por cambios de otros componentes. Fáciles de entender por separado y de reutilizar (32).

Evidencia en la propuesta de solución: Clase sfAbstractLom.php, pues con la asignación de responsabilidades no afecta el funcionamiento de su relación con DefaultController.php.

2.6.2 Patrones arquitectónicos

Definen la estructura de un sistema software y se componen de subsistemas con sus responsabilidades, también tienen una serie de directivas para organizar los componentes del mismo sistema, con el objetivo de facilitar la tarea del diseño de tal sistema.

Patrón arquitectónico Modelo - Vista – Controlador

Define la organización independiente del **Modelo** (Objetos de Negocio), la **Vista** (interfaz con el usuario u otro sistema) y el **Controlador** (controlador del workflow de la aplicación) (33).

El framework Symfony2, propuesto para la implementación de este módulo, está basado en este patrón conformado por tres niveles:

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

Para entender cómo funciona el patrón Modelo Vista Controlador, se debe entender la división a través del conjunto de estos tres elementos y la comunicación de estos componentes entre ellos mismos y con otras vistas y controladores externos al modelo principal. Para ello, es importante saber que el controlador interpreta las entradas del usuario (tanto teclado como el ratón), enviando el mensaje de acción al modelo y a la vista para que se proceda con los cambios que se consideren adecuados (33).

Diagrama de Clases del Diseño

Los diagramas de clases del diseño (DCD) muestran el diseño estático a través de las clases, subsistemas y relaciones que participan. A continuación se define el siguiente diagrama de diseño del sistema:

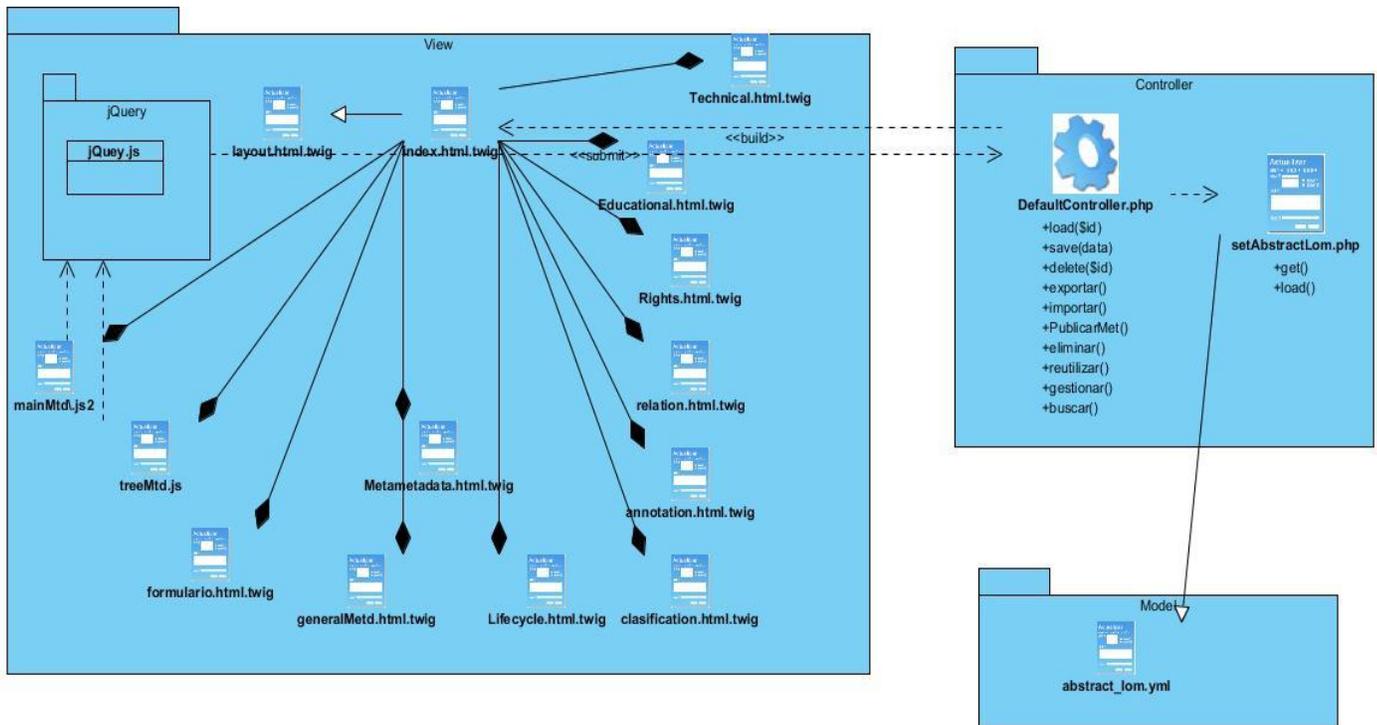


Ilustración 12: Diagrama de clases de diseño del sistema.

Capítulo 3: Implementación y validación de la solución propuesta

Introducción

En el desarrollo del presente capítulo se realizará la descripción de la implementación en términos de componentes del modelo de diseño.

Un aspecto de vital importancia en el desarrollo del software son las pruebas debido a que propician la obtención de buenos resultados. Se ejecutan con el objetivo de revisar que el software tenga el nivel de calidad requerido. Permiten detectar errores durante su funcionamiento. 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 durante el desarrollo del presente capítulo.

3.1 Diagrama de componentes

Los diagramas de componentes permiten visualizar con más facilidad la estructura general del sistema y el comportamiento del servicio que estos componentes proporcionan y utilizan a través de las interfaces.

Se puede usar un diagrama de componentes para describir un diseño que se implemente en cualquier lenguaje o estilo, solo es necesario identificar los elementos del diseño que interactúan con otros elementos del diseño a través de un conjunto restringido de entradas y salidas. Los componentes pueden tener cualquier escala y pueden estar interconectados de cualquier manera (30).

A continuación se muestra el diagrama de componentes del sistema:

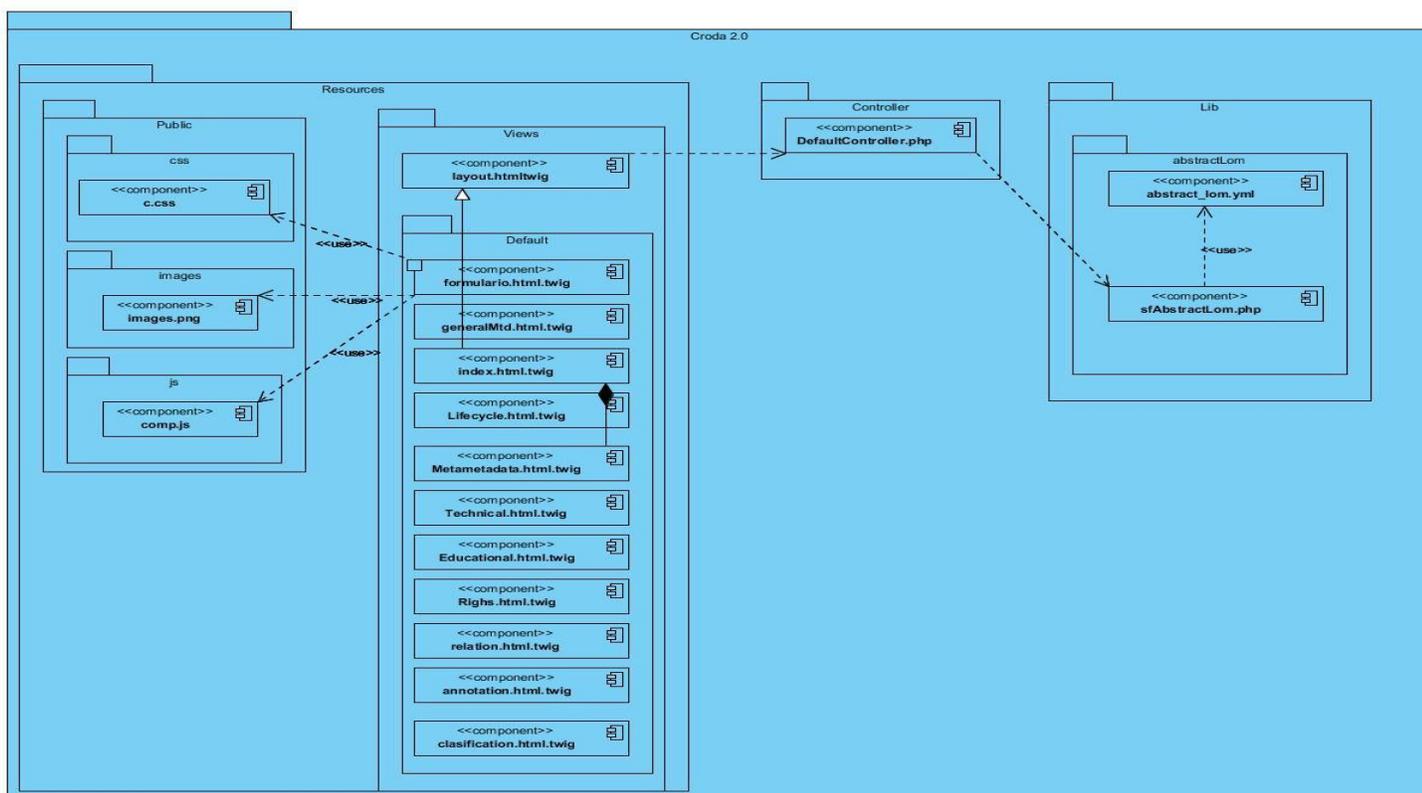


Ilustración 13: Diagrama de Componentes.

El diagrama de componentes está compuesto por los componentes definidos a continuación:

El paquete css contiene los estilos de cada uno de los componentes html.twig.

El paquete images contiene almacenada todas las imágenes del sistema.

El paquete js contiene todas las funcionalidades JavaScript con la utilización del framework jQuery.

En el paquete Default se encuentran todos los componentes referentes a las páginas distribuidas por cada una de las categorías correspondientes al esquema LOM.

El paquete controlador posee toda la lógica del negocio para la edición de metadatos basado en el esquema LOM.

El paquete abstractLom posee algunas funcionalidades que permiten el trabajo con el modelo generado por el esquema LOM (XML), con la utilización del abstractLom.yml como componente de referencia al XML.

3.2 Métodos de validación

Actualmente existen métodos con diferentes enfoques que posibilitan la validación del software. Para la presente investigación se utilizan las pruebas de integración y de funcionalidad y la técnica de ladov para saber el nivel de satisfacción grupal.

3.2.1 Pruebas de integración

Son aquellas que se realizan en el ámbito del desarrollo de software una vez que se han realizado las pruebas unitarias. Únicamente se refieren a la prueba o pruebas de todos los elementos unitarios que componen un proceso, hecha en conjunto, de una sola vez. El objetivo es verificar el correcto ensamblaje entre los distintos componentes una vez que han sido probados unitariamente con el fin de comprobar que interactúan correctamente a través de sus interfaces tanto internas como externas, cubren la funcionalidad establecida y se ajustan a los requisitos no funcionales especificados (34).

3.2.2 Pruebas de funcionalidad

La realización de pruebas es un elemento necesario 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 son aplicadas en las diferentes fases del desarrollo del software, con el objetivo de ir limando 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.

Pruebas de Caja Blanca: Se basan en un examen cercano al detalle procedimental. Se prueban las rutas lógicas del software y la colaboración entre componentes, al proporcionar casos de pruebas que ejerciten conjuntos específicos de condiciones, bucle o ambos. Se realizan al inicio del proceso de prueba (28).

Pruebas de Caja Negra: Se concentran en los requisitos funcionales del software, se aplican en su interfaz. Una prueba de este tipo examina algún aspecto funcional de un sistema que tiene poca relación con la estructura lógica interna del software. Tienden a aplicarse durante las últimas etapas de la prueba (28).

En la validación de la solución propuesta se utilizaron las pruebas de caja blanca y caja negra. En el caso de las pruebas de caja blanca se utilizaron las pruebas de unidad que se hacen durante el desarrollo, las cuales no se documentan y en las de caja negra se realizaron las pruebas orientadas a los requisitos funcionales del software con 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 permite la

creación de juegos de datos para lograr probar a profundidad todas las funcionalidades del software y la identificación de los diseños de casos de pruebas (DCP) (28).

Caja Negra

Mediante las descripciones de los 7 casos de uso (CU) se generaron los DCP correspondientes, que constituyen la guía principal del probador. Se confeccionó cada caso de prueba con la identificación de los principales escenarios, en dependencia de cada acción del actor, con sus posibles entradas. A continuación se muestran los casos de pruebas correspondientes a cada caso de uso.

1- DCP basado en CU Gestionar metadatos.

Descripción general

El usuario decide insertar todos los metadatos, modificar y deshacer los cambios efectuados y consultar la ayuda.

Condiciones de ejecución

El autor debe encontrarse en la generación o edición de un OA.

Secciones (SC) a probar en el caso de uso

Tabla 3: DCP de la SC Editar metadatos.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Editar metadatos.	EC 1.1: "Editar"	<p>El sistema muestra una vista de árbol junto a la vista formulario correspondiente, con los campos establecidos por LOM y por el bibliotecario de contenidos. El sistema muestra además los siguientes elementos autocompletados por categorías:</p> <p>Categoría General:</p> <p><Estructura: Jerárquica>, <Identificador: IDE- MD5 (time ())>, <Entrada: número continuo>.</p> <p>Categoría ciclo de Vida:</p>

		<p><Tipo: Autor, Creador>,<Fecha: date>, <Estado: Borrador>, <Versión: 2.0>, <Entidad: email>.</p> <p>Categoría Meta- Metadatos: <Contribución (Tipo): rol del usuario autenticado>, <Esquema de metadatos: LOM V2.0>, <Fecha: date>, Identificador, Entrada.</p> <p>Categoría Técnica: <Formato: ZIP>, <Requisitos (Tipo): Sistema operativo>, <Nombre: Multi-SO>, Tamaño.</p> <p>Categoría Uso educativo: <Destinatario: Aprendiz></p>
--	--	--

Tabla 4: DCP de la SC Insertar metadatos.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC1: Guardar los metadatos.	EC 1.2: "Guardar"	El sistema guarda los valores introducidos por el usuario y muestra los mismos en ambas vistas y finaliza el caso de uso.
SC2: Insertar todos los Metadatos.	EC2.1:"Insertar todos los metadatos"	El sistema muestra en ambas vistas todos los metadatos del esquema LOM.
SC3: Consultar sistema de ayuda.	EC 3.1: "Consultar"	El sistema muestra la explicación y un breve ejemplo del elemento de información.

Tabla 5: Descripción de la variable Categoría General.

No	Nombre de campo	Clasificación	Valor	Descripción

Capítulo 3: Implementación y Validación de la solución propuesta

			Nulo	
1	Estructura	Lista Desplegable	No	En este campo se permite seleccionar la estructura organizativa subyacente a este objeto educativo.
3	Entrada	Campo de texto	No	En este campo se permite introducir el valor del identificador dentro del esquema de identificación o catalogación que designa o identifica este objeto educativo. Una cadena específica de un espacio de nombre.

Tabla 6: Descripción de la variable Categoría Ciclo de Vida.

ID del escenario	Escenario	Tipo	Fecha	Estado	Versión	Respuesta del sistema	Resultado de la prueba
EC 1.1	Editar	V	V	V	V	Se espera que se edite el metadato exitosamente con la entrada de los datos correcto.	
		I	I	I	I	No se espera que se edite el Metadato exitosamente, pues los datos fueron entrados incorrectamente.	
		I	V	V	V	No se espera que se edite el metadato exitosamente, pues no se introdujo el tipo.	
		V	I	V	V	No se espera que se edite el Metadato exitosamente, pues no se introdujo la fecha correctamente.	
		V	V	I	V	No se espera que se edite el Metadato exitosamente, pues no se selecciona el estado.	

Capítulo 3: Implementación y Validación de la solución propuesta

		V	V	V	I	No se espera que se edite el Metadato exitosamente, pues no se seleccionó la versión.	
--	--	---	---	---	---	---	--

Tabla 7: Descripción de la variable Categoría Meta-Metadatos.

ID del escenario	Escenario	Contribución	Esquema de metadatos	Fecha	Identificador	Entrada	Respuesta del sistema	Resultado de la prueba
EC 1.1	Editar	V	V	V	V	V	Se espera que se edite el metadato exitosamente con la entrada de los datos correcto.	
		I	I	I	I	I	No se espera que se edite el metadato exitosamente, pues los datos fueron entrados incorrectamente.	
		I	V	V	V	V	No se espera que se edite el metadato exitosamente, pues no se selecciona la contribución correctamente.	
		V	I	V	V	V	No se espera que se edite el metadato exitosamente, pues no se introdujo el esquema de metadatos correctamente.	

Capítulo 3: Implementación y Validación de la solución propuesta

		V	V	I	V	V	No se espera que se edite el metadato exitosamente, pues no se introduce la fecha correctamente.	
		V	V	V	V	I	No se espera que se edite el metadato exitosamente, pues no se introduce la entrada correctamente.	

Tabla 8: Descripción de la variable Categoría Técnica.

ID del escenario	Escenario	Formato	Requisitos	Nombre	Tamaño	Respuesta del sistema	Resultado de la prueba
EC 1.1	Editar	V	V	V	V	Se espera que se edite el metadato exitosamente con la entrada correcta de los datos.	
		I	I	I	I	No se espera que se edite el metadato exitosamente, pues los datos fueron entrados incorrectamente.	
		I	V	V	V	No se espera que se edite el metadato exitosamente, pues no se selecciona la contribución	

Capítulo 3: Implementación y Validación de la solución propuesta

						correctamente.	
		V	I	V	V	No se espera que se edite el metadato exitosamente, pues no se introdujo el esquema de metadatos correctamente.	
		V	V	I	V	No se espera que se edite el metadato exitosamente, pues no se introduce la fecha correctamente.	
		V	V	V	I	No se espera que se edite el metadato exitosamente, pues no se introduce la entrada correctamente.	

Tabla 9: Descripción de la variable Categoría Uso educativo.

ID del escenario	Escenario	Destinatario	Respuesta del sistema	Resultado de la prueba
EC 1.1	Editar	V	Se espera que se edite el metadato exitosamente con la entrada del destinatario correctamente.	
		I	No se espera que se edite el metadato exitosamente, pues el destinatario no fue especificado.	

2- DCP basado en CU Exportar archivo XML.

Descripción general

El caso de uso se inicia cuando el actor decide exportar el archivo XML de la instancia de metadatos que se está creando hacia una dirección física en el equipo donde está trabajando y finaliza cuando realiza la acción.

Condiciones de ejecución

El actor debe encontrarse en la edición de los metadatos.

Secciones a probar en el caso de uso

Tabla 10: DCP de la SC Exportar metadatos.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Exportar Metadatos	EC 1.1: "Exportar Metadatos"	El sistema permite buscar la ubicación donde se guardará el archivo.

3- DCP basado en CU Importar archivo XML.

Descripción general

El caso de uso se inicia cuando el actor decide Importar desde una dirección física del equipo donde está trabajando, un archivo XML de los metadatos y finaliza cuando realiza la acción.

Condiciones de ejecución

El autor debe encontrarse en la edición de los metadatos.

Secciones a probar en el caso de uso

Tabla 11: DCP de la SC Importar metadatos.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC1: Importar Metadatos"	EC 1.1: "Importar Metadatos"	El sistema permite buscar la ubicación donde se guardará el archivo, importar el archivo y cancelar la acción de importar el archivo.

Tabla 12: Descripción de variable.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Ubicación	Campo de texto	Si	Este campo permite buscar la ubicación de donde se importará el archivo XML

Tabla 13: Matriz de datos de la SC1 Importar metadatos.

ID del escenario	Escenario	Ubicación	Respuesta del sistema	Resultado de la prueba
[EC 1,1]	Importar Metadatos	V	Se espera que se importe la instancia de metadatos.	
		I	No se Importa el archivo.	

4- DCP basado en CU Publicar metadatos.

Descripción general

El caso de uso se inicia cuando el actor decide publicar los metadatos para que sean visualizados y reutilizados por otros usuarios. El caso de uso termina cuando realiza estas acciones.

Condiciones de ejecución

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

El actor debe haber guardado al menos una vez los cambios efectuados en el OA.

Secciones a probar en el caso de uso

Tabla 14: DCP de la SC Publicar metadatos.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
		Para publicar metadatos el sistema verifica que los metadatos no estén ya publicados y los publica con el

SC1: Publicar Metadatos.	1: "Publicar"	título del recurso educativo, la fecha de publicación, descripción y muestra si la publicación es de metadatos de una agregación de contenido, de un ítem, organización o de un sco.
--------------------------	---------------	--

5- Eliminar metadatos publicados.

Descripción general

El caso de uso se inicia cuando el actor decide eliminar metadatos que han sido publicados.

Condiciones de ejecución

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

El actor debe haber guardado al menos una vez los cambios efectuados en el OA.

Secciones a probar en el caso de uso

Tabla 15: DCP de la SC Eliminar metadatos publicados.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC1: Eliminar Metadatos Publicados.	1: "Eliminar Metadatos"	El actor accede a la opción Eliminar metadatos. El sistema ofrece un listado con los metadatos publicados. El actor busca el metadato publicado que desea eliminar. El sistema muestra una ventana para confirmar si desea eliminar el metadato seleccionado. Si el actor elige que No, el sistema no elimina el metadato seleccionado. Si el actor elige la opción Si, el sistema elimina el metadato seleccionado.

6- Reutilizar metadatos.

Descripción general

El caso de uso se inicia cuando el actor decide visualizar o reutilizar metadatos obtenidos en el resultado de una búsqueda de los metadatos publicados por otros usuarios y termina cuando realiza estas acciones. Este caso de uso permite al actor utilizar metadatos creados por otros usuarios para describir sus contenidos, el sistema posibilita editar estos metadatos o reutilizar exactamente los mismos.

Condiciones de ejecución

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

El actor debe haber obtenido el resultado de una búsqueda de metadatos.

Secciones a probar en el caso de uso

Tabla 16: DCP de la SC Reutilizar metadatos.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Visualizar metadatos.	EC 1.1: "Visualizar metadatos"	El sistema muestra los elementos de información con su valor correspondiente en forma jerárquica. Ejemplo: General Título: Inteligencia artificial. Palabras claves: IA. Ciclo de vida Contribución: Autor.
SC 2: Reutilizar metadatos.	EC 2.1: "Reutilizar metadatos"	Luego el sistema ubica al actor en la vista de edición de los metadatos y carga el XML de los metadatos publicados en la instancia que el actor esté creando en ese momento y finaliza el caso de uso.

7- 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

Tabla 17: DCP de la SC Buscar metadatos.

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. 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.

Tabla 18: Descripción de la variable.

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.
2	Elemento de LOM.		Si	En este campo se selecciona aquel elemento perteneciente a

Capítulo 3: Implementación y Validación de la solución propuesta

				LOM por el cual se quiere hacer la búsqueda.
3	Valor.		Si	En este campo se introduce el valor del elemento asociado por el cual se desea realizar la búsqueda.

Tabla 19: Matriz de datos de la SC 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 metadatos.		Para buscar metadatos se: -Accede al menú lateral izquierdo a la opción
	V	I	V	El sistema busca los metadatos.		

Capítulo 3: Implementación y Validación de la solución propuesta

	V	V	I	El sistema busca los metadatos.		Metadato. -Luego se da clic en la opción Buscar Metadatos.
	V	V	V	El sistema busca los metadatos.		-Después sale la ventana con nombre Edición de Metadatos
	I	I	I	El sistema no hace la búsqueda de los metadatos porque falta el tipo de metadatos.		mostrando todos los datos referentes para su correcto establecimiento.

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.

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

No Conformidades Significativas: Son aquellas que afectan la calidad del producto o servicio de manera visible, impidiendo o no el cumplimiento de algún requisito.

No Conformidades No Significativas: Son aquellas que resultan menos visibles, que no atentan el cumplimiento de algún requisito.

Recomendaciones: Son aquellas que quedan en función de la apreciación del probador para oportunidades de mejoras del producto o servicio.

Resultados de las pruebas de Caja negra

Los resultados obtenidos durante las iteraciones son los descritos en la Tabla 20. 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.

Tabla 20: Resultado de las pruebas de caja negra.

Iteración	No conformidades	Tipo	Cerradas	No proceden
1ra	15	Alta	6	0
2da	9	Media	8	0
3ra	1	Baja	1	0

Como se puede apreciar a partir de las pruebas realizadas a todos los casos de uso se obtuvieron 15 no conformidades en la primera iteración, de las cuales 5 resultaron significativas, 5 no significativas y el resto recomendaciones. Para la segunda iteración se disminuyó a 6 no significativas y 3 recomendaciones. Y en la tercera y última iteración 1 recomendación.

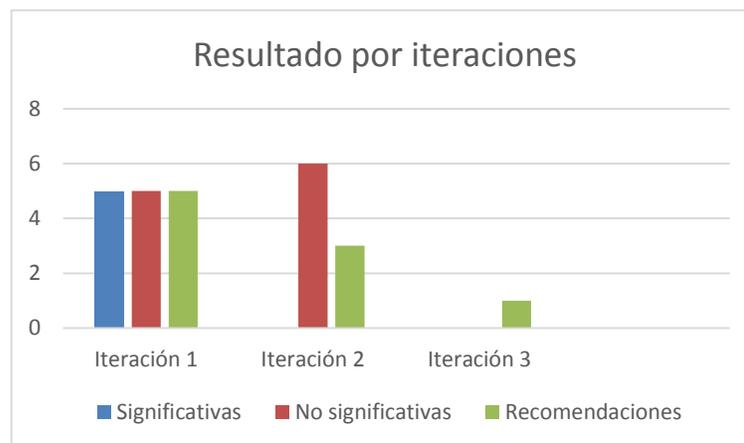


Ilustración 14: Resultado de las pruebas por iteraciones.

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 y la posterior

edición de los metadatos. Con el objetivo de erradicarlas se utilizaron expresiones regulares y se redefinieron determinadas funcionalidades, hasta erradicar todas las no conformidades.

3.2.3 Técnica de ladov

La técnica de ladov constituye una vía indirecta para el estudio de la satisfacción, ya que los criterios que se utilizan se fundamentan en las relaciones que se establecen entre tres preguntas cerradas que se intercalan dentro de un cuestionario y cuya relación el sujeto desconoce. Estas tres preguntas se relacionan a través de lo que se denomina el "Cuadro Lógico de ladov" (35).

Tabla 21: Cuadro Lógico de ladov.

¿Le gusta la solución propuesta para editar metadatos de los recurso educativos en el marco de trabajo Xalix?	1- ¿Considera que se garantiza un componente integrable en Xalix con la utilización directa del módulo de edición de metadatos en CRODA V2.0?								
	No			No sé			Sí		
	2- ¿Considera que para lograr la edición de metadatos en el marco de trabajo Xalix sería factible la utilización del editor de metadatos de la presente investigación basado en el esquema LOM?								
	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No
Me gusta mucho.	1	2	6	2	2	6	6	6	6
Me gusta más de lo que me disgusta.	2	2	3	2	3	3	6	3	6
Me es indiferente.	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta.	6	3	6	3	4	4	3	4	4
No me gusta en lo absoluto.	6	6	6	6	4	4	6	4	5

Capítulo 3: Implementación y Validación de la solución propuesta

No sé.	2	3	6	3	3	3	6	3	4
--------	---	---	---	---	---	---	---	---	---

El número resultante de la interrelación de las tres preguntas indica la posición de cada encuestado en la escala de satisfacción siguiente (35):

1. Clara satisfacción.
2. Más satisfecho que insatisfecho.
3. No definida.
4. Más insatisfecho que satisfecho.
5. Clara insatisfacción.
6. Contradictoria.

Por ejemplo: Si un encuestado responde a la pregunta uno "No", se va a la zona izquierda del cuadro, debajo de la pregunta uno, donde aparece No. Si a la pregunta dos responde "No sé" se busca el "No sé" que aparece debajo del No anterior. Si a la pregunta tres responde "Me disgusta más de lo que me gusta" entonces se busca en las filas, a la izquierda, la casilla donde aparece esa respuesta y se localiza el punto donde se interceptan la fila "Me disgusta más de lo que me gusta" con la columna "No sé". El resultado de dicho encuestado es "3", que equivale a "satisfacción no definida". Así se procede con cada usuario de la muestra, en dependencia de sus respuestas. De esta forma se van clasificando en las 6 categorías antes mencionadas.

Para obtener el índice de satisfacción grupal (ISG) se trabaja con los diferentes niveles de satisfacción que se expresan en una escala numérica que oscila entre +1 y - 1 de la siguiente forma (35):

Tabla 22: Niveles de satisfacción.

+1	Máximo de satisfacción
0.5	Más satisfecho que insatisfecho
0	No definido y contradictorio
-0.5	Más insatisfecho que satisfecho
-1	Máxima insatisfacción

La satisfacción grupal se calcula por la siguiente fórmula:

Capítulo 3: Implementación y Validación de la solución propuesta

$$ISG = \frac{A (+1) + B (+0,5) + C (0) + D (-0,5) + E (-1)}{N}$$

Ilustración 15: Fórmula de ISG.

En esta fórmula A, B, C, D, E, representan el número de encuestados con índice individual (1; 2; 3 ó 6; 4; 5) y N representa el número total de la muestra.

El índice grupal arroja valores entre + 1 y - 1. Los valores que se encuentran comprendidos entre - 1 y - 0,5 indican insatisfacción; los comprendidos entre - 0,49 y + 0,49 evidencian contradicción y los que caen entre 0,5 y 1 indican que existe satisfacción.



Ilustración 16: Nivel de satisfacción.

Para aplicar la técnica se escogió como población el centro FORTES debido a su experiencia tanto en el desarrollo como en el uso de la herramienta de autor CRODA, este cuenta con un total de 52 profesores, de ellos se seleccionó una muestra de 20. Después del análisis de los cuestionarios aplicados se obtuvo el siguiente resultado:

Tabla 23: Resultado de las escala de satisfacción.

Total de usuarios de la muestra (N).	20
Clara satisfacción.	13
Más satisfecho que insatisfecho.	6
No definida.	0
Más insatisfecho que satisfecho.	0
Clara insatisfacción.	0
Contradictoria.	1

A partir de estos resultados para calcular el ISG, las variables de la fórmula tomarían los siguientes valores:

Tabla 24: Variables de la fórmula del ISG.

A	13
B	6

Capítulo 3: Implementación y Validación de la solución propuesta

C	1
D	0
E	0

Calculando el ISG quedaría de la forma siguiente:

$$\text{ISG} = \frac{13(+1) + 6(+0.5) + 1(0) + 0(-0.5) + 0(-1)}{20}$$

$$20$$

$$\text{ISG} = 16 / 20$$

$$\text{ISG} = 0.8$$

El ISG resultante está dentro del intervalo de 0,5 a 1 por tanto el nivel de satisfacción de la muestra es de "Satisfecho".

Conclusiones generales

Al término de la presente investigación se ha dado cumplimiento a los objetivos planteados en la implementación del componente Editor de Metadatos basado en el esquema LOM para Xalix y de la cual se concluye que:

- El estudio realizado a la herramienta CRODA 2.0 y a la especificación LOM, permitió definir las características y tecnologías necesarias para un correcto desarrollo del Editor de metadatos basado en el esquema LOM para Xalix.
- Se obtuvo un editor de metadatos integrable al marco de trabajo Xalix, que facilitará la descripción de los recursos educativos, validada por el esquema LOM.
- Se logró un nivel de satisfacción grupal de 0.8 según la técnica de Iadov por parte de especialistas vinculados al trabajo con CRODA y el desarrollo de Xalix.
- Los métodos de validación definidos permitieron identificar y erradicar posibles errores del sistema, resaltando las pruebas de caja negra con particiones equivalentes.

Recomendaciones

Para futuras versiones del componente se recomienda:

Adaptar el módulo de manera tal, que acepte otros esquemas de catalogación además de LOM.

Bibliografía

Bibliografía

1. **Revista e-FORMADORES. De los Ángeles Serrano Isla, María.** s.l. : Instituto Latinoamericano de la Comunicación .
2. **Griffiths, David, y otros.** *La aportación de IMS Learning Design a la creación de recursos pedagógicos reutilizables.* s.l. : Grupo de Tecnologías Interactivas -Universitat Pompeu Fabra.
3. **López, Guzmán, y otros.** *Estándares y Especificaciones para los Entornos e-learning: Convergencia en Contenidos y Sistemas.* México : s.n., 2006.
4. **López, Guzmán y Clara.** *Los repositorios de Objetos de Aprendizaje como soporte a un entorno e-learning.* México : s.n., 2005.
5. **EcuRed.** Conocimiento con todos y para todos EcuRed. [En línea] [Citado el: 20 de 1 de 2014.] http://www.ecured.cu/index.php/Aprendizaje_electr%C3%B3nico.
6. *TIC en el proceso de enseñanza-aprendizaje. Serie Científica de la Universidad de las Ciencias Informáticas.* 10, s.l. : Entorno colaborativo para la elaboración de objetos de aprendizaje en CRODA., 2011, Vol. 4. ISSN | RNPS.
7. **Chan, Maria Elena.** *OBJETOS DE APRENDIZAJE: una herramienta para la innovación educativa.* 2001.
8. *CNICE Revista de Tecnologías de la Información y Comunicación Educativas . Segura Escobar, Mariano.* s.l. : Red Digital.
9. **Hernández, Eduardo.** Unidades de aprendizaje. [En línea] [Citado el: 6 de 4 de 2014.] http://campus.usal.es/~teoriaeducacion/rev_numero_06_2/n6_.
10. **Duval, Erick, y otros.** *Modelo de Agregación de Contenido.* Alexandria : s.n., 2006.
11. *Sharable Content Objetc. Panar, Angelo.* Alexandria : s.n., 2009, Vol. 1.
12. **Berlanga, Adriana, y otros.** *Consideraciones para Reforzar el Valor de los Objetos de aprendizaje.* España : Departamento de Informática y Automática, Universidad de Salamanca.
13. **Figuerola, Roberth G., Solís, Camilo J. y Cabrera, Armando A.** *METODOLOGIAS TRADICIONALES VS.METODOLOGIAS AGILES.* Loja : Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación.
14. **Zaragoza, Mtra. María de Lourdes Santiago.** *Desarrollando aplicaciones informáticas con el Proceso de Desarrollo Unificado (RUP).* s.l. : Programa Educativo de Tecnologías de la Información y Comunicación.
15. **Company Headquarters.** Visual Paradigm. [En línea] 2009. [Citado el: 18 de 3 de 2014.] <http://www.visual-paradigm.com>.
16. **Broche, Orlando Felipe Salvador.** *Indicaciones para el trabajo en el marco de trabajo Xalix .* 2013.

17. **PHP.net.** PHP. [En línea] 2001. [Citado el: 5 de 3 de 2014.] <http://www.php.net/manual/es/intro-what-is.php>.
18. **Eguíluz Pérez, Javier.** *Introducción a JavaScript*. Ciudad de la Habana : s.n., 2009.
19. **Alvarez, Miguel Angel.** desarrolloweb.com. [En línea] 11 de 1 de 2011. [Citado el: 7 de 4 de 2014.] <http://www.desarrolloweb.com/html>.
20. **Tangient LLC.** SistemasdeInformacion2. [En línea] 2014. [Citado el: 2 de 3 de 2014.] <http://sistemasdeinformacion2.wikispaces.com/LENGUAJE+UNIFICADO+DE+MODELADO%C2%A0%C2%A0%28UML%29>.
21. **Netbeans.** Netbeans. [En línea] [Citado el: 6 de 4 de 2014.] www.netbeans.org/community/releases/68/index_es.html.
22. **Pachucho Hernández, Betty Marlene.** Scribd. [En línea] 5 de 2009. [Citado el: 3 de 2 de 2014.] <http://es.scribd.com/doc/55136485/47/Ventajas-de-utilizar-un>.
23. **Javier.** symfony.es. [En línea] [Citado el: 3 de 4 de 2014.] <http://www.symfony.es/>.
24. **WordPress.** jQuery. [En línea] WordPress, 2014. [Citado el: 10 de 1 de 2014.] <http://docs.jquery.com/>.
25. **Ortín, María José, y otros.** *El Modelo del Negocio como basedel Modelo de Requisitos*. España : Departamento de Informática y Sistemas.
26. **Larman, Craig.** *MODELO DEL DOMINIO*. s.l. : Prentice Hall, 2003.
27. **Sommerville, Ian.** *Ingeniería del Software*. Madrid : s.n., 2005. ISBN:84-7829-074-5.
28. **Pressman, Roger.** *Ingeniería del Software. Un enfoque práctico*. Salamanca : s.n., 2002. ISBN: 970-10-5473-3.
29. **JACOBSON, IVAR, BOOCH, GRADY y RUMBAUCH, JAMES.** *El PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE*. Madrid : s.n., 2000. ISBN 84-7829-036-2.
30. **Microsoft.** Microsoft Developer Network. [En línea] 2014. [Citado el: 6 de 5 de 2014.] <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.
31. **Hernández Tejada, David.** TELEPROGRAMADORES. GUIA DE PATRONES DE DISEÑO. [En línea] 2002. <http://www.teleprogramadores.com>.
32. **LARMAN, CRAIG.** *UML Y PATRONES. INTRODUCCION AL ANALISIS Y DISEÑO ORIENTADO A OBJETOS*. México : Cámara Nacional de la Industria Editorial Mexicana , 1999. ISBN 970-1 7-0261-1.
33. **Tremps Gómez, Javier.** Patrón de arquitectura Modelo Vista Controlador (MVC). [En línea] [Citado el: 20 de 4 de 2014.] <http://www.lab.inf.uc3m.es/~a0080802/RAI/mvc.html>.
34. **DMCA.** slideshare. [En línea] 2014. <http://www.slideshare.net/pablis001/estrategias-de-aplicaciones-para-las-pruebas-de-integracin>.

35. **López Rodríguez, Alejandro y González Maura, Viviana.** La técnica de ladov. Una aplicación para el estudio de la satisfacción de los alumnos para las clases de educación física. *Revista Digital -Buenos Aires*. [En línea] 4 de 2002. <http://www.efdeportes.com/>.

Glosario de términos

1. **Accesibilidad:** Capacidad que hace identificable y ubicable el 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.
2. **Acoplamiento:** Medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas.
3. **Ajax:** Técnica de desarrollo web para crear aplicaciones interactivas.
4. **Blogs:** Sitio web en el que uno o varios autores publican cronológicamente textos o artículos, apareciendo primero el más reciente, donde el autor conserva siempre la libertad de dejar publicado lo que crea pertinente y donde suele ser habitual que los propios lectores participen activamente a través de sus comentarios. Un blog puede servir para publicar ideas propias y opiniones sobre diversos temas.
5. **Catalogación:** Registro ordenado de libros, documentos o material semejante de acuerdo con algunas normas.
6. **Cohesión:** Medida de cuan relacionadas y enfocadas están las responsabilidades de una clase.
7. **Colaborativa:** Procesos intencionales de un grupo para alcanzar objetivos específicos, más herramientas para dar soporte y facilitar el trabajo.
8. **Estándar:** Tecnología, formato o método ratificado por algún organismo oficial de estandarización.
9. **Especificación:** Propuesta de una tecnología, formato o método, que busca obtener la aprobación de un organismo oficial de estandarización para convertirse en un estándar.
10. **Framework (marco de trabajo):** Define un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar. En el desarrollo del software es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software.
11. **Hipertexto:** Modelo teórico de organizar la información de manera no secuencial. Herramienta de software o programa informático que permite leer y escribir documentos hipertextuales. Documento digital que aprovecha la ventaja de la computabilidad para permitir un acceso no secuencial a la información. En este sentido, se entiende un hipertexto como contenido de la información, incluyendo los fragmentos de información y las conexiones entre esos fragmentos, indiferentemente de cuál sea el sistema utilizado para leer o escribir tal documento.

12. **HTTP:** Hypertext Transfer Protocol (en español protocolo de transferencia de hipertexto).
13. **IEEE/LOM (Institute of Electrical and Electronics Engineers/Learning Technology Standards Committee):** Asociación internacional, cuya misión es promover los procesos ingenieriles para la creación, desarrollo, integración, compartición y aplicaciones del conocimiento sobre tecnologías eléctricas y de información.
14. **IMAP:** Internet Message Access Protocol (IMAP, Protocolo de acceso a mensajes de internet), es un protocolo de aplicación que permite el acceso a mensajes almacenados en un servidor de Internet. Mediante IMAP se puede tener acceso al correo electrónico desde cualquier equipo que tenga una conexión a Internet.
15. **Integración:** Conjunto de elementos relacionados o que interactúan, que permiten implantar y alcanzar la política y los objetivos de una organización, en lo que se refiere a aspectos diversos como pueden ser los de seguridad. Es el proceso de combinar y probar múltiples componentes juntos. El objetivo es tomar los componentes probados en unidad y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño. Se llama integración incremental cuando el programa se construye y se prueba en pequeños segmentos en los que los errores son más fáciles de aislar y corregir, es más probable que se puedan probar completamente las interfaces y se pueda aplicar un enfoque de prueba sistemática.
16. **Interoperabilidad:** Capacidad de las plataformas para comunicarse e integrarse de una forma efectiva sin importar que puedan ser diferentes.
17. **Java:** Lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en clases, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo.
18. **LDAP:** LDAP son las siglas de Lightweight Directory Access Protocol (en español Protocolo Ligero de Acceso a Directorios) que hacen referencia a un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.
19. **NNTP:** Network News Transport Protocol (NNTP), protocolo para la transferencia de noticias en red.
20. **POP3:** Post Office Protocol (POP3, Protocolo de Oficina de Correo o "Protocolo de Oficina Postal"). Protocolo empleado para obtener correos desde un servidor.
21. **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 volver a utilizarlos.

- 22. SNMP:** Protocolo Simple de Administración de Red o SNMP (del inglés Simple Network Management Protocol), es un protocolo de la capa de aplicación que facilita el intercambio de información de administración entre dispositivos de red. Permite a los administradores supervisar el funcionamiento de la red, buscar, resolver sus problemas y planear su crecimiento.
- 23. Usabilidad:** Se dirige a conseguir el objetivo de satisfacer más a los usuarios, con un sitio web más eficaz y eficiente.
- 24. Wikis:** Sitio web cuyas páginas pueden ser editadas por múltiples voluntarios a través del navegador web. Los usuarios pueden crear, modificar o eliminar un mismo texto que comparten. Los textos o «páginas wiki» tienen títulos únicos. Si se escribe el título de una «página wiki» en algún sitio del wiki entre dobles corchetes ([[...]]), esta palabra se convierte en un «enlace web» a la página correspondiente. Permite crear y mejorar las páginas de forma inmediata, dando una gran libertad al usuario y por medio de una interfaz muy simple.
- 25. Workflow (Flujo de trabajo):** Estudio de los aspectos operacionales de una actividad de trabajo, cómo se estructuran las tareas, cómo se realizan, cuál es su orden correlativo, cómo se sincronizan, cómo fluye la información que soporta las tareas y cómo se le da seguimiento al cumplimiento de las tareas.

Anexos

Anexo 1. Descripción de los casos de uso identificados en el sistema

Tabla 25: Descripción del caso de uso Exportar archivo XML.

CU-2	Exportar archivo XML de los metadatos.	
Actor	Bibliotecario de contenidos.	
Resumen	El caso de uso se inicia cuando el actor decide exportar el archivo XML de la instancia de metadatos que está creando hacia una dirección física en el equipo donde está trabajando y finaliza cuando realiza la acción.	
Referencias	R8	
Prioridad	Opcional	
Precondiciones	El bibliotecario de contenidos debe encontrarse en la edición de los metadatos.	
Poscondiciones	El archivo XML se exportó correctamente.	
CU asociados	-	
Flujo Normal		
Acción del actor	Respuesta del sistema	
1. El actor accede a la opción "Exportar metadatos".	2. El sistema ofrece mediante el navegador web la posibilidad de buscar la ubicación donde guardará el archivo.	
3. El actor busca la ubicación y guarda el archivo.	4. El sistema guarda en la ubicación seleccionada el archivo XML de los metadatos y finaliza el caso de uso.	
Prototipo de interfaz de usuario		

Guardar	Cancelar	General	Ciclo de Vida	Meta metadatos	Técnica	Uso educativo	Derechos	Relación	Anotación	Clasificación	
		<ul style="list-style-type: none"> Metadatos LOM <ul style="list-style-type: none"> General <ul style="list-style-type: none"> Identificador Catálogo Entrada Título Idioma Descripción Palabras claves Ámbito Estructura Nivel de agregación Ciclo de Vida Meta-metadatos Técnica Uso educativo Derechos Relación Anotación Clasificación 	<p>* Identificador <input type="text"/> ?</p> <p>Catálogo <input type="text"/> ?</p> <p>Entrada <input type="text"/> ?</p> <p>Título <input type="text"/> ?</p> <p>Idioma <input type="text"/> ▼ ?</p> <p>Descripción <input type="text"/> ?</p> <p>Palabras claves <input type="text"/> ?</p> <p>Ámbito <input type="text"/> ▼ ?</p> <p>Estructura <input type="text"/> ▼ ?</p> <p>Nivel de Agregación <input type="text"/> ▼ ?</p>								

Tabla 26: Descripción del caso de uso Importar archivo XML.

CU-3	Importar archivo XML de los metadatos.
Actor	Bibliotecario de contenidos.
Resumen	El caso de uso se inicia cuando el actor decide Importar desde una dirección física del equipo donde está trabajando un archivo XML de los metadatos y finaliza cuando realiza la acción.
Referencias	R9.
Prioridad	Opcional.
Precondiciones	El bibliotecario de contenidos debe encontrarse en la edición de los metadatos.
Poscondiciones	Es importado un archivo XML a la instancia de metadatos.
CU asociados	-
Flujo Normal	
Acción del actor	Respuesta del sistema
1. El actor accede a la opción "Importar metadatos".	2. El sistema ofrece la posibilidad de buscar la ubicación del archivo, importarlo o cancelar la acción de importar el

	archivo.
3. El actor busca la ubicación del archivo y accede a Importar el archivo seleccionado.	4. El sistema carga el archivo XML en la instancia de metadatos que se está creando y finaliza el caso de uso.
Flujo Alterno “Importar sin localizar el archivo”	
Acción del actor	Respuesta del sistema
3. El actor no busca la ubicación donde se encuentra el archivo y accede a la opción de “Importar”.	4. El sistema muestra el mensaje de aviso: “Debe buscar el archivo primeramente”. Vuelve al paso 3 del Flujo Normal.
Flujo Alterno “Cancelar”	
Acción del actor	Respuesta del sistema
3. El actor accede a la opción Cancelar.	4. El sistema cierra la ventana de importar el archivo y finaliza el caso de uso.
Flujo Alterno “Extensión del archivo no correcta”	
Acción del actor	Respuesta del sistema
3. El actor accede a la opción “Examinar” y busca un archivo que no es un XML de una instancia de metadatos LOM y accede a la opción de “Exportar”.	4. El sistema muestra el siguiente mensaje: “El archivo que usted seleccionó no es un archivo XML de metadatos del esquema LOM”. Regresa al paso 1 del flujo normal de eventos.

Prototipo de interfaz de usuario

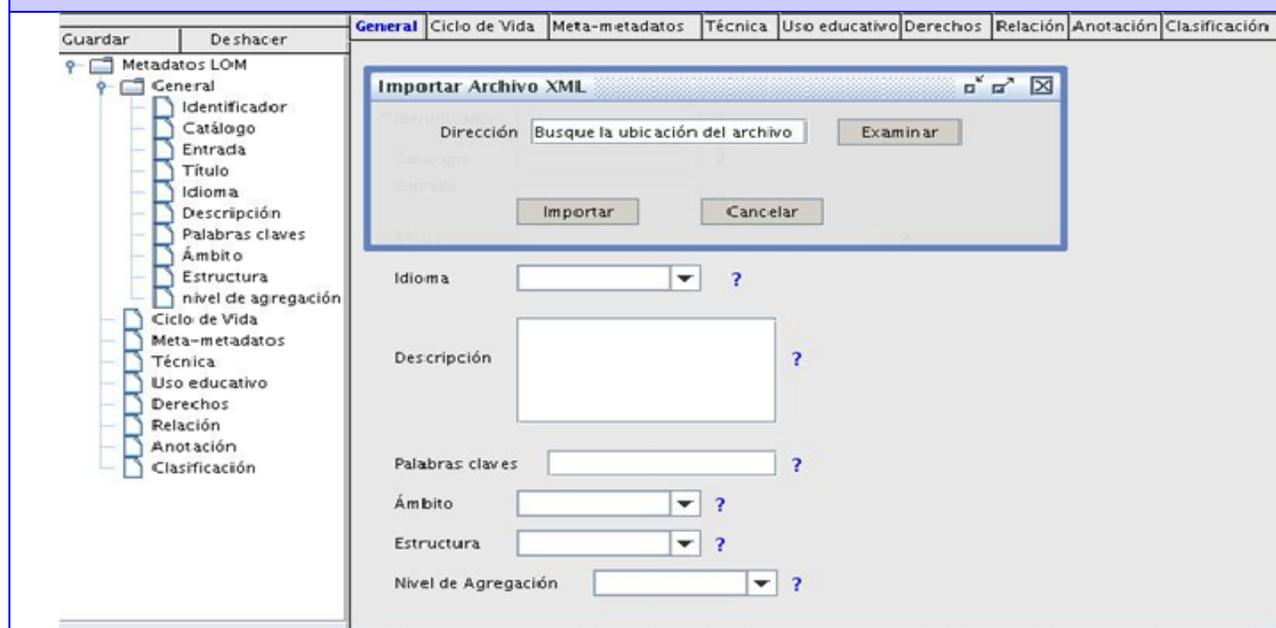


Tabla 27: Descripción del caso de uso Publicar metadatos.

CU-4	Publicar metadatos.	
Actor	Bibliotecario de contenidos.	
Resumen	El caso de uso se inicia cuando el actor decide publicar los metadatos para que sean visualizados y reutilizados por otros usuarios. El caso de uso termina cuando realiza estas acciones.	
Referencias	R10.	
Prioridad	Crítico.	
Precondiciones	El bibliotecario de contenidos debe encontrarse autenticado en el sistema y en la edición de los metadatos. El actor debe haber guardado al menos una vez los cambios efectuados en el recurso educativo.	
Poscondiciones	Los metadatos fueron publicados correctamente por el actor.	
CU asociados	-	
Flujo Normal		
Acción del actor	Respuesta del sistema	
1. El actor accede a la opción “Publicar metadatos”.	2. El sistema verifica que los metadatos no estén ya publicados y los publica con el título del recurso educativo, la fecha de publicación, descripción y muestra si la publicación es de metadatos de una agregación de contenido, de un ítem u organización o de un SCO. 3. El sistema muestra el mensaje: “Sus metadatos han sido publicados para ser visualizados y reutilizados por otros usuarios.” Finaliza el caso de uso.	
Flujo Alternativo		
Acción del actor	Respuesta del sistema	
1. El actor accede a la opción Publicar metadatos.	2. El sistema verifica que los metadatos ya fueron publicados. Muestra al actor el siguiente mensaje: “Usted ya realizó una publicación de esta instancia de metadatos.” Finaliza el caso de uso.	
Prototipo de interfaz de usuario		

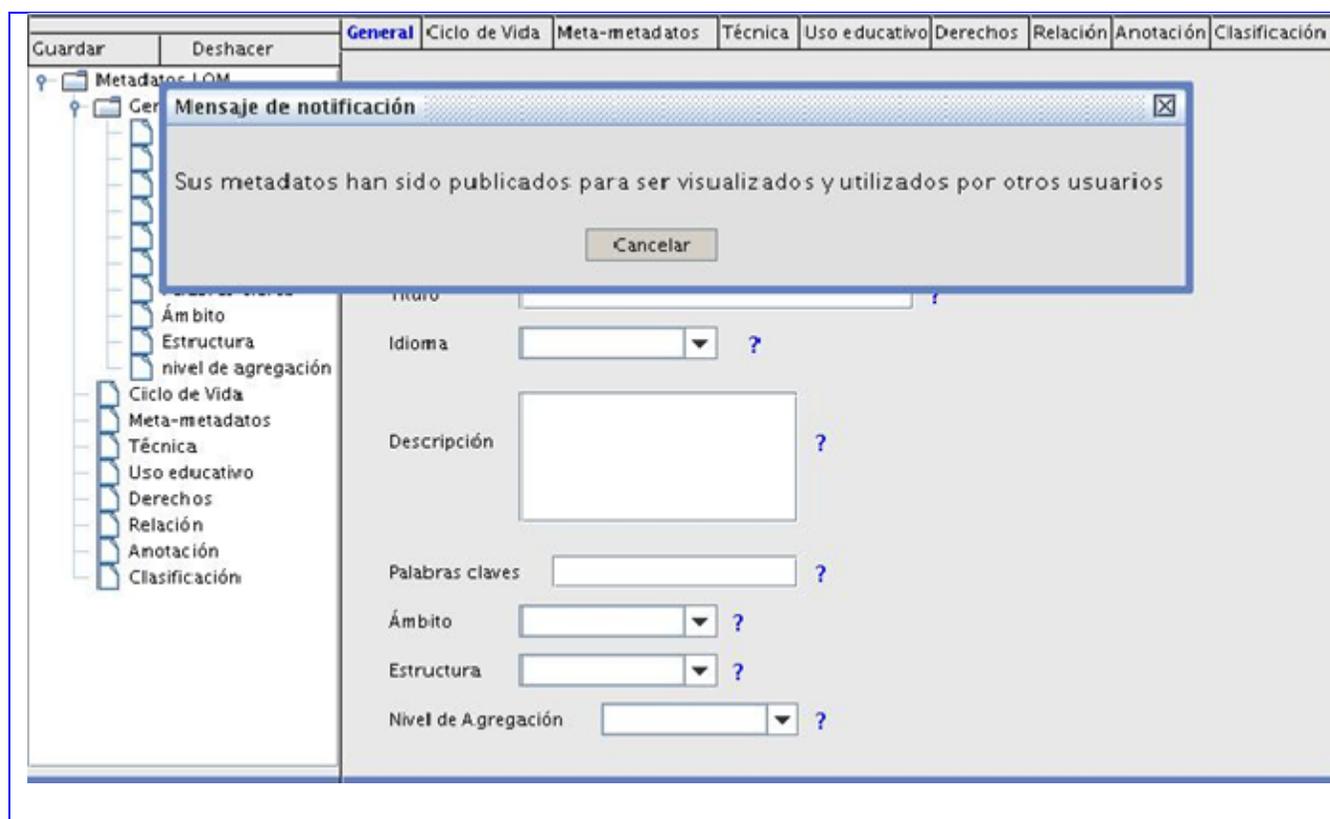


Tabla 28: Descripción del caso de uso Eliminar metadatos publicados.

CU-5	Eliminar metadatos publicados.	
Actor	Bibliotecario de contenidos.	
Resumen	El caso de uso se inicia cuando el actor decide eliminar metadatos que han sido publicados.	
Referencias	R15	
Prioridad	Crítico.	
Precondiciones	El archivo debe estar publicado.	
Poscondiciones	El archivo se eliminó correctamente.	
CU asociados	-	
Flujo Normal		
Acción del actor	Respuesta del sistema	
1. El actor accede a la opción "Eliminar metadatos".	2. El sistema ofrece un listado con los metadatos publicados.	
3. El actor busca el metadato publicado que desea eliminar.	4. El sistema muestra una ventana para confirmar si desea eliminar el metadato seleccionado.	

Flujo Alterno	
Acción del actor	Respuesta del sistema
1. El actor elige la opción NO.	2. El sistema no elimina el metadato seleccionado.
3. El actor elige la opción SI.	4. El sistema elimina el metadato seleccionado.
Prototipo de interfaz de usuario	

Tabla 29: Descripción del caso de uso Reutilizar metadatos.

CU-6	Reutilizar metadatos.
Actor	Bibliotecario de contenidos.
Resumen	El caso de uso se inicia cuando el actor decide visualizar o reutilizar metadatos obtenidos en el resultado de una búsqueda de los metadatos publicados por otros usuarios y termina cuando realiza estas acciones. Este caso de uso permite al actor utilizar metadatos creados por otros usuarios para describir sus contenidos, el sistema posibilita editar estos metadatos o reutilizar exactamente los mismos.
Referencias	R12, R13.
Prioridad	Critico.

Precondiciones	El actor debe encontrarse autenticado en el sistema y en la edición de los metadatos. El actor debe haber obtenido el resultado de una búsqueda de metadatos.	
Poscondiciones	Los metadatos publicados fueron visualizados y utilizados correctamente por el actor.	
CU asociados	-	
Flujo Normal		
Acción del actor	Respuesta del sistema	
1. El actor realiza una búsqueda de metadatos publicados.	2. El sistema en cada publicación de metadatos listada o buscada por el usuario muestra las opciones. Visualizar (Ver sección 1). Reutilizar (Ver sección 2).	
Sección 1 :“Visualizar metadatos”		
Acción del actor	Respuesta del sistema	
1. El actor accede a la opción “Visualizar metadatos” que se encuentra en las publicaciones listadas o buscadas.	2. El sistema muestra los elementos de información con su valor correspondiente en forma jerárquica. Ejemplo: General Título: Inteligencia artificial. Palabras claves: IA. Ciclo de vida Contribución: Autor. Finaliza el caso de uso.	
Sección 2 :“Reutilizar metadatos”		
Acción del actor	Respuesta del sistema	
1. El actor accede a la opción Reutilizar metadatos que se encuentra en las publicaciones listadas.	2. El sistema ubica al actor en la vista de edición de los metadatos y carga el XML de los metadatos publicados en la instancia que el actor esté creando en ese momento y finaliza el caso de uso.	
Prototipo de interfaz de usuario		

Título del diseño instruccional: *Título*

Fecha de publicación: *Fecha*

metadatos de: *Componente del diseño(rol, ambiente, actividad)*

Visualizar metadatos **Reutilizar metadatos**

Título del diseño instruccional: *Título1*

Fecha de publicación: *Fecha1*

metadatos de: *Componente del diseño(rol, ambiente, actividad)1*

Visualizar metadatos **Reutilizar metadatos**

Título del OA: *Título*

Fecha de publicación: *Fecha*

Descripción del OA: *Descripción*

Metadatos de: *{ Organización, item o agregación de contenido }*

Visualizar metadatos **Reutilizar metadatos**

Título del OA: *Título1*

Fecha de publicación: *Fecha1*

Descripción del OA: *Descripción1*

Metadatos de: *{ Organización, item o agregación de contenido }*

Visualizar metadatos **Reutilizar metadatos**

Tabla 30: Descripción del caso de uso Buscar metadatos.

CU-7	Buscar metadatos (Incluye del caso de uso Reutilizar metadatos).	
Actor	Bibliotecario de contenidos.	
Resumen	El caso de uso se inicia cuando el actor decide realizar una búsqueda por criterios de metadatos publicados por todos los usuarios o listar todas las publicaciones realizadas por estos. El caso de uso termina cuando realiza estas acciones.	
Referencias	R11, R14.	
Prioridad	Critico.	
Precondiciones	El actor debe encontrarse autenticado en el sistema y en la edición de los metadatos.	
Poscondiciones	Las publicaciones de metadatos fueron buscadas y listadas correctamente por el actor.	
CU asociados	CU Reutilizar metadatos.	
Flujo Normal		
Acción del actor	Respuesta del sistema	
1. El actor decide realizar una búsqueda de metadatos publicados y accede a la opción "Buscar metadatos".	2. El sistema brinda la posibilidad de escoger el tipo de metadatos que desea buscar mediante las opciones: -Metadatos de recursos educativos. Permite al actor escoger los elementos del esquema LOM por los cuales desea realizar la búsqueda e introducir los valores que desea encontrar en esos elementos. Muestra además, la opción Buscar para ejecutar la búsqueda por los criterios establecidos.	
3. El actor selecciona el tipo de metadatos que desea buscar.		
4. El actor selecciona los elementos por los cuales desea buscar, inserta los valores de los mismos y accede a la opción Buscar.	5. El sistema muestra una lista de publicaciones de metadatos que coinciden con los criterios de búsqueda introducidos por el actor y finaliza el caso de uso.	
Flujo Alternativo "No selecciona el tipo de metadatos a buscar."		
Acción del actor	Respuesta del sistema	
3. El actor no selecciona el tipo de metadatos que desea buscar.		
4. El actor selecciona los elementos por los	5. El sistema muestra un mensaje de aviso: "Debe	

cuales desea buscar, inserta los valores de los mismos y accede a la opción "Buscar".	seleccionar el tipo de metadatos que desea buscar". Vuelve al paso 3 del flujo normal.
---	---

Flujo Alterno "No inserta valores para los elementos de búsqueda"

Acción del actor	Respuesta del sistema
4. El actor selecciona los elementos por los cuales desea buscar, no inserta ningún valor para estos elementos y accede a la opción "Buscar".	5. El sistema muestra una lista de publicaciones de metadatos que coinciden con los criterios de búsqueda (elementos seleccionados) introducidos por el actor y finaliza el caso de uso.

Flujo Alterno "No inserta elementos ni valores para la búsqueda"

Acción del actor	Respuesta del sistema
4. El actor no selecciona ningún criterio de búsqueda y accede a la opción "Buscar".	5. El sistema lista todos los metadatos públicos del tipo seleccionado y finaliza el caso de uso.

Flujo Alterno "No se encuentran resultados para los criterios de búsqueda"

Acción del actor	Respuesta del sistema
4. El actor selecciona los elementos por los cuales desea buscar, inserta los valores de los mismos y accede a la opción "Buscar".	5. El sistema no encuentra publicaciones de metadatos que coinciden con los criterios de búsqueda introducidos por el actor, muestra el mensaje de notificación: "No se encontraron metadatos publicados con los criterios de búsqueda introducidos". Finaliza el caso de uso.

Prototipo de interfaz de usuario

The screenshot shows a search interface with the following elements:

- A dropdown menu labeled "Tipo de metadatos a buscar" with the value "Selecciona" and a downward arrow.
- Two columns of input fields:
 - "Elementos de LOM": Four dropdown menus, each with "Selecciona" and a downward arrow.
 - "Valor": Four empty text input boxes.
- A "Buscar" button centered at the bottom.

Anexo 2. Diagramas de colaboración de los casos de uso definidos

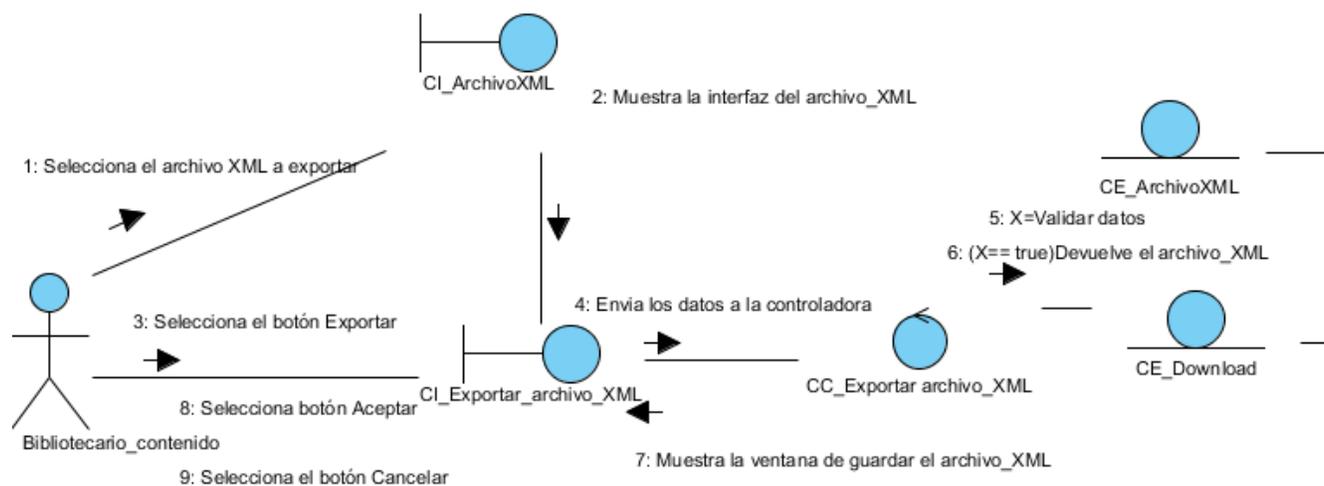


Ilustración 17: Diagrama de colaboración del escenario Exportar archivo XML.

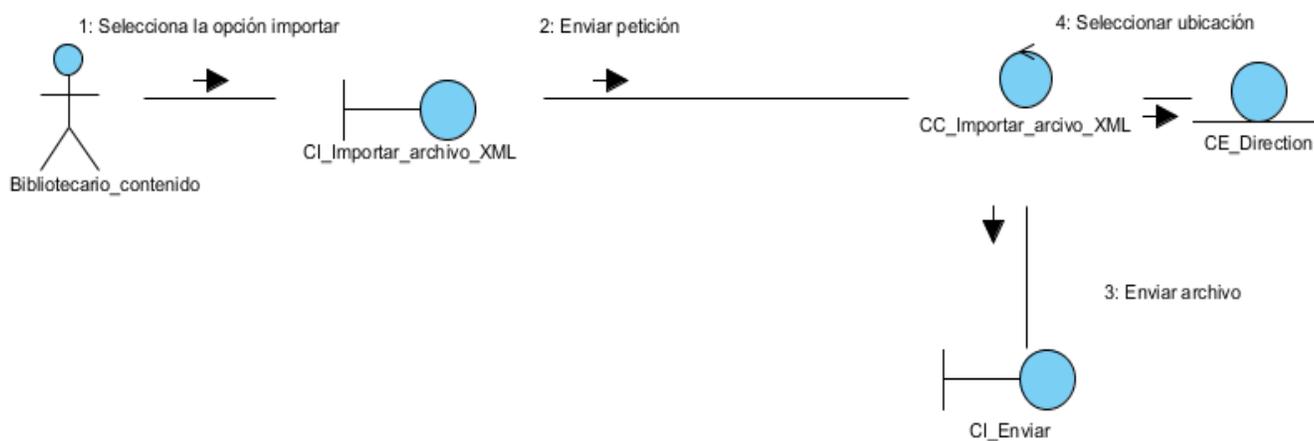


Ilustración 18: Diagrama de colaboración del escenario Importar archivo XML.

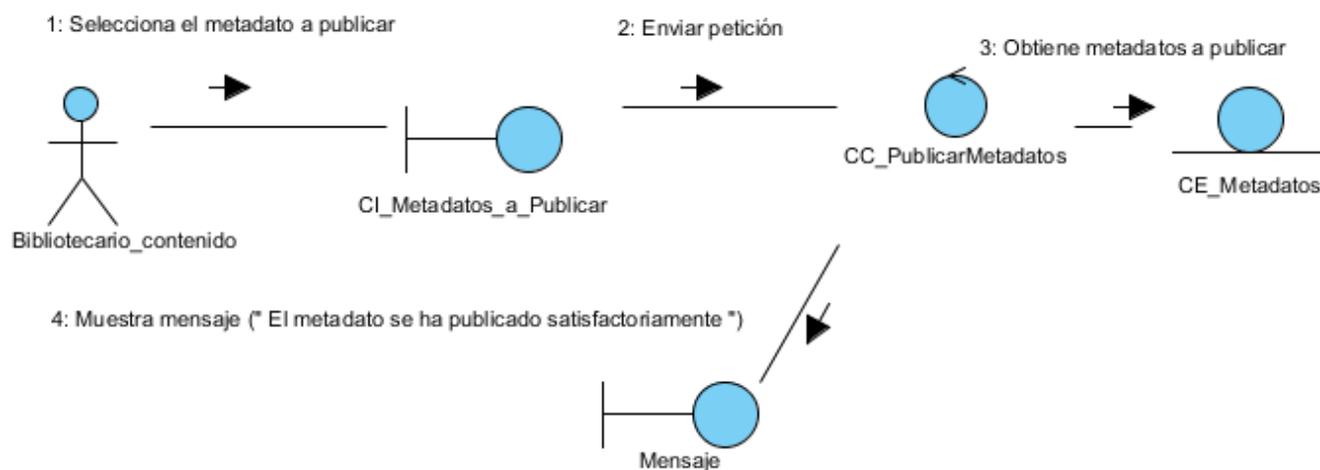


Ilustración 19: Diagrama de colaboración del escenario Publicar metadatos.

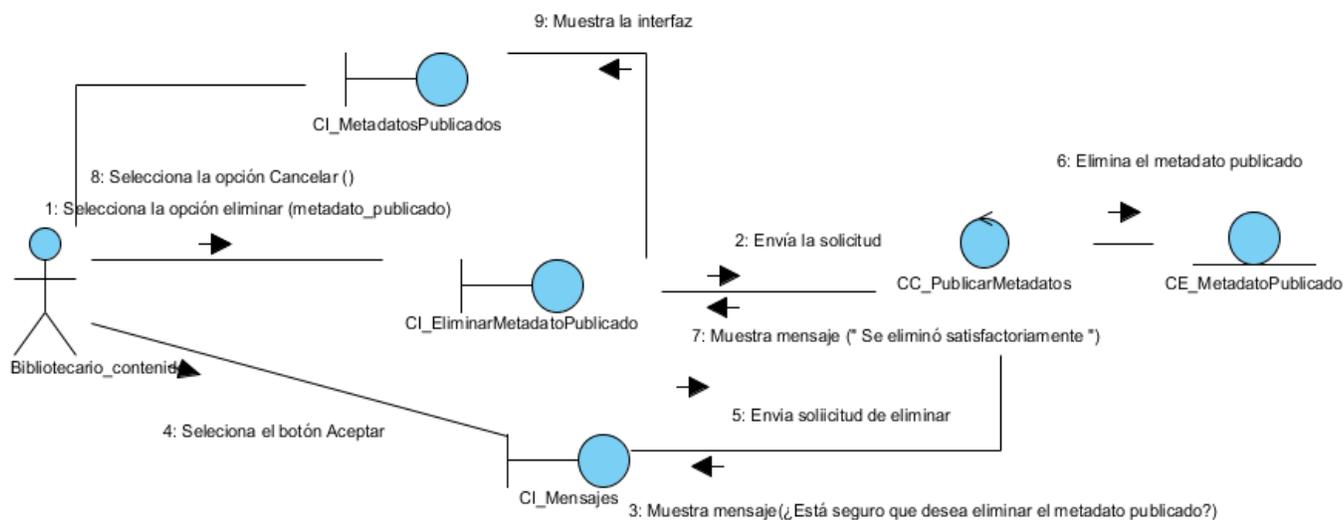


Ilustración 20: Diagrama de colaboración del escenario Eliminar metadatos publicados.

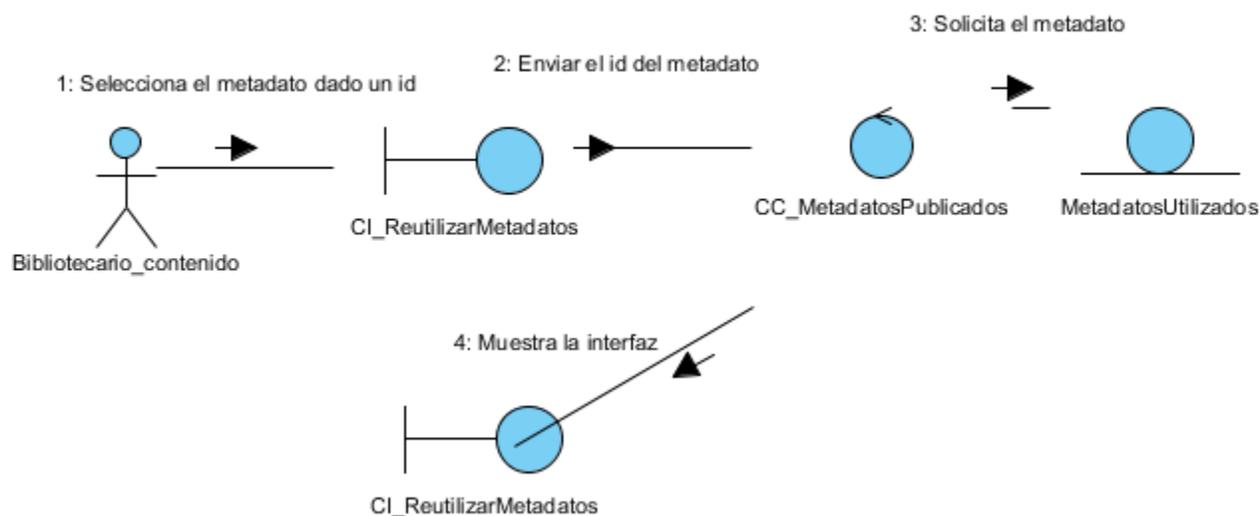


Ilustración 21: Diagrama de colaboración del escenario Reutilizar metadatos.

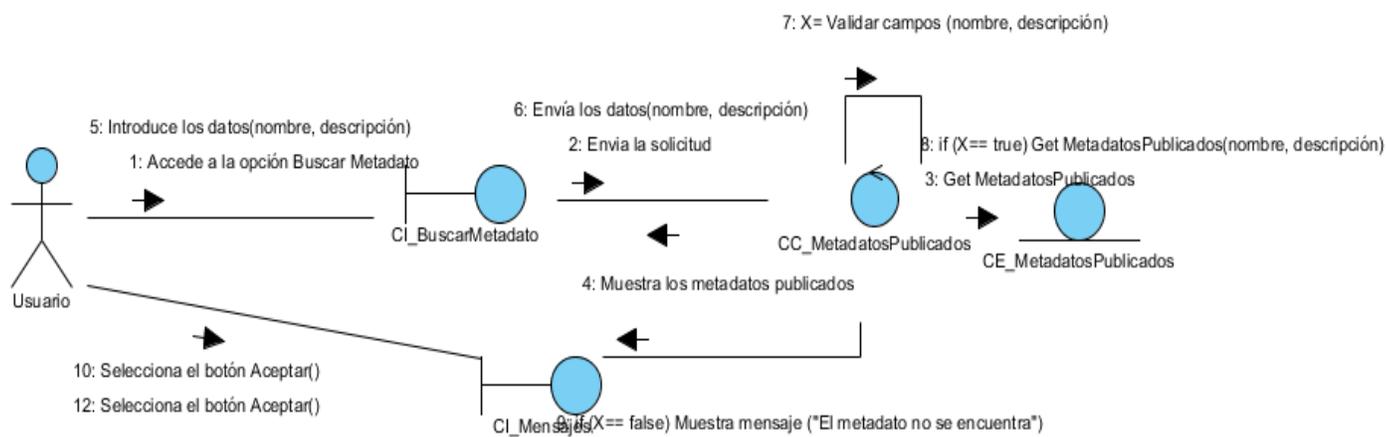


Ilustración 22: Diagrama de colaboración del escenario Buscar metadatos.