



Universidad de las Ciencias Informáticas

*Análisis y diseño del módulo de Revisión de la
calidad de Objetos de Aprendizaje para los
proyectos CRODA y RHODA.*

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

Autor:

Daymis Durán Peña

Tutores:

Ing. Arlan Galvez Alonso

Ing. Jose Antonio Soto Pérez

Lic. Jorge Alberto León Cumblera

La Habana, Junio de 2011



“...Sí un hombre es perseverante, aunque sea duro de entendimiento se hará inteligente y aunque sea débil se transformará en fuerte.”

Leonardo Da Vinci

DECLARACIÓN DE AUTORÍA

Declaro ser autora de la presente tesis y autorizo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los _____ días del mes de _____ del año 2011.

Daymis Durán Peña

Autora

Ing. Arlan Galvez Alonso

Tutor

Ing. Jose A. Soto Pérez

Tutor

Lic. Jorge A. León Cumbreira

Tutor

DEDICATORIA

A mi mamita Modesta Peña García y mi pa´ Antenor Durán Góngora por todo cariño que me han dado, así como el esfuerzo y sacrificio que han hecho para que yo pudiera realizar este sueño.

AGRADECIMIENTOS

A mi mamita, que nunca ha dejado de confiar en mis decisiones y me ha apoyado siempre.

A mi pa', que nunca ha dejado de aconsejarme para que sea mejor persona.

A Pipo mi hermanito lindo, a mi negrito Brian, en fin, a toda mi familia, la que todos quisieran tener, la que es muy grande y no puedo mencionarlos a todos, gracias por preocuparse por mi y quererme tanto.

A mis amigos y hermanos Ailen (La mimi bella), Yei (La mimi Aldany), Joseito (El primito), Yosvany (el ingeniero), Jose Luis, Yibi, Mayre, Adry y Silvita por regalarme su amistad.

A los integrantes de Bertha's Paladar poder disfrutar de momentos tan lindos con ellos.

A la Isabel Lombillo madre-profesora que siempre está aconsejando y ayudando para que se cumpliera mi sueño.

A mis tutores Jose, Jorge y Arlan que no importó la cantidad de trabajo que tenían por hacer siempre dejaron un tiempito para mí.

A mis compañeros de aula que aportaron siempre su granito de arena.

A los profes que durante estos cinco años de carrera, trataron de formarme como ingeniera.

A todas mis amistades que me llevo un poquito de cada uno, a todos los quiero.

A los que me pusieron piedras en el camino, me ayudaron a hacerme más fuerte.

A los que hicieron la pregunta ¿Cómo va la tesis?

A todos muchas gracias.

RESUMEN

Con la incorporación de las TIC a la educación, se evidenciaron innumerables avances, entre ellos el surgimiento del e-learning, en el cual fue apareciendo la necesidad de compartir recursos y reutilizarlos en cualquier área del conocimiento, surgiendo el término Objetos de Aprendizaje (OA). La Universidad de las Ciencias Informáticas (UCI) ha introducido los OA en función de mejorar la calidad de los recursos didácticos que se desarrollan en la misma. En el presente trabajo de diploma se tiene como objetivo principal realizar el análisis y diseño del módulo de Revisión de la calidad de OA para los proyectos CRODA y RHODA. Para ello se realiza un estudio de diferentes modelos de evaluación de la calidad de OA en algunas universidades del mundo, y se trae una propuesta para ser utilizada en el módulo que se desea implementar para un futuro.

Como resultado del presente trabajo, se generan algunos artefactos como son el diagrama de casos de uso, diagrama de clases del análisis, diagrama de clases del diseño, todos del flujo de trabajo Análisis y diseño que establece la metodología Proceso Unificado de Desarrollo (RUP).

Palabras claves: Objetos de Aprendizaje, CRODA, RHODA, calidad, evaluación.

TABLA DE CONTENIDOS

DEDICATORIA	I
AGRADECIMIENTOS.....	II
RESUMEN.....	1
INTRODUCCIÓN	1
CAPÍTULO#1: FUNDAMENTACIÓN TEÓRICA.....	6
1.1 Introducción	6
1.2 Objeto de Aprendizaje	6
1.2.1 Características de un OA.....	7
1.2.2 Principales componentes de un OA.....	8
1.3 Modelos de evaluación de la calidad de OA	9
1.3.1 Universidad Autónoma de Aguascalientes de México.....	10
1.3.2 Universidad de Salamanca de España	11
1.3.3 Universidad del BioBio de Chile en unión con la Universidad de Castilla-La Mancha de España	12
1.3.4 Instrumento de Revisión de Objetos de Aprendizaje	12
1.3.5 Fundamentación del modelo seleccionado.....	13
1.4 Estándares y especificaciones para la gestión de OA.....	13
1.4.1 Modelo Referenciado de Objetos de Contenido Compartible.....	14
1.4.2 Sistema de Instrucción para la Gestión de Empaquetamiento de Contenidos	15
1.4.3 Fundamentación del estándar seleccionado.....	15
1.5 Metodologías de Desarrollo de Software	16
1.5.1 Programación Extrema	16
1.5.2 El Proceso Unificado de Desarrollo (RUP).....	17
1.5.3 Fundamentación de la metodología seleccionada	18
1.6 Lenguajes de programación del lado del servidor.....	18
1.6.1 PHP.....	18

1.6.2 Páginas de Servidor Activo.....	19
1.6.3 Fundamentación del lenguaje de programación del lado del servidor.....	20
1.7 Lenguajes de programación del lado del cliente	20
1.7.1 JavaScript.....	20
1.7.2 HTML	21
1.7.3 Fundamentación del lenguaje del lado cliente seleccionado.....	21
1.8 Lenguaje de Modelado	22
1.8.1 Lenguaje Unificado de Modelado	22
1.8.2 Lenguaje de modelado de procesos de negocio.....	22
1.8.3 Fundamentación del lenguaje de modelado seleccionado.....	23
1.9 Herramientas de Modelado Visual	23
1.9.1 Visual Paradigm para UML.....	23
1.9.2 Rational Rose.....	24
1.9.3 Fundamentación de la herramienta seleccionada.....	24
1.10 Frameworks de desarrollo	25
1.10.1 Symfony	25
1.10.2 Zend Framework	26
1.10.3 Fundamentación del framework seleccionado	27
1.11 Sistemas Gestores de Bases de Datos.....	27
1.11.1 MySQL	27
1.11.2 PostgreSQL.....	28
1.11.3 Fundamentación de la selección del SGBD.....	28
1.12 Base de Datos Nativas en XML	29
1.12.1 eXist	29
1.12.2 Tamino XML Server.....	29
1.12.3 Fundamentación de la selección de la Base de Datos Nativas en XML.....	30
1.13 Entorno de Desarrollo Integrado	30
1.13.1 NetBeans para PHP	30
1.13.2 Zend Studio	31

1.13.3 Fundamentación del IDE de desarrollo	31
1.14 Conclusiones parciales	32
CAPÍTULO#2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA	33
2.1 Introducción	33
2.2 Propuesta del Modelo de evaluación de la calidad de OA	33
2.3.1 Propuesta del cuestionario para la Categoría Contenido	35
2.3.2 Propuesta del cuestionario para la Categoría Técnica-Estética	36
2.4 Modelo de dominio	37
2.4.1 Definición de las clases de dominio	37
2.5 Requisitos del sistema	38
2.5.1 Requisitos Funcionales.....	38
2.5.2 Requisitos No Funcionales	39
2.6 Identificación de los actores del sistema.....	40
2.7 Identificación de los casos de uso del sistema.....	41
2.8 Diagrama de casos de uso del sistema	41
2.9 Descripción de los casos de uso.....	42
2.10 Conclusiones parciales	45
CAPÍTULO#3: ANÁLISIS Y DISEÑO DEL SISTEMA.....	46
3.1 Introducción	46
3.2 Análisis	46
3.2.1 Diagrama de Clases del Análisis	46
3.2.2 Diagramas de interacción (colaboración).....	49
3.3 Diseño	49
3.3.1 Diagrama de clases del diseño.....	50
3.3 Modelo de datos	50
3.4 Conclusiones parciales	50
CONCLUSIONES	51
RECOMENDACIONES.....	52
GLOSARIO DE TÉRMINOS	58

ANEXOS.....	60
-------------	----

INTRODUCCIÓN

En la historia de la humanidad el hombre ha tenido la necesidad de aprender para sobrevivir y desarrollarse. Primeramente aprendían de lo que la naturaleza les brindaba como enseñanza, para este entonces el hombre no tenía la preocupación del estudio, pero percibió la necesidad de transmitir los conocimientos adquiridos para las futuras generaciones. Al pasar de los siglos surge la enseñanza intencional, para el estudio de las diferentes materias y aspectos de la vida. Históricamente, las personas aprendían de sus maestros, que eran los que tenían mayor conocimiento.

En la actualidad el conocimiento se transmite de manera diferente y con el surgimiento de las TIC, ocurrieron muchos cambios que les brindan mayores comodidades a los usuarios, provocando gran impacto en el proceso de enseñanza-aprendizaje, ya que este se volvió más accesible para todos los interesados en superarse. La educación a distancia o el e-learning, como también se conoce es uno de esos cambios; el mismo ha sido definido de diversas maneras por diferentes autores, pero en el presente trabajo investigativo se tomará como *“un medio electrónico para el aprendizaje a distancia o virtual, donde se puede interactuar con los profesores por medio de la red”*.

(1)

El e-learning es considerado una de las alternativas más prometedoras para elevar el nivel de educación y capacitación de la sociedad de usuarios en el mundo a un costo mínimo. Con la creación de este fue creciendo la necesidad de compartir recursos de apoyo a la enseñanza y reutilizarlos en cualquier área del conocimiento. Al plantearse el problema muchos especialistas trataron de buscar la forma de resolver la interrogante de reutilizar el conocimiento, investigando la manera más factible de utilizar estos recursos. Se obtuvo como resultado de la investigación, el surgimiento del término Objeto de Aprendizaje (OA).

A los OA se les ha definido con el paso de los años de disímiles formas por diferentes especialistas en el tema, una de las tantas definiciones y que la autora está de acuerdo, aunque no se utilizará para el presente trabajo es la dada por la Dra. Clara López Guzmán donde define como *“cualquier recurso con una intención formativa, compuesto de uno o varios elementos digitales, descrito con metadatos, que pueda ser utilizado y reutilizado dentro de un entorno e-learning puede considerarse un OA.”*

(2)

Es importante destacar que a los OA se les caracteriza fundamentalmente por la reutilización en diferentes áreas del conocimiento, la cual trae consigo grandes ventajas, entre ellas el ahorro de tiempo y de esfuerzo para su creación. Para lograr

dicha reutilización es necesario que el OA se encuentre descrito de forma correcta a través de descriptores o metadatos, los mismos deben tener la información del autor y los derechos que a él se confieren, una breve descripción del OA, su nombre, la localización, entre otros datos; permitiendo de esta forma ser identificados. Entiéndase por metadatos a *"la información que describe entre otras, la calidad, distribución, actualidad y referencia espacial de un conjunto de datos"*. Muchos especialistas llaman a los metadatos como "datos acerca de los datos". (3)

Un Repositorio de Objetos de Aprendizaje (ROA) es: *"una herramienta diseñada para almacenar organizadamente OA, permitiendo así su localización, recuperación, catalogación y reutilización"* (4). En América Latina existen varios ROA, entre ellos se encuentra ESPOL de Ecuador; el de UAA de Aguascalientes en México; el de CSI de Colombia, este tiene como una de sus más importantes funcionalidades la evaluación en línea que permite la mejora de sus recursos de manera permanente.

La Universidad de las Ciencias Informáticas (UCI), como institución de avanzada en el uso de las TIC en su proceso docente-educativo, ha introducido los OA en función de mejorar la calidad de los recursos didácticos que en ella se desarrollan. La universidad cuenta con el centro de Tecnologías para la Formación (FORTES) que desarrolla software educativo como medios de enseñanza. Sus proyectos desarrollan OA para el apoyo al proceso docente educativo tanto dentro como fuera de la universidad, estos se almacenan en un ROA desarrollado en el centro, nombrado RHODA.

Debido al modelo de formación aplicado en la UCI, el cual se caracteriza por ser semipresencial y que la carrera que se estudia es la de Ciencias Informáticas, los estudiantes poseen grandes habilidades al interactuar con las tecnologías para estudiar, utilizando para ello los recursos didácticos que se encuentran publicados en la universidad; esto conlleva a que se eleve la exigencia en la calidad de los OA almacenados en el repositorio, que han sido desarrollados por el proyecto CRODA u otros proyectos. Por ello, se hace imprescindible la evaluación de la calidad de los OA para su utilización y reutilización en productos futuros así como la compatibilidad con otros repositorios.

A nivel mundial se evalúa la calidad de los objetos de aprendizaje tomando como referencia instrumentos o modelos de evaluación tales como HEODAR, LORI, entre otros. Estos se centran en evaluar el contenido desde el punto de vista formativo en apoyo al proceso de enseñanza-aprendizaje, otros de los aspectos que miden es el diseño y presentación que posee el OA, además de la parte técnica del mismo .

De estas herramientas antes mencionadas la que más se utiliza en la universidad es el instrumento LORI; que facilita al especialista un cuestionario que contiene nueve

variables que se calificarán en una escala de 1 a 5 puntos de acuerdo con la información contenida en el OA, si la variable no es relevante para la evaluación del OA o el especialista no se siente capacitado para juzgar una variable concreta, entonces puede marcar la opción No Aplica (NA) brindando la final el resultado de los puntos al especialista. El proceso de evaluación en el centro FORTES no se está llevando a cabo correctamente, debido a que los especialistas del centro solo utilizan un manual de usuario perteneciente al LORI como guía a seguir para realizar la evaluación.

Actualmente en CRODA y RHODA proyectos encargados del manejo de OA en el centro realizan la evaluación de la calidad de forma manual, visualizando el OA desde el repositorio, usando como guía el manual de usuario de LORI, obteniendo una calificación de acuerdo con la puntuación de cada una de las características que propone el mismo. Sumándole a esto que la calificación y opinión del estado del OA no se guardan ni son informados al autor.

Por lo antes expuesto surge la siguiente **situación problemática** Inexistencia de un módulo para la revisión de la calidad de los OA en los proyectos CRODA y RHODA que facilite el apoyo al proceso de evaluación de los OA previamente creados. A partir de lo antes expuesto se tiene como **problema a resolver**: ¿Cómo generar los artefactos necesarios para realizar la implementación del módulo Revisión de la calidad de los OA en los proyectos CRODA y RHODA?

El **objeto de estudio** se centra en el proceso de análisis y diseño de un módulo encargado de evaluar OA para aplicaciones web.

Como **objetivo general** se tiene Análisis y diseño del módulo Revisión de la calidad de OA para los proyectos CRODA y RHODA. Para dar cumplimiento a este se trazan los siguientes **objetivos específicos** desglosados en:

- Definir la propuesta del modelo de evaluación de la calidad de los OA producidos por el centro FORTES.
- Realizar la captura de requisitos.
- Realizar análisis y diseño de la propuesta de solución.

El **campo de acción** se centra en el proceso de análisis y diseño de un módulo que se encargue de evaluar la calidad de los OA para las aplicaciones CRODA y RHODA.

Para guiar el desarrollo de este proceso se plantea como **idea a defender**: Con el desarrollo del análisis y diseño del módulo de Revisión de la calidad de los OA de CRODA y RHODA se garantizarán la obtención de los artefactos necesarios para la implementación de dicho módulo que agilizará el proceso actual en el centro FORTES, con un alto valor agregado y facilidad para el especialista.

Las **tareas a desarrollar** son las siguientes:

1. Revisión y actualización de la bibliografía relacionada con el tema a tratar en la investigación.
2. Investigar el estado del arte de los modelos de evaluación de la calidad de los OA.
3. Levantamiento de requisitos.
4. Revisión los requisitos funcionales definidos.
5. Diseñar los casos de uso definidos como críticos en el sistema.
6. Modelar los casos de uso definidos como críticos en el sistema.
7. Estructuración del documento de tesis.

En esta investigación, para dar cumplimiento a las tareas es necesario utilizar los siguientes **métodos científicos**:

Los **métodos teóricos** que se utilizaron fueron los siguientes:

Histórico-lógico: A través de este método se profundizó en la evolución de la educación a través de los años, así como los cambios ocurridos en los modelos de evaluación existentes en el mundo.

Analítico-Sintético: En el proceso de investigación se hizo un estudio de los diferentes modelos de evaluación de la calidad de los OA existentes en otras universidades del mundo, luego se realizó un análisis y se propuso un modelo para llevarlo a cabo en el módulo a implementar. Este método facilitó además, la elección de la metodología de desarrollo a usar, los lenguajes de programación, y las herramientas que se utilizaron para llevar a cabo el análisis y diseño del módulo propuesto; además de las herramientas y tecnologías para su posterior implementación.

Modelación: Este método permitió llevar a cabo los diagramas generados en el presente trabajo utilizando el lenguaje de modelado UML.

Los **métodos empíricos** que se utilizaron fueron los siguientes:

La **observación** de la evaluación de la calidad de los OA en el centro.

La **revisión de documentos:** con el fin de seleccionar y recopilar la información necesaria de los antecedentes históricos de los OA, la calidad de los mismos y la manera en que se lleva a cabo para la evaluación de la calidad de los OA.

El presente trabajo de diploma estará estructurado en un máximo de 80 páginas, el mismo se dividirá en tres capítulos. Para su mayor entendimiento a continuación se detallará cada sección:

Capítulo#1: Fundamentación teórica.

En este capítulo se construye el marco teórico y referencial de la investigación referido a la evaluación de la calidad de los objetos de aprendizaje a partir del análisis de la bibliografía consultada. Se hace referencia a las definiciones de los conceptos de OA, las características, las partes que lo componen, así como los modelos de evaluación de la calidad en otras universidades del mundo y el porqué no se utilizan. Se realizó un estudio de las herramientas y tecnologías más factibles para el análisis y diseño, así como las propuestas para una futura implementación del módulo.

Capítulo#2: Descripción de la solución propuesta.

En este capítulo se realiza la descripción de la propuesta de solución del problema a resolver. Se propone el modelo de evaluación de la calidad de los OA a aplicar en el módulo. También se hace referencia a los artefactos generados como el modelo de dominio, los requisitos funcionales y los no funcionales, diagrama de casos de uso y la descripción de los casos de uso para un mejor entendimiento por el desarrollador.

Capítulo#3: Análisis y diseño del sistema.

En este capítulo se generan el resto de los artefactos del flujo de trabajo Análisis y diseño de la metodología RUP: diagrama de clases del análisis, diagrama de colaboración, diagrama de clases del diseño, el diagrama de casos de uso del sistema y el modelo de datos.

CAPÍTULO#1: FUNDAMENTACIÓN TEÓRICA.

1.1 Introducción

En el presente capítulo se abordan algunos conceptos referentes a las tendencias y tecnologías actuales a través de un estudio profundo del estado del arte, en las cuales se hace referencia a conceptos, características y componentes de los OA; así como el análisis de modelos de evaluación de la calidad de OA, para ser usados en el módulo que se desea desarrollar. En esta sección se tratan además, los estándares que existen mundialmente para la encapsulación, descripción y la interoperabilidad entre sistemas referentes a los OA. Se fundamenta la selección de la metodología de desarrollo de software, tecnologías a utilizar en la web, servidores web, Sistemas Gestores de Base de Datos (SGBD) y lenguajes de programación.

1.2 Objeto de Aprendizaje

Formalmente no hay una única definición del concepto de objeto de aprendizaje y las definiciones son muy amplias. Wiley plantea en el 2000 que un OA es *“cualquier recurso digital que se puede utilizar como apoyo para el aprendizaje (todo tipo de archivo digital como texto, video, artículo, página Web, etc.)”*; la definición que reelaboró el profesor Jorge Enrique Gil de la Universidad de La Habana donde entiende oportuno hacer una modificación a la definición antes dada quedando de la siguiente forma, *“cualquier recurso digital estructurado que puede ser reutilizado para alcanzar un objetivo de aprendizaje”*. La doctora Clara López Guzmán describe que es un objeto de aprendizaje abarcándolo de manera general, definiéndolo como *“cualquier recurso con una intención formativa, compuesto de uno o varios elementos digitales, descrito con metadatos, que pueda ser utilizado y reutilizado dentro de un entorno e-learning puede considerarse un OA”*. (2)

Para el presente trabajo se asume la redefinición realizada por el proyecto cliente RHODA, quedando de la siguiente forma: *“objeto de aprendizaje puede ser cualquier recurso digital con una granularidad apropiada y una marcada intención formativa, compuesto por uno o varios objetos de información, con un único objetivo, descrito con metadatos y con un comportamiento secuenciado que asegure el correcto enlace entre*

los elementos de su estructura didáctica y que puede ser reutilizado en entornos e-learning". De los conceptos estudiados este es el que más se encuentra en correspondencia a lo que en la práctica se entiende como un OA.

1.2.1 Características de un OA

Los OA son utilizados y reutilizados para la creación de nuevos OA definidos para otro contexto, para lograr la calidad de la creación de OA a continuación se expondrán algunos de los requisitos que deben cumplir los mismos:

- *Formato digital:* en este formato el mismo permite que sea modificado o actualizado constantemente, por lo que puede ser utilizable y accesible por la red, por diferentes usuarios simultáneamente en diferentes lugares.
- *Intención pedagógica:* el OA incluye no sólo contenidos sino que también guía el propio proceso de aprendizaje del estudiante.
- *Contenido interactivo:* implica la participación activa de cada individuo (profesor-alumno) en el intercambio de información. Para ello es necesario que el objeto incluya actividades como: ejercicios, simulaciones, cuestionarios, diagramas, gráficos, diapositivas, tablas, exámenes, experimentos entre otros, que permitan facilitar el proceso de asimilación y el seguimiento del progreso de cada alumno.
- *Indivisible e independiente:* no puede descomponerse en partes más pequeñas, debe tener sentido en sí mismo y ser autocontenido.
- *Reutilizable:* en cualquier contexto educativo para el que fue creado, siendo esta característica la que determina el valor de un OA.
- *Interoperables.* Pueden operar entre diferentes plataformas de hardware y software.
- *Durables.* Deben permanecer intactos a las actualizaciones de software y hardware. (5)

Conforme con lo anterior, la autora del presente trabajo de diploma opina que si los objetos de aprendizaje creados en el centro FORTES cumplieran con estas características pues los mismos serian de mayor calidad.

1.2.2 Principales componentes de un OA

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

- *Unidad de Información:* Contenidos multimedia individuales (texto, imágenes, audio, vídeo) en la que se tiene la posibilidad de generar contenido textual mediante el acceso a través de editores de texto.
- *Unidad de Contenido:* Define la ubicación en la que se encuentran albergados los contenidos, facilitando la generación de plantillas.
- *Unidad Didáctica:* Abarca cada uno de los elementos que permiten generar planteamientos de aprendizaje significativo, determinar criterios de evaluación, contenidos, recursos y actividades de enseñanza-aprendizaje. (6)

La Universidad del BioBio de Chile en unión con la Universidad de Castilla-La Mancha de España define como partes de un OA al Contenido, Metadato, Diseño Instruccional y Diseño Estético. (7)



Ilustración 1: Componentes de un OA. Universidades de BioBio de Chile y Castilla-La Mancha de España.

La Universidad de las Ciencias Informáticas utiliza los componentes de los Objetos de Aprendizaje como se muestran en la figura 2: Objetos de información, descritos por metadatos y estructurados didácticamente de manera correcta.

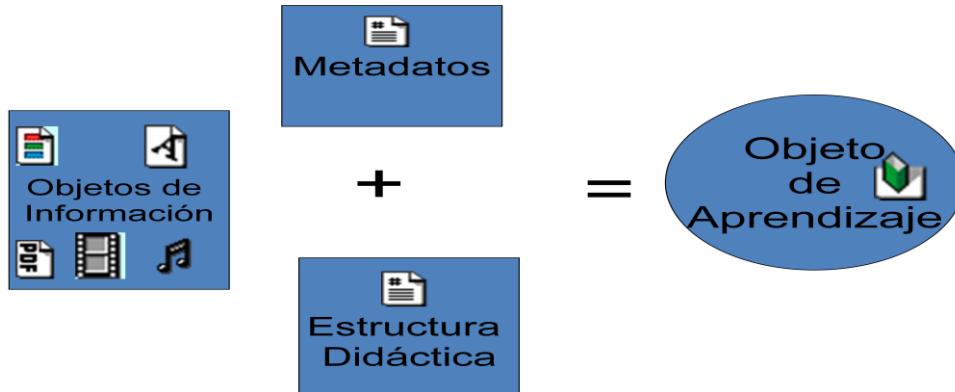


Ilustración 2: Componentes del Objeto de Aprendizaje utilizado en la UCI. (8)

Objetos de Información:

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

Metadatos:

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

Estructura didáctica:

Es la organización del contenido educativo basada en un diseño instruccional, definida pedagógicamente.

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

1.3 Modelos de evaluación de la calidad de OA

En la actualidad existen muchas maneras de evaluar un objeto de aprendizaje, estas pueden ser guías, modelos o herramientas teniendo como objetivo fundamental facilitar proceso de evaluación a los especialistas y permiten tener en cuenta aspectos que se desean seguir con rigurosidad y exactitud de manera ordenada.

1.3.1 Universidad Autónoma de Aguascalientes de México

La Universidad Autónoma de Aguascalientes ha ido evolucionando la conceptualización de los elementos que determinan la calidad de un OA, actualmente se reconocen, los elementos tecnológicos, pedagógicos, de contenido y elementos estéticos.

Los aspectos técnicos pueden abordarse desde la perspectiva de ingeniería de software considerando que se está frente a un producto de software del desarrollo orientado a objetos.

En el aspecto tecnológico se pueden mencionar todos los elementos que pueden proporcionar las ventajas que se atribuyen a los productos desarrollados bajo el paradigma del desarrollo orientado a objetos como la reutilización y la adaptabilidad, también se hace necesario mencionar algunas propiedades de cualquier software de calidad como es la compatibilidad y la eficiencia.

En los aspectos pedagógicos se pueden mencionar a aquellos que facilitan el proceso de enseñanza-aprendizaje, ejemplo de ello, que el objetivo se encuentre bien especificado, el número de medios utilizados, así como los ejemplos y la evaluación que se realiza.

El aspecto pedagógico, en la Universidad de Aguascalientes puede expresarse empleando la taxonomía de Bloom, con la misma pueden darse cuenta si elementos como la experimentación, evaluación, colaboración y relación realmente están proporcionando el tipo de aprendizaje que se requiere.

En el aspecto de contenido tenemos aquellos elementos que brindan información sobre el nivel de complejidad del tema que se trata en el OA y el nivel de detalle que se aborda en el mismo.

Los OA se asemejan a las características de los sistemas y aplicaciones basados en Web, donde la apariencia de estas y la disposición de sus elementos son de suma importancia para que sea amigable y atractiva. Si estas aplicaciones se hacen con el fin de comercializarse o vender estos productos, la estética de estas va a ser un factor clave para el éxito, como lo puede ser la parte técnica.

En la propuesta de la Universidad de Aguascalientes los aspectos estéticos de un OA se refiere a la presentación de la información que el mismo posea (fuentes, colores, tamaño, en sí todos los elementos de formato) y la disposición de la misma (colocación simétrica o asimétrica). Además, estos aspectos van a determinar un fácil,

rápido y adecuado uso del OA, como contar con una ordenada proporcionalidad y disposición de los elementos que lo conforman. (9)

1.3.2 Universidad de Salamanca de España

En la de Universidad de Salamanca un grupo de expertos en OA decidieron realizar un estudio para la evaluación de los mismos creados por su universidad; el modelo que utiliza la herramienta consiste en dos pasos:

Clasificar los objetos de aprendizaje según su nivel cognitivo: se sugiere asociar los objetivos de los OA con alguno de los niveles de dominio cognitivo de la taxonomía de Bloom. Los niveles se encuentran clasificados como: de menor complejidad (conocimiento, comprensión y aplicación) y mayor complejidad (análisis, síntesis y evaluación); cada uno de estos dominios indica lo que el alumno es capaz de hacer.

Clasificar los objetos en tres tipos de contenidos: datos y conceptos, procedimientos y procesos y, finalmente, reflexión y actitudes. La clasificación de contenidos como datos y conceptos está dirigida a los objetos que contengan información básica para enseñar algo. Los contenidos de los objetos clasificados como procedimientos y procesos se relacionan a la enseñanza de pasos o fases de algún proceso. Los tipos de contenidos que promueven la reflexión y la adopción de una actitud, ya sea de forma consciente o inconsciente están relacionados al aprendizaje de principios, normas, etc.

La herramienta y metodología de evaluación aquí propuesta está diseñada para la revisión de OA que corresponden a unidades mínimas de reutilización. Sobre esta base se proponen diversas categorías de evaluación con criterios que permiten evaluar los OA desde diversos puntos de vista con lo que es posible conseguir una evaluación integral de éstos. Por otra parte, esta herramienta posibilita la evaluación de cada criterio de forma individual lo que permite una valoración más exacta.

A continuación se hacen mención de las categorías de evaluación, sus criterios y su relación con los metadatos.

- *Categoría didáctico-curricular:* permite evaluar el objeto de acuerdo con el objetivo y dónde va a ser aplicado. Además del contenido que posee el mismo.
- *Categoría técnica-estética:* permite evaluar aspectos asociados al diseño de los objetos.
- *Categoría funcional:* la evaluación del funcionamiento de los objetos.

En cuanto a la metodología de evaluación de los objetos idealmente deberán participar diversos expertos relacionados a los objetos de aprendizaje: diseñadores instruccionales, diseñadores gráficos, profesores, etc. De esta manera, será posible obtener opiniones expertas sobre cada una de las categorías. Los expertos deberán evaluar los OA de forma individual y posteriormente colaborativa, de esta manera, se podrá contrastar la evaluación inicial con la de los demás expertos y finalmente tratar de llegar a un consenso sobre la calidad de los objetos. Para realizar la evaluación cada uno de los revisores deberá puntuar con el instrumento el criterio de evaluación con el siguiente rango. 0 si el OA no cumple con el criterio, 1=muy bajo, 2=bajo, 3=medio, 4=alto, 5=muy alto. Este rango permite una valoración precisa sobre el nivel de calidad porque la numeración está relacionada con palabras que indican su significado. Finalmente, se promediará la puntuación de cada categoría y entre ellas se sacará el promedio final que reflejará la calidad obtenida por el objeto. (10)

1.3.3 Universidad del BioBio de Chile en unión con la Universidad de Castilla-La Mancha de España

Las universidades BioBio y Castilla-La Mancha se unieron para realizar en conjunto una propuesta de guía para evaluar la calidad de los OA. Ellos primeramente realizan un análisis de estudios donde se discuten y proponen dimensiones de los OA que se relacionan con la calidad, después se proponen finalmente las características y rasgos que sirvan para la evaluación de los OA, basándose en la Norma ISO/IEC 9126.

La propuesta de evaluación de los OA considera establecer las características, sub-características y rasgos que pueden facilitar la medición. La mayor parte de las características y sub-características de calidad están basadas en una adaptación al estándar ISO 9126. (11)

1.3.4 Instrumento de Revisión de Objetos de Aprendizaje

Learning Object Review Instrument (LORI) propone un marco para la evaluación de OA, basado en el análisis de nueve variables. Cada variable se evalúa mediante una escala de cinco niveles. Las nueve dimensiones de los OA evaluadas por este instrumento son: Calidad de los contenidos, Adecuación de los objetivos de Aprendizaje, Feedback (retroalimentación) y Adaptabilidad, Motivación, Diseño y presentación, Usabilidad, Accesibilidad, Reusabilidad y Cumplimiento de estándares.

Esta evaluación individual puede ser complementada con una evaluación colaborativa denominada Modelo de Participación Convergente. En este modelo la evaluación la desarrolla un grupo de evaluadores quienes mediante actividades de evaluación individual y luego discusiones grupales califican al OA. (12)

1.3.5 Fundamentación del modelo seleccionado

Al realizar un análisis de los modelos anteriormente analizados, la autora cree conveniente que no se debe utilizar ninguna de estas herramientas o modelos, porque son independientes o aplicables a un sistema específico, definido para las necesidades de sus universidades lo que resultaría difícil realizar la integración de alguno de ellos en las plataformas CRODA y RHODA, sin darle solución de un todo la situación que existe en la actualidad.

Como decisión final, luego de un análisis la autora del presente trabajo de diploma cree conveniente desarrollar un nuevo modelo de evaluación de la calidad de OA basados en la integración de las ventajas de cada uno de los modelos estudiados como propuesta de solución [Ver epígrafe 2.2].

1.4 Estándares y especificaciones para la gestión de OA

Con el uso de las TIC en la educación, han surgido diversos productos educativos, su aparición ha traído como resultado que se busquen vías para lograr una estandarización entre ellos. Entre estos software educativos se persigue como objetivo fundamental el logro de una mejor comunicación, reutilización, accesibilidad, interoperabilidad y durabilidad entre ellos.

En la actualidad se busca llegar a un entendimiento entre los procesos relacionados con los estándares y especificaciones del e-learning para hacerlos de una manera más eficiente. Muchos de estos definen un modelo de empaquetamiento estándar para los contenidos donde se especifica cómo representar la estructura de la información y cómo agregar distintos recursos educativos entre diversos entornos.

Los contenidos pueden ser empaquetados como OA, estos deben ser creados por los programadores de modo que sean reutilizables e integrados a otros cursos, además de garantizar la durabilidad de estos. Existen otros estándares que normalizan aspectos como la descripción de los OA de modo que puedan ser gestionados, indexados, clasificados y almacenados en repositorios. (13)

El empleo de los estándares facilita la interoperabilidad entre los sistemas de e-learning, ya que estos pueden intercambiar materiales empaquetados según los estándares definidos y adoptados.

En la actualidad existen múltiples propuestas de diversas organizaciones en cuanto se refiere a estándares de empaquetamiento referente a los OA, solo dos de ellos se adaptaron a las necesidades del cliente por sus características. Estas propuestas están vinculadas a los estándares: Sistema de Instrucción para la Gestión de Empaquetamiento de Contenidos (en inglés, IMS Content Packaging o IMS CP) y Modelo Referenciado de Objetos de Contenido Compartible (por sus siglas en inglés, SCORM), encontrándose ambos con una estrecha relación debido a la integración que hace SCORM de IMS CP.

1.4.1 Modelo Referenciado de Objetos de Contenido Compartible

Modelo Referenciado de Objetos de Contenido Compartible (Shareable Content Object Reference Model, en lo adelante SCORM) es un conjunto de estándares y especificaciones para compartir, reutilizar, importar y exportar OA, utilizado en el mundo para proyectos educativos a distancia. (14)

Este modelo describe cómo las unidades de contenidos se relacionan unas con otras, se comunican los contenidos con el LMS y define cómo empaquetar los contenidos para importarse y exportarse entre plataformas, además puntualiza algunas formas en las que el desarrollador representa un contenido en el LMS. SCORM indica cómo crear materiales que puedan ser reutilizados y compartidos a través de diferentes sistemas.

Los principales requerimientos que el modelo SCORM trata de satisfacer son:

- *Accesibilidad*: la capacidad de ubicar y de tener acceso a componentes educativos desde una ubicación remota y de entregarlos a muchas otras ubicaciones.
- *Adaptabilidad*: la capacidad de personalizar la enseñanza a las necesidades de individuos y organizaciones.
- *Rentabilidad*: la capacidad de aumentar la eficiencia y productividad reduciendo el tiempo y los costos implicados en entregar la enseñanza.
- *Durabilidad*: la capacidad de soportar la evolución y cambios tecnológicos sin reajuste, reconfiguración o recodificación que impliquen un alto costo.
- *Interoperabilidad*: la capacidad de tomar componentes educativos desarrollados en una ubicación con un conjunto de herramientas o con una

plataforma y utilizarlas en otra ubicación con un conjunto de herramientas o una plataforma distinta.

- *Reutilidad*: la flexibilidad de incorporar componentes educativos en múltiples aplicaciones y contextos. (14)

1.4.2 Sistema de Instrucción para la Gestión de Empaquetamiento de Contenidos

El Sistema de Instrucción para la Gestión de Empaquetamiento de Contenidos (IMS Content Packaging, en adelante IMS CP) es un estándar para el empaquetamiento de contenidos que normaliza la estructura interna de los recursos didácticos incluidos en un paquete y su organización. (15)

IMS CP es uno de los estándares de empaquetamiento de contenidos más populares en el mundo, dentro de sus características están:

- *Gestión de contenidos*: permite describir estructuras de datos que pueden ser intercambiados entre sistemas que deseen importar, exportar, agregar, y eliminar paquetes de contenido. También permite juntar, estructurar y agregar contenido en una variedad de formatos considerable.
- *Exportación de contenidos*: Este estándar permite exportar contenido de un Sistema de Gestión de Aprendizaje (Learning Management System, en adelante LMS) hacia otro, mientras haya información que está siendo descrita por los paquetes de contenidos de IMS CP.
- *No cuenta con modelos pedagógicos*: Se basa en el empaquetamiento y distribución de recursos. Además, se enfoca solo a la descripción y la estructura de los contenidos. Este estándar no se involucra con los modelos pedagógicos, dejando al profesor que utilice el que crea adecuado.

1.4.3 Fundamentación del estándar seleccionado

Para la selección del estándar de gestión de Objetos de Aprendizaje, las características que posee SCORM hacen factible el sistema a desarrollar observando su amplio empleo a nivel mundial y ser un modelo de referencia que contiene varios estándares y especificaciones. Este estándar tiene como ventajas que permite crear materiales que luego pueden ser reutilizados y compartidos entre otros sistemas. SCORM hace referencia a otros estándares, específicamente, a la descripción de OA

como Metadatos para Objetos de Aprendizaje (por sus siglas en inglés, LOM), permitiendo así utilizar el indicado, dejando libre opción a los desarrolladores.

1.5 Metodologías de Desarrollo de Software

Las metodologías aplican disciplina en el desarrollo de software con el fin de hacerlo más predecible y eficiente. En un proyecto de desarrollo de software la metodología define quién debe hacer qué, cuándo y cómo debe hacerlo. No existe una metodología de software universal. Las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigen que el proceso sea configurable.

1.5.1 Programación Extrema

Programación Extrema (del inglés “Extreme Programming”, en adelante XP) es la metodología más popular de las metodologías ágiles. Entre sus principales características se encuentran:

- *Retroalimentación con el cliente:* Al menos uno de los miembros del equipo de trabajo del proyecto, es un cliente. Propiciando así, una constante interacción del mismo con el producto.
- *Cortas iteraciones:* En cada iteración, se obtiene un producto listo para entregar y que tiene valor para el cliente. La entrega continua de resultados compromete a ambas partes en la evolución del proyecto e indirectamente influye de forma positiva en la calidad del producto final.
- *Muy flexible a cambios:* Una de las características más reconocidas de XP. La constante retroalimentación con los clientes permite prever futuros cambios y evita llegar a momentos que paralizan el desarrollo del producto. (16)

Cuenta con una serie de prácticas que van en vías de aumentar la productividad del equipo de trabajo, tales como la programación en pares, reuniones diarias y planes de entrega a corto plazo, por mencionar algunos. Al igual que RUP, también utiliza UML si el equipo de desarrollo lo decide.

1.5.2 El Proceso Unificado de Desarrollo (RUP)

RUP es una metodología de desarrollo de software de las más utilizadas en el mundo actualmente, está pensado para el análisis, implementación y documentación de sistemas orientados a objetos. Su ciclo de vida se caracteriza por ser:

- *Iterativo e Incremental:* El Proceso Unificado es un marco de desarrollo iterativo e incremental compuesto de cuatro fases denominadas Inicio, Elaboración, Construcción y Transición. Cada una de estas fases es a su vez dividida en una serie de iteraciones (la de inicio sólo consta de varias iteraciones en proyectos grandes). Estas iteraciones ofrecen como resultado un incremento del producto desarrollado que añade o mejora las funcionalidades del sistema en desarrollo. Cada una de estas iteraciones se divide a su vez en varias disciplinas que recuerdan a las definidas en el ciclo de vida clásico o en cascada: Análisis de requisitos, Diseño, Implementación y Prueba. Aunque todas las iteraciones suelen incluir trabajo en casi todas las disciplinas, el grado de esfuerzo dentro de cada una de ellas varía a lo largo del proyecto.
- *Dirigido por los casos de uso:* En el Proceso Unificado los casos de uso se utilizan para capturar los requisitos funcionales y para definir los contenidos de las iteraciones. La idea es que cada iteración tome un conjunto de casos de uso o escenarios y desarrolle todo el camino a través de las distintas fases: diseño, implementación, prueba.
- *Centrado en la arquitectura:* El proceso unificado asume que no existe un modelo único que cubra todos los aspectos del sistema. Por dicho motivo existen múltiples modelos y vistas que definen la arquitectura de software de un sistema.

RUP define nueve flujos de trabajo de los cuales seis son de ingeniería y los restantes de apoyo, cada flujo está destinado a actividades específicas y están divididos en cuatro fases. Estos flujos se desarrollan en paralelo durante todo el ciclo de desarrollo, estando presente unos más que otros en las diferentes iteraciones. (17)

RUP está basado en componentes, que utiliza el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema software. Es utilizado principalmente en proyectos grandes y a largo plazo. (18)

1.5.3 Fundamentación de la metodología seleccionada

Después de un estudio detallado de las metodologías de desarrollo de software se propone usar RUP, pues la misma genera toda la información necesaria durante las fases de análisis y diseño para implementar el módulo de Revisión de la calidad de OA no siendo así con la metodología XP. RUP se adapta a las características propias del proyecto; el tamaño del mismo, su tipo, así como las regulaciones que lo condicionan, las cuales influirán en su diseño específico. Se divide en fases y se entrega el producto cada vez que se realiza una iteración.

1.6 Lenguajes de programación del lado del servidor

Actualmente existen diferentes lenguajes de programación para desarrollar en la web, estos han ido surgiendo debido a las tendencias y necesidades de las plataformas. En el presente epígrafe se pretende mostrar las ventajas y desventajas de algunos de los más conocidos y selección del uno de ellos.

1.6.1 PHP

Preprocesador de Hipertextos (Hypertext Preprocessor, en lo adelante PHP) es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. (19)

Dentro de sus características se destaca:

- *Soporte para PHP 4:* su potencia, su alto rendimiento, su facilidad de aprendizaje, su escaso consumo de recursos y la documentación que posee.
- *Conexiones a diferentes servidores de base de datos:* posee otras características como la conexión a diferentes tipos de servidores de bases de datos tales como PostgreSQL, Microsoft SQL Server, Oracle, MySQL.
- *Multiplataforma:* Este lenguaje también presenta la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como UNIX (como Linux o Mac OS) y Windows, y puede además interactuar con los servidores de web más populares.
- *Lenguaje del lado servidor:* lenguajes de lado servidor más extendidos en la Web, este permite insertar sus pequeños fragmentos de código dentro de la página HTML y realizar determinadas acciones de una forma fácil y eficaz sin tener que generar programas desarrollados íntegramente en un lenguaje distinto de HTML.

Algunas de las ventajas que lo hacen ser tan utilizado por los programadores son:

- Es un lenguaje multiplataforma.
- PHP tiene la capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad. Una de las bases de datos que presenta mayor conectividad con PHP es MySQL.
- Posee además la capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones. (20)

1.6.2 Páginas de Servidor Activo

Páginas de Servidor Activo (Microsoft Active Server Page, en lo adelante ASP) es una tecnología creada por Microsoft, destinada a la creación de sitios web. No se trata de un lenguaje de programación en sí mismo (ya que los ASP se pueden programar en VBScript, JavaScript, PerlScript o en varios lenguajes), sino de un marco sobre el que construir aplicaciones para la web. (21)

Algunas de las características de ASP son:

- El código ASP se puede mezclar con el código HTML en la misma página.
- El código ASP se puede escribir con un simple editor de textos como el Bloc de notas de Windows o UltraEdit.
- Como el código ASP se ejecuta en el servidor, y produce como salida código HTML puro, su resultado es entendible por todos los navegadores existentes.
- Mediante ASP se pueden manipular bases de datos (consultas, actualizaciones, borrados, etc.) de prácticamente cualquier plataforma, con tal de que proporcione un driver OLEDB u ODBC.
- ASP permite usar componentes escritos en otros lenguajes (C++, Visual Basic, Delphi), que se pueden llamar desde los archivos o desde las clases ASP.

- Sin modificar la instalación, los archivos ASP se pueden programar en JScript o VBScript (este último es el más usado porque más programadores lo dominan), pero también existen otros lenguajes, como Perlscript y Rexx, que se pueden emplear para programar ASP.
- Se ha portado a la plataforma Java por Chili! Soft y Halcyon Software, lo que permite que ASP sea usado en casi cualquier sistema operativo. (22)

1.6.3 Fundamentación del lenguaje de programación del lado del servidor

En la actualidad los lenguajes propuestos en esta investigación son reconocidos y utilizados a nivel mundial por su potencia y seguridad en la programación web. En esta investigación se utilizará PHP 5.5, porque es un lenguaje multiplataforma. Tiene la capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad. PHP es un lenguaje libre, por lo que se presenta como una alternativa de fácil acceso para todos. Además, permite las técnicas de Programación Orientada a Objetos. Y ASP es propietaria, creada por los desarrolladores que trabajan para Microsoft. Otras de las razones por lo que se selecciona a PHP como lenguaje de programación es porque el mismo es utilizado para la implementación de los proyectos CRODA y RHODA.

1.7 Lenguajes de programación del lado del cliente

Son los lenguajes que basan su procesamiento en el cliente web, es decir, que se ejecutan en el navegador del usuario, a continuación se hará referencia las ventajas de algunos de ellos y la selección de uno de ellos para su uso el futuro seguimiento del presente trabajo.

1.7.1 JavaScript

JavaScript es el lenguaje que permite interactuar con el navegador de manera dinámica y eficaz, proporcionando a las páginas web dinamismo y vida. (23)

Aclarar que JavaScript no es un lenguaje de programación propiamente dicho. Es un lenguaje script u orientado a documento, como pueden ser los lenguajes de macros que tienen muchos procesadores de texto. Con JavaScript, podrás mejorar tu página Web con algunas funciones para hacerla más amena y atractiva. (24)

Algunas de las características de JavaScript:

- Es interpretado por el cliente.

- Está basado en objetos.
- JavaScript no emplea clases ni herencia, típicas de la OOP.
- Su código se integra en las páginas HTML, incluido en las propias páginas.
- No es necesario declarar los tipos de variables que van a utilizarse.
- Las referencias a objetos se comprueban en tiempo de ejecución, por lo tanto, no se compila.
- No tiene acceso a escritura en HDD de las máquinas clientes como medida de seguridad.
- La ventaja que presenta JavaScript sobre el HTML es que permite crear páginas más dinámicas, lo que las hace más atractivas para el usuario.

1.7.2 HTML

HTML significa (HyperText Mark-Up Language), traducido al español sería, Lenguaje para el Formato de Documentos de Hipertexto. Es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con enlaces (hyperlinks) que conducen a otros documentos o fuentes de información relacionadas, y con inserciones multimedia (gráficos, sonido...) La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc.) así como los diferentes efectos que se quieren dar (especificar los lugares del documento donde se debe poner cursiva, negrita, o un gráfico determinado) y dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado (como Mosaic, o Netscape).

1.7.3 Fundamentación del lenguaje del lado cliente seleccionado

Actualmente, los lenguajes de programación del lado cliente son muy utilizados para brindar una interfaz más amigable al usuario. Para la implementación de esta herramienta se utilizará JavaScript, por ser un lenguaje de scripting seguro y fiable. Además, en este lenguaje, los scripts del mismo tienen capacidades limitadas, por razones de seguridad. Otra de las ventajas de este lenguaje es que el código es visible y puede ser leído por cualquiera, permitiendo así la retroalimentación del desarrollador, aunque para evitar fraude existen leyes de copyright. JavaScript permite el manejo de entornos visuales teniendo ventajas sobre HTML. Además, como razón

fundamental es usado en las plataformas CRODA y RHODA como lenguaje de programación del lado cliente en aplicaciones Web.

1.8 Lenguaje de Modelado

El modelado de sistemas de software es una técnica para tratar con la complejidad inherente a estos sistemas. El uso de modelos ayuda al ingeniero de software a "visualizar" el sistema a construir. Además, los modelos de un nivel de abstracción mayor pueden utilizarse para la comunicación con el cliente.

1.8.1 Lenguaje Unificado de Modelado

UML conocido en español como el Lenguaje Unificado de Modelado desempeña un papel protagónico en cualquier fase de una metodología de desarrollo. Es un lenguaje que permite modelar artefactos, relaciones, entidades, procesos, métodos de una situación determinada. Cabe recalcar que UML no es un lenguaje de programación, su objetivo es solo modelar situaciones, y se encuentra muy relacionado con la programación Orientada a Objetos, siendo el complemento que permite culminar de una forma directa lo que se modela con UML. En UML existen varios tipos de diagramas los cuales se agrupan en varias categorías por el tipo de ente que modelan. En la versión 2.0 existen 13 tipos de diagramas de forma categorizada, en esta categorización existen dos clasificaciones básicas basándose en el fin que persiguen los modelos que agrupa (25)

- *Diagramas de estructura:* agrupa seis diagramas dedicados a especificar cómo se modela desde un punto de vista estructural alguna situación del problema a resolver.
- *Diagramas de comportamiento:* agrupa siete diagramas dedicados a modelar el comportamiento, ya sea del sistema o de algún proceso del negocio que se pretende automatizar o servicio a brindar.

1.8.2 Lenguaje de modelado de procesos de negocio

Lenguaje de modelado de procesos de negocio (BPML, por sus siglas en inglés, Business Process Modeling Languages). Es utilizado para el modelado de procesos de negocio. Define un modelo abstracto y la gramática para expresar procesos genéricos, lo que le permite ser usado para una variedad de propósitos entre los que se incluyen:

definición de procesos empresariales, definición de complejos servicios Web y definición de colaboraciones multiparte. Unifica la semántica de definición de proceso, permitiendo desarrollar un solo procesador que pueda ejecutar cualquier proceso de negocio.

1.8.3 Fundamentación del lenguaje de modelado seleccionado

El BPML, presenta características bastante significativas, no es considerado como el más adecuado para ser utilizado en el desarrollo del futuro sistema, ya que se enmarca sólo en los procesos del negocio y precisamente lo que se busca es una notación que abarque todo el ciclo de vida del desarrollo del software siendo estas su mayor desventaja para la selección. Sin embargo, UML permite especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos. Documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, etc.). Puede conectarse con lenguajes de programación (ingeniería directa e inversa), ayudando así al programador. Dicho lo anterior se selecciona UML como lenguaje de modelado a utilizar en el trabajo de diploma.

1.9 Herramientas de Modelado Visual

Se puede definir a las Herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora) como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software. Como se sabe, los estados en el Ciclo de Vida de desarrollo de un Software son: Investigación Preliminar, Análisis, Diseño, Implementación e Instalación. (26)

Entre las Herramientas CASE más usadas actualmente se encuentra el Visual Paradigm, Umbrello UML Modeller, BOUML, entre otras muchas más.

1.9.1 Visual Paradigm para UML

Visual Paradigm es empleada para visualizar y diseñar elementos de software, utiliza el lenguaje UML y proporciona a los desarrolladores diseñar, integrar y modelar visualmente los distintos diagramas que se generan a lo largo del desarrollo del software para obtener como resultado un producto con calidad.

A continuación se ofrece una lista de las características principales que posee esta herramienta:

- Entorno de creación de diagramas para UML 2.0.
- Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad en múltiples plataformas.
- Producto de calidad.
- Soporta aplicaciones Web.
- Varios idiomas.
- Fácil de instalar y actualizar.
- Compatibilidad entre ediciones. (19)

1.9.2 Rational Rose

Es una herramienta de software propietaria utilizada para el modelado visual mediante UML. Rational, es la herramienta CASE desarrollada por los creadores de UML (Booch, Rumbaugh y Jacobson), que cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases.

Características de Rational Rose:

- Permite especificar, analizar y diseñar el sistema antes de codificarlo.
- Permite el chequeo de la sintaxis UML.
- Genera la documentación de forma automática.
- Genera código a partir de los modelos.
- Mantiene la consistencia de los modelos del sistema de software.
- Permite realizar ingeniería inversa. (17)

1.9.3 Fundamentación de la herramienta seleccionada

Estas herramientas pueden ayudar en el proceso de modelación de diagramas que se generan durante la vida del software. Visual Paradigm en su versión 5.0 se integra con

diversos IDE's como: NetBeans (de Sun). Está disponible en varias ediciones: Enterprise, Professional, Community, Standard, Modeler y Personal. Genera código y realiza ingeniería inversa para diferentes lenguajes de programación como: Java, C++, CORBA IDL, PHP, XML Schema y ADA, lo que lo hace una herramienta que se adecua a las necesidades para realizar la implementación del sistema.

1.10 Frameworks de desarrollo

El RHODA y el CRODA son aplicaciones implementadas sobre tecnologías Web, que necesitan un fuerte modelo de datos, gestión de usuarios y otras prestaciones que hacen pensar en el uso del patrón de diseño Modelo-Vista-Controlador. Una de las mejores alternativas es el uso de un framework de desarrollo para la implementación de este tipo de aplicaciones. En su generalidad la mayoría de estos frameworks brindan las siguientes facilidades: (27)

- Mapeo Objeto-Relacional u ORM según sus siglas en inglés.
- Seguridad ante diferentes tipos de ataques como la inyección SQL.
- Gran diversidad de bibliotecas y utilidades para realizar tareas muchas veces engorrosas.
- Logran un Modelo-Vista-Controlador con una calidad aceptable.
- Muchas actividades y validaciones son posibles de implementar mediante archivos de configuración del Framework.
- El acceso a la aplicación es mediante una sola página servidora conocida como controlador frontal, por lo cual la aplicación tiene que estar accesible mediante la Web, lo que aporta un grado de seguridad mayor.

1.10.1 Symfony

En la actualidad existe una gran diversidad de frameworks en diferentes plataformas y lenguajes, tal es el caso de Spring para Java, Ruby On Rails para Ruby, siendo este último posiblemente el más famoso de todos. Para el caso de PHP existen varios como: CakePhp, Prado, ZendFramework, Symfony, entre otros. Symfony ha tomado las mejores ideas de Rails y de muchos otros frameworks, ha incorporado ideas propias y el resultado es un framework elegante, estable, productivo y muy bien documentado, en los últimos años Symfony ha tenido un gran auge y desarrollo, y en la UCI ha alcanzado gran aceptación por parte de los desarrolladores, por tener numerosas ventajas como:

- Es infinitamente escalable si se disponen de los recursos necesarios.

- Symfony sigue una política de tipo LTS (Long Term Support), por lo que las versiones estables se mantienen durante 3 años sin cambios, pero con una continua corrección de errores.
- Se publica bajo licencia MIT, con la que puedes desarrollar aplicaciones webs comerciales, gratuitas y/o de software libre.
- Symfony permite controlar hasta el último acceso a la información e incluye por defecto protección contra ataques XSS y CSRF.
- Desde su primera versión Symfony ha sido creado sólo para PHP 5, para obtener el máximo rendimiento de PHP y aprovechar todas sus características.
- Los creadores de Symfony no viven del framework, sino de las aplicaciones que desarrollan con él, por lo que les interesa tanto como a cualquiera, los aspectos como el rendimiento, la buena documentación y el soporte muy largo.
- Symfony es un framework está bien documentado, ya que sus creadores han publicado cinco libros gratuitos de calidad y siempre actualizados. Además, toda la documentación está traducida al español.
- Su código fuente incluye más de 9.000 pruebas unitarias y funcionales.
- Symfony está traducido a más de 40 idiomas e incluye todas las herramientas necesarias para traducir fácilmente las aplicaciones desarrolladas. (28)

1.10.2 Zend Framework

El Zend Framework (en lo adelante, ZF) es un framework de código abierto de desarrollo de aplicaciones web y servicios web para PHP. Te brinda soluciones para construir sitios web modernos, robustos y seguros.

Algunas características que hacen interesante el framework ZF:

- ZF es una implementación que usa código 100% orientado a objetos.
- Es muy desacoplado, por lo que mucha gente lo considera una biblioteca de componentes más que un framework. Por otra parte, gracias a esto, podemos utilizarlo en conjunto con otros frameworks.
- Es un framework de Open Source.
- Trabaja con PHP 5.
- ZF ofrece un gran rendimiento y una robusta implementación Modelo Vista Controlador (MVC).
- Completa documentación y tests de alta calidad.

1.10.3 Fundamentación del framework seleccionado

Por las características antes expuestas para cada uno de los frameworks, por las ventajas que aporta Symfony, se selecciona al mismo por ser uno de los frameworks más completos y bien integrados, además de tener una gran comunidad Open Source muy activa y profesional. Symfony integra algunas de las mejores ideas y herramientas del mundo PHP como una capa ORM a través del uso de Propel o Doctrine, YAML, arquitectura Modelo-Vista-Controlador. Otras de las razones por lo que se selecciona a Symfony como framework de desarrollo es porque el mismo es utilizado para la implementación de los proyectos CRODA y RHODA.

1.11 Sistemas Gestores de Bases de Datos

Un Sistema de Gestión de Bases de Datos (SGBD) se puede definir como un conjunto de datos interrelacionados y un conjunto de programas para acceder a esos datos, o sea, un conjunto coordinado de programas, procedimientos, lenguajes, que suministra tanto a los usuarios no informáticos como a los analistas, programadores o al administrador los medios necesarios para describir, recuperar y manipular los datos almacenados en la base, manteniendo su integridad, confidencialidad y seguridad. Entre los sistemas Gestores de Base de Datos más utilizados se tienen PostgreSQL, MySQL, Oracle, FireBird, SQLite, Apache Derby, Microsoft SQL Server 2000.

1.11.1 MySQL

MySQL es un sistema Gestor de Base de Datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.

- *Lenguaje SQL:* usa el lenguaje SQL estandarizado para el almacenamiento, actualización y acceso a información.
- *Lenguajes de programación:* Dispone de API's en gran cantidad de lenguajes como: C, C++, Java, PERL, y PHP solo por mencionar algunas.
- *Seguridad:* Posee opciones de protección mediante contraseña, la cual es flexible y segura. Además de permitir la gestión de usuarios y contraseñas, manteniendo un muy buen nivel de seguridad en los datos. (29)

Otras de las características de MySQL son:

- Aprovecha la potencia de sistemas multiprocesadores, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Es una de las herramientas más utilizadas por los programadores orientados a Internet.
- Fácil instalación y configuración. (30)

1.11.2 PostgreSQL

PostgreSQL es un servidor de base de datos relacional libre, bajo la licencia BSD. Fue desarrollado en la Universidad de California en el departamento de Informática de Berkeley, basado en el proyecto POSTGRES. Es pionero en los sistemas gestores de bases de datos disponibles actualmente a nivel mundial. (18)

- *Altamente extensible*: soporta operadores, funcionales métodos de acceso y tipos de datos definidos por el usuario.
- *Soporte SQL comprensivo*: cuenta con la más reciente especificación de SQL orientado a características avanzadas tales como las uniones (joins).
- *Integridad referencial*: presenta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.
- *API Flexible*: la flexibilidad del API de PostgreSQL ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente incluyendo diversos lenguajes.
- *Multiplataforma*: es funcional en los principales sistemas operativos como: Linux, Unix, Mac OS, Windows.

1.11.3 Fundamentación de la selección del SGBD

Por las características anteriormente expuestas, para el desarrollo del módulo de Revisión de la calidad de OA, el SGBD se selecciona a PostgreSQL porque es compatible para diversos sistemas operativos, como Linux, Solaris y Microsoft Windows. Actualmente el centro FORTES se encuentra en un proceso de migración al software libre, además de que el código fuente es manejable, y puede ser compilado, y ejecutado satisfactoriamente por cualquiera de estos sistemas sin ningún problema. Otra de las ventajas de utilizar este gestor es que es altamente adaptable a las

necesidades del cliente. Otras de las características de PostgreSQL es que soporta el protocolo de comunicación encriptado SSL. Además, de ser el SGBD utilizado en los proyectos CRHODA y RHODA.

1.12 Base de Datos Nativas en XML

Aunque no existe una definición estándar de una base de datos nativa en XML pero la organización XML: DB Initiative for XML Databases describe una base de datos de este tipo como un modelo lógico para documentos XML y almacena y recupera documentos de acuerdo con dicho modelo. (31)

1.12.1 eXist

Exist-db es un gestor de base de datos construido para sistemas abiertos utilizando la tecnología XML. Sirve para almacenar datos XML de acuerdo con el modelo de datos de la misma, basado en procesamiento de XQuery. (32)

Existen dos tipos de bases de datos XML que permiten el almacenamiento.

- *XML-Enabled*, que mapean documentos XML en una base de datos relacionales u objeto-relacional.
- *XNDB*, bases de datos nativas donde se utilizan como unidad funcional de almacenamiento estructuras específicas para documentos XML.

Dentro de sus principales características se encuentran:

- Soporte para lenguajes de consultas XML como XQuery, XPath y XSLT.
- Es posible indexar documentos.
- Soporte para actualización de datos y para multitud de protocolos como SOAP, XML-RPC, WebDav y REST.
- Es compatible con muchos estándares de la tecnología, por lo que se plantea ser una excelente base de datos para el desarrollo web.
- Ofrece un entorno eficaz para el desarrollo de aplicaciones web.
- Proporciona una interfaz de búsqueda agradable.
- Está bajo la licencia GNU LGPL.

1.12.2 Tamino XML Server

Esta base de datos nativa está centrada en interfaces nativos basados en estándares para SOA (Service Oriented Architecture), acceso más rápido a los datos, nuevos

conceptos de acceso y seguridad de los datos, así como soporte del protocolo Subversión para el versionado de documentos tanto XML como no XML.

Es el primer XML nativo del mercado, ofrece el archivado y procesamiento de mensajes y documentos, así como su uso en un SOA. Las aplicaciones centradas en datos se benefician también de optimizaciones en el procesamiento de consultas, en la tecnología de copia de seguridad ampliada y en conceptos de seguridad más flexibles.

1.12.3 Fundamentación de la selección de la Base de Datos Nativas en XML

Según el estudio realizado anteriormente, el principal factor que propició a la selección de eXist como Base de Datos Nativa para XML para la nueva solución al módulo de Revisiones de OA, que soporta varios lenguajes, es mucho más eficaz y flexible. Esta base de datos nativa proporciona un potente entorno para el desarrollo de aplicaciones web basadas en XQuery. Además se selecciona la misma, ya que es la usada en los proyectos clientes.

1.13 Entorno de Desarrollo Integrado

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes, a continuación se realizará el análisis de dos de ellos y la elección del que posee más ventajas para el desarrollo de la propuesta de solución en futuro.

1.13.1 NetBeans para PHP

NetBeans es un Entorno de Desarrollo Integrado (Integrated Development Environment, IDE) escrito en Java y de código abierto, aunque puede servir para casi cualquier otro lenguaje de programación. Este ofrece una versión para desarrollar en PHP que comprende una variedad de secuencias de comandos y lenguajes de marcado. El editor de PHP está dinámicamente integrado con HTML, JavaScript y CSS. Permite la integración de diferentes frameworks como Symfony y Zend, permitiendo filtrar y ayudar a ver los comandos de los mismos, incluyendo Doctrine o Propel. El editor de PHP entiende espacios de nombres y definiciones de las variables tipo de comentarios que mejora la finalización de código. Además de poseer características adicionales que incrementan su usabilidad:

- Fácil navegación de código.
- Cobertura de código.
- Unidad de Prueba de PHP.
- Depuración de PHP.
- Desarrollo de proyectos remotos y locales.
- Integración con MySQL.

NetBeans permite la gestión de proyectos, pues ofrece la posibilidad de agrupar proyectos, cuenta con plantillas de archivos, permite renombrar, copiar, mover, borrar archivos entre otras funciones. Además este IDE se puede instalar sencillamente y se ve de forma muy similar tanto en GNU/Linux como en Windows. (33)

1.13.2 Zend Studio

Zend Studio es un editor de texto para páginas PHP que proporciona un buen número de ayudas desde la creación y gestión de proyectos hasta la depuración del código.

Dentro de sus características se encuentra:

- No requiere la instalación previa de PHP ni del entorno de ejecución de Java.
- Soporte para PHP 4 y PHP 5.
- Resaltado de sintaxis, autocompletado de código, ayuda de código y lista de parámetros de funciones y métodos de clase.
- Detección de errores de sintaxis en tiempo real.
- Soporte para gestión de grandes proyectos de desarrollo.
- Manual de PHP integrado. (34)

1.13.3 Fundamentación del IDE de desarrollo

Por las características y ventajas expuestas anteriormente Netbeans por encima de Zend Studio se ha seleccionado primero, este IDE es un excelente depurador de errores, y que se integra con diferentes lenguajes como JavaScript y HTML con autocompletado, navegación por el código, resaltado de errores y gestión de tareas. Además de ser un IDE que puede ser utilizado en diferentes plataformas. Una de las desventajas que posee el IDE Zend Studio es que es privativo, dificultándose el uso del mismo en Cuba. Además, se utiliza NetBeans ya que el mismo se utiliza en los proyectos clientes CRODA y RHODA.

1.14 Conclusiones parciales

A partir del análisis y la revisión de la bibliografía de varias fuentes relacionada con los OA se define un nuevo concepto del mismo. Se realiza un estudio de los métodos de evaluación de la calidad de OA existentes en el mundo actualmente. Además, se realizó un estudio de las herramientas y tecnologías que permitirán el análisis y diseño del módulo; así como las propuestas de las que permitirán en el futuro la implementación del mismo. Se eligió como lenguajes de programación a PHP, y JavaScript por su potencial, además de consumir pocos recursos de la computadora, Visual Paradigm para el modelado de los artefactos necesarios en el flujo de análisis y diseño. Se propone a Symfony como framework de desarrollo web, además de sistema gestor de base de datos a PostgreSQL.

CAPÍTULO#2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

2.1 Introducción

En presente capítulo se da a conocer la propuesta de modelo de evaluación de la calidad de OA, se describe la propuesta de solución, se describe el modelo de dominio y los procesos del sistema que tienen que ver con el campo de acción, se enumeran los requisitos funcionales que tendrá la propuesta del sistema a desarrollar, se identifican y describen los actores, los casos de uso y las relaciones existentes entre ellos.

2.2 Propuesta del Modelo de evaluación de la calidad de OA

A partir de la necesidad de evaluar la calidad de los OA creados en FORTES, se presentan a continuación el modelo de evaluación a utilizar en el centro. El modelo consta de dos categorías que en su desglose facilitarán la evaluación OA teniendo en cuenta los modelos de evaluación estudiados.

Categoría Contenido: permite evaluar OA observando si está relacionado a los objetivos curriculares de acuerdo con el contexto donde será aplicado. En esta categoría se evaluarán los criterios que estén enfocados a los objetivos (correctamente formulado, factibilidad) y al contenido (información correcta, precisa, clara, estructuración de la materia, adecuación de los objetivos a las características de los usuarios, ortografía y gramática).

Categoría Técnica-Estética: permite evaluar al OA con respecto a los aspectos que estén asociados al diseño. Los aspectos a evaluar, se encuentran tamaño del objeto, nivel de organización de los archivos, compatibilidad con navegadores entre otros que se encuentren asociados a los metadatos. Una vez visualizados los objetos, estas valoraciones deberían ser incluidas o guardada en la base de datos, para posterior uso del OA u otra evaluación tener un conocimiento del mismo. Otros aspectos a evaluar están relacionados a las características de los OA, como por ejemplo, la pertenencia a un estándar o especificación como también el formato de los metadatos.

A continuación se desglosan para cada categoría los diferentes criterios que al final permitirán dar una evaluación a los OA.

Capítulo#2: Descripción de la solución propuesta

Categoría Contenido:

1. El OA debe estar vigente, sino perecerá en el tiempo.
2. La información que brinda el OA debe ser de una fuente confiable.
3. El OA debe contener referencias complementarias como justificación de su contenido.
4. Los medios usados por el OA deben de estar de acuerdo con los objetivos del mismo.
5. La información que posee el OA debe ser detallada, clara y precisa para cumplir con sus objetivos.
6. La presentación del contenido debe de ser de correcta y fácil comprensión.
7. La información del OA debe tener aplicadas las reglas de ortografía y gramática.
8. Las ideas deben ser presentadas de manera organizada y ordenada de acuerdo con su complejidad en el aprendizaje.
9. El OA debe incorporar funcionalidades y mecanismos que favorezcan el trabajo colaborativo.
10. El OA debe incorporar funcionalidades y mecanismos que propicien altos niveles de interactividad con el alumno.
11. El OA debe incorporar mecanismos, funcionalidades o espacios que propicien creatividad en el estudiante.
12. El OA debe incorporar mecanismos o debe estar diseñado para que motive a ser usado por el estudiante.
13. El OA debe poseer un único objetivo.
14. La estructura didáctica tenga un mínimo conjunto de ítems (elementos) o que coincidan con la plantilla definida por CRODA para la creación del OA.

Categoría Técnica-Estética:

15. Compatibilidad con cualquier LMS (Learning Management System).
16. Nivel de organización de la estructura de archivos y directorios.
17. La aparición de archivos ni directorios que no se utilicen.
18. Visualización de las imágenes, con respecto a nitidez, brillo entre otras características.
19. Formato de los vídeos en este caso que sean 'asf', 'avi', 'fla', 'flv', 'mov', 'mp4', 'mpc', 'mpeg', 'mpg', 'qt', 'ram', 'rm', 'rmi', 'rmvb' o 'swf'.
20. Visualización de los vídeos.
21. Formato de las imágenes sea 'bmp', 'gif', 'jpeg', 'jpg' o 'png'.

Capítulo#2: Descripción de la solución propuesta

22. Al navegar por la estructura didáctica se verifica que no existan enlaces rotos en el OA.
23. Visualización en diferentes navegadores como Mozilla Firefox, Internet Explorer, Opera, NetScape.
24. Completamiento de la información en los metadatos y acordes con lo que brinda el OA.
25. El OA ante una falla debe ser capaz de recuperarse al estado inicial del recorrido en el que se encontraba antes de ocurrir dicha falla.
26. El tamaño del OA, debe estar en un rango que permita su uso eficiente. Se pueden establecer estándares máximos puesto que los OA deben ser reutilizables en distintas plataformas y sistemas, por lo tanto, si el tamaño es menor se facilita su incorporación en aplicaciones Web.
27. El diseño del OA debe ser uniforme dentro del mismo.

2.2 Propuesta de solución

El presente trabajo como solución propone la elaboración de un cuestionario para cada una de las categorías explicadas en el modelo de evaluación de OA propuesto. El cuestionario será capaz de automatizar algunas preguntas, brindándole al especialista información sobre el OA que no sabrá por sí solo. El especialista da respuesta a las diferentes interrogantes relacionadas con cada una de las categorías, tendrá la posibilidad de dar una evaluación y opinión sobre el OA evaluado guardándose en la base de datos para que posteriormente el autor pueda conocer que calidad tiene su producto. Pues para que esto sea posible se propone en el presente trabajo de diploma la generación de los artefactos necesario para la implementación del Módulo de revisión de la calidad de OA de los proyectos CRODA y RHODA.

2.3.1 Propuesta del cuestionario para la Categoría Contenido

1. ¿La última fecha de actualización del OA preexiste a 5 años?
2. ¿La información que posee el OA procede de fuentes bibliográficas confiables?
3. ¿El OA contiene referencias complementarias de acuerdo con el nivel de enseñanza al que va dirigido?
4. ¿Los medios utilizados en el OA son afines con los objetivos para quienes va dirigido el mismo?
5. ¿La información del OA es detallada, clara y precisa?

Capítulo#2: Descripción de la solución propuesta

6. ¿La presentación del OA es de correcta y fácil comprensión para el usuario a quien va dirigido?
7. ¿El OA contiene faltas de ortografías y/o errores gramaticales?
8. ¿De acuerdo al nivel de complejidad del aprendizaje del contenido del OA las ideas están organizadas?
9. ¿El OA facilita el trabajo colaborativo con el estudiante?
10. ¿El OA es interactivo?
11. ¿El OA propicia la creatividad en el estudiante?
12. ¿El OA está diseñado para que el estudiante se motive a usar al mismo?
13. ¿El OA tiene un único objetivo?
14. ¿La estructura didáctica del OA, o la cantidad de ítems (elementos) coincide con la plantilla definida por el proyecto CRODA para la creación del OA?

2.3.2 Propuesta del cuestionario para la Categoría Técnica-Estética

15. ¿Es compatible con otros LMS?
16. ¿La organización de archivos y directorios coincide con la plantilla definida por los proyectos clientes?
17. ¿Existen archivos y directorios que no se utilicen?
18. ¿Las imágenes que contiene el OA poseen buena visualización?
19. ¿Los videos que contiene el OA coincide con los formatos 'asf', 'avi', 'fla', 'flv', 'mov', 'mp4', 'mpc', 'mpeg', 'mpg', 'qt', 'ram', 'rm', 'rmi', 'rmvb' o 'swf'?
20. ¿Los videos que aparecen en el OA poseen buena visualización?
21. ¿Las imágenes que contiene el OA coinciden con los formatos 'bmp', 'gif', 'jpeg', 'jpg' o 'png'?
22. ¿Existen enlaces rotos en la estructura didáctica del OA?
23. ¿El OA puede ser visualizado en cualquier navegador?
24. ¿Los metadatos están completos y la información de los mismos es acorde con el OA?
25. ¿El OA es capaz de recuperarse y volver al estado inicial del recorrido en el que se encontraba antes de ocurrir la falla?
26. ¿El tamaño del OA permite ser reutilizado en otras plataformas?
27. ¿El diseño del OA es uniforme?

2.4 Modelo de dominio

En el estudio realizado del problema planteado, no fue posible identificar procesos del negocio, ya que está altamente centrado en tecnologías informáticas, haciéndose difícil determinar los elementos más importantes que intervienen, así como el establecimiento de las reglas de funcionamiento. Por estas razones es que se propone realizar un modelo conceptual o modelo de dominio. (35)

Este modelo permitirá mostrar de manera visual los principales conceptos que se manejan, ayudando a los usuarios, desarrolladores e interesados; a utilizar un vocabulario común para poder entender el contexto en que se sitúa el sistema. Además, contribuirá a identificar personas, eventos, transacciones y objetos involucrados en el sistema.

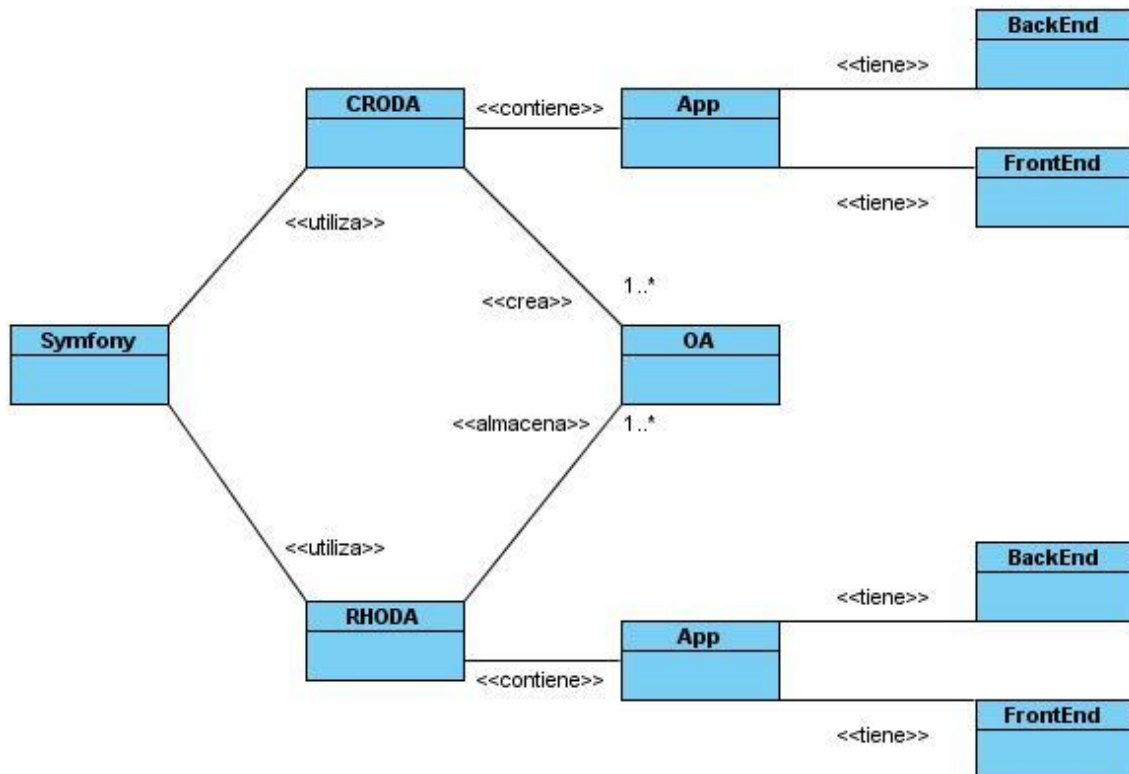


Ilustración 3: Modelo de dominio

2.4.1 Definición de las clases de dominio

App: Lugar donde se implementarán las aplicaciones con este framework.

Backend: la parte de administración de los proyectos clientes.

CRODA: Herramienta de autor para la creación de Objetos de Aprendizaje.

Frontend: La parte pública de los proyectos clientes.

OA: Objeto de Aprendizaje.

RHODA: Nombre del Repositorio de Objetos de Aprendizaje.

Symfony: Framework de desarrollo web que se aplica en los proyectos RHODA y CRODA.

2.5 Requisitos del sistema

Los requerimientos son condiciones o capacidades que el sistema tiene que tener para satisfacer algún documento formal. Estos describen a su vez todo lo que el sistema debe hacer o tener. Asimismo deben ser especificados por escrito como acuerdo de contrato. (35)

2.5.1 Requisitos Funcionales

Los requisitos funcionales pueden definirse como un atributo necesario dentro de un sistema, que puede representar una capacidad, una característica o un factor de calidad del sistema de tal manera que le sea útil a los clientes o a los usuarios finales. (35)

Permiten mediante una forma clara y detallada conocer todo lo que el sistema debe hacer. Estos se mantienen invariables sin importar con que propiedades o cualidades se relacionen.

RF1: Evaluar Categoría Contenido.

RF2: Evaluar la vigencia del OA.

RF3: Evaluar la confiabilidad de la fuente utilizada.

RF4: Evaluar las referencias complementarias que debe contener el OA.

RF5: Evaluar la estructura didáctica.

RF6: Evaluar Categoría Técnica_Estética

RF7: Evaluar la compatibilidad con otros LMS.

RF8: Evaluar nivel de organización de la estructura de archivos y directorios del OA.

RF9: Evaluar la aparición de archivos y directorios que no se utilicen.

RF10: Evaluar el formato de los vídeos.

RF11: Evaluar el formato de las imágenes.

RF12: Evaluar que estén completos los metadatos y acordes con el OA.

RF13: Evaluar el tamaño del OA.

2.5.2 Requisitos No Funcionales

Estos forman una parte significativa de la especificación. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto. Así si se conoce como el mismo cumple con todas las funcionalidades requeridas, las propiedades no funcionales, como cuán usable, seguro, conveniente y agradable, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación. (35)

Apariencia o interfaz externa

- Se debe tener en cuenta algunos elementos de diseño de cada uno de los proyectos clientes, RHODA y CRODA, como encabezamiento, estilos y formatos de texto.
- La aplicación debe ser diseñada con una interfaz amigable, procurando la fácil navegación del usuario, ajustándose de esta manera a los estándares establecidos para su realización.

Rendimiento

- La herramienta propuesta debe ser rápida, pero el tiempo de respuesta depende del tamaño y la carga del OA que se desea evaluar.

Confiabilidad

- Validar la captación de datos para evitar entradas inadecuadas.

Portabilidad

- El sistema puede ser accedido desde cualquier sistema operativo, debido a las características de las tecnologías propuestas para el desarrollo mismo. Además, es uno de los objetivos principales, que sea posible su utilización en cualquier entorno donde se necesite.

Seguridad y privacidad

- El sistema debe asegurar que los materiales y recursos sean vistos y analizados únicamente por las personas capacitadas y las establecidas previamente por el administrador de la aplicación, se hace necesario el uso de roles especificando los privilegios para cada uno en el sistema.
- La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes, de la misma forma será considerada igual a la fuente o autoridad de los datos.

Legales

- La plataforma escogida para el desarrollo de la aplicación está basada en la licencia GNU/GPL.

Capítulo#2: Descripción de la solución propuesta

Ayuda y documentación en línea

- Concebir un módulo que brinde ayuda de forma tal que en todo momento le permita al usuario orientarse respecto a las opciones que tiene el sistema, utilizando textos explicativos que describan la acción que se está realizando y también la ayuda por tópicos.

Software

El módulo debe correr en una estación de trabajo que conste con los programas que se exponen a continuación:

- Navegador Web (Internet Explorer, Mozilla Firefox, Opera, NetScape).
- Servidores de base de datos: PostgreSQL 8.4 o inferior.
- Base de datos XML: Exist 1.4.
- Apache 2.
- PHP 5.

Hardware mínimo para el servidor

- Microprocesador 2 GHz.
- 1 GB de memoria RAM.
- 80 GB de disco duro.

2.6 Identificación de los actores del sistema

Un actor es un usuario que intercambia información, este puede ser una persona, una máquina o un software que interactúe con el sistema. O sea, el actor es una entidad externa de la aplicación que interactúa con los casos de uso. (35)

En la siguiente tabla se muestra la interacción de los actores con el sistema:

Actor	Descripción
Revisor	Es la persona encargada de evaluar los OA. Es el de mayor participación y activa en este módulo, ya que la mayor cantidad de tareas están designadas a él.

Tabla 1: Actores del sistema.

2.7 Identificación de los casos de uso del sistema

Los casos de uso ofrecen información de cómo debe trabajar el sistema, son descripciones de la funcionalidad del sistema, describen bajo la forma de acciones y reacciones el comportamiento de un sistema desde el punto de vista del usuario. Estos deben hacer referencia al menos a un requerimiento, o sea, cada requerimiento debe quedar reflejado en un caso de uso y cualquier modificación que exista en algún requerimiento pueda afectar al caso de uso correspondiente, de la misma forma, si un caso de uso es modificado, se debe revisar esa modificación y ver qué requerimiento pueda estar afectado también, todo este control se puede llevar gracias a la trazabilidad que existe entre ambos elementos. (35)

Para cumplir con los requerimientos se definen los siguientes casos de uso:

CU#1: Evaluar Categoría Contenido.

CU#2: Evaluar vigencia del OA.

CU#3: Evaluar fuente bibliográfica.

CU#4: Evaluar referencias complementarias.

CU#5: Evaluar estructura didáctica.

CU#6: Evaluar Categoría Técnica_Estética.

CU#7: Evaluar compatibilidad LMS.

CU#8: Evaluar organización de archivos y directorios.

CU#9: Evaluar inutilización de archivos o directorios.

CU#10: Evaluar formato de los videos.

CU#11: Evaluar formato de las imágenes.

CU#12: Evaluar información de metadatos.

CU#13: Evaluar tamaño del OA.

2.8 Diagrama de casos de uso del sistema

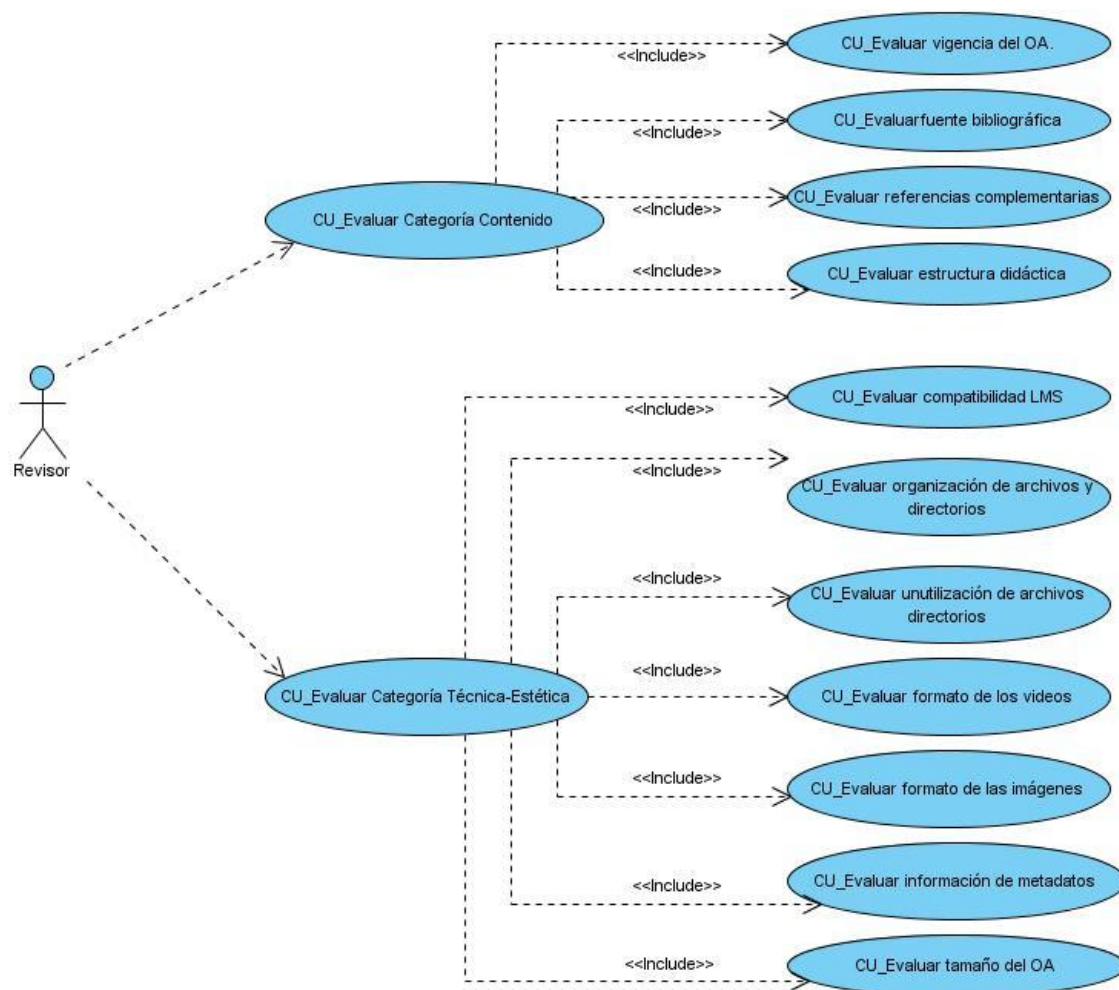


Ilustración 4: Diagrama de casos de uso del sistema.

2.9 Descripción de los casos de uso

A continuación se pondrán dos de las descripciones de los casos de uso del sistema, el resto se encuentran en el Anexo#2, para una mejor organización y legibilidad del documento.

Caso de uso: " Evaluar Categoría Contenido"	
CU# 1	Evaluar Categoría Contenido.
Propósito	Permitir evaluar el OA en la Categoría Contenido.
Actores	Revisor
Resumen	El caso de uso es iniciado por el revisor cuando selecciona realizar el cuestionario Evaluar Categoría Contenido el sistema

Capítulo#2: Descripción de la solución propuesta

	muestra el mismo y finaliza cuando el revisor culmina el cuestionario y lo envía.
Referencias	RF1
Precondiciones	El usuario debe estar autenticado y sea un revisor.
Poscondiciones	-
Curso normal de los eventos	
Acción del actor	Respuesta del sistema
	<ol style="list-style-type: none"> 1. El sistema muestra el cuestionario "Evaluar OA en Categoría Contenido" 2. El sistema llama a CU_ Evaluar vigencia del OA. 3. El sistema llama a CU_ Evaluar fuente bibliográfica. 4. El sistema llama a CU_ Evaluar referencias complementarias. 5. El sistema llama a CU_ Evaluar estructura didáctica.
<ol style="list-style-type: none"> 6. El revisor redacta una valoración de cada pregunta del cuestionario. 7. El revisor pulsa el botón Enviar, para guardar la evaluación e informe. 	<ol style="list-style-type: none"> 8. El sistema guarda informe en la base de datos.

Tabla 2: Descripción CU_Evaluar Categoría Contenido.

Caso de uso: " Evaluar Categoría Técnica_Estética"	
CU# 6	Evaluar Categoría Técnica_Estética.
Propósito	Permitir evaluar el OA en la Categoría Técnica_Estética.
Actores	Revisor

Capítulo#2: Descripción de la solución propuesta

Resumen	El caso de uso es iniciado por el revisor cuando selecciona realizar el cuestionario Evaluar Categoría Técnica_Estética y el sistema muestra el mismo y finaliza cuando el revisor culmina el cuestionario y lo envía.
Referencias	RF3
Precondiciones	El usuario debe estar autenticado y sea un revisor.
Poscondiciones	-
Curso normal de los eventos	
Acción del actor	Respuesta del sistema
	<ol style="list-style-type: none"> 1. El sistema muestra el cuestionario "Evaluar OA en Categoría Técnica_Estética". 2. El sistema llama a CU_ Evaluar compatibilidad LMS. 3. El sistema llama a CU_ Evaluar organización de archivos y directorios. 4. El sistema llama a CU_ Evaluar no uso de archivos ni directorios. 5. El sistema llama a CU_ Evaluar formato de los videos. 6. El sistema llama a CU_ Evaluar formato de las imágenes. 7. El sistema llama a CU_ Evaluar información de metadatos. 8. El sistema llama a CU_ Evaluar tamaño del OA.
<ol style="list-style-type: none"> 9. El revisor redacta una valoración de cada pregunta del cuestionario. 10. El revisor pulsa el botón Enviar, 	<ol style="list-style-type: none"> 11. El sistema guarda informe en la base de datos.

Capítulo#2: Descripción de la solución propuesta

para guardar la evaluación e informe.	
---------------------------------------	--

Tabla 3: Descripción CU_Evaluar Categoría Técnica_Estética

2.10 Conclusiones parciales

Se elaboró el modelo de evaluación de la calidad de los OA. La creación de un cuestionario que recogiera los criterios a evaluar en dicho modelo. Además, se identificaron los requisitos funcionales y no funcionales dando paso a un entendimiento eficiente para el desarrollo del módulo. A través de los requisitos se realizó la selección de los casos de uso que darían paso a las funcionalidades al sistema con la descripción textual de cada uno de ellos el cual permitirá la comprensión y la solución más exacta a la hora de implementar el módulo.

CAPÍTULO#3: ANÁLISIS Y DISEÑO DEL SISTEMA.

3.1 Introducción

En este capítulo presenta todos los artefactos y procesos realizados en los flujos de trabajo análisis y diseño de la metodología RUP, en el Módulo de Revisión de la calidad de OA para los proyectos CRODA y RHODA.

3.2 Análisis

El proceso de análisis consiste en obtener una visión del sistema que permita entender qué hay que hacer a la hora de implementar el módulo, interesado solamente por los requisitos funcionales. Se identifican las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellas, necesarias para la construcción del modelo de análisis.

3.2.1 Diagrama de Clases del Análisis

En este epígrafe se definen las siguientes clases del análisis:

- *Clase Entidad*: Modelan información que posee larga vida y que es a menudo persistente.
- *Clase Controladora*: Coordinan la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.
- *Clase Interfaz*: Modelan la interacción entre el sistema y sus actores. (35)

A continuación se muestran los diagramas de clases del análisis para cada uno de los casos de uso del sistema del Módulo de Revisión de la calidad de OA:

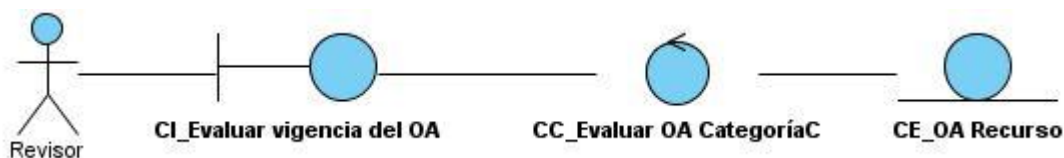


Ilustración 5: Diagrama de clases del análisis: Evaluar vigencia del OA.

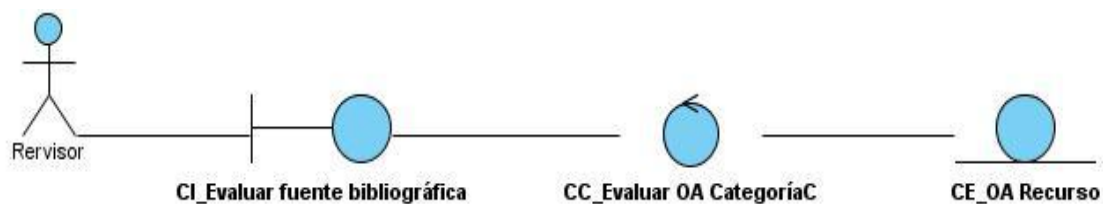


Ilustración 6: Diagrama de clases del análisis: Evaluar fuente bibliográfica.

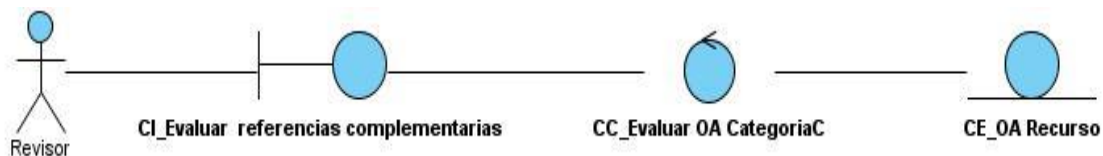


Ilustración 7: Diagrama de clases del análisis: Evaluar referencias complementarias.



Ilustración 8: Diagrama de clases del análisis: Evaluar estructura didáctica.



Ilustración 9: Diagrama de clases del análisis: Evaluar compatibilidad LMS.



Ilustración 10: Diagrama de clases del análisis: Evaluar organización de archivos y directorios.



Ilustración 11: Diagrama de clases del análisis: Evaluar inutilización de archivos y directorios.

Capítulo#3: Análisis y diseño del sistema

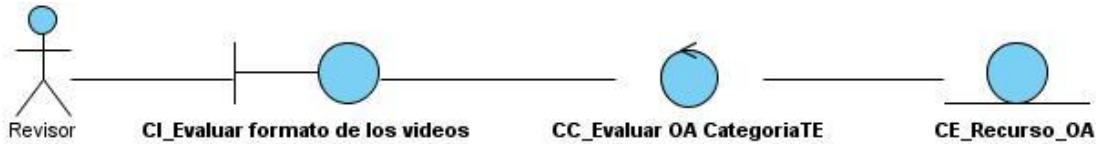


Ilustración 12: Diagrama de clases del análisis: Evaluar formato de los videos.



Ilustración 13: Diagrama de clases del análisis: Evaluar formato de las imágenes.



Ilustración 14: Diagrama de clases del análisis: Evaluar información metadatos.



Ilustración 15: Diagrama de clases del análisis: Evaluar tamaño del OA.

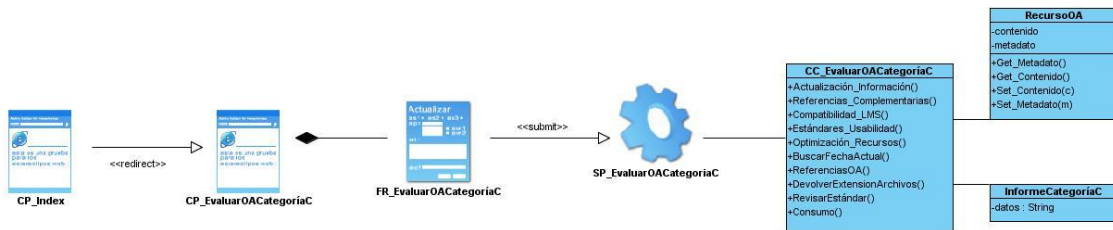


Ilustración 16: Diagrama de clases del análisis: Evaluar Categoría Contenido.

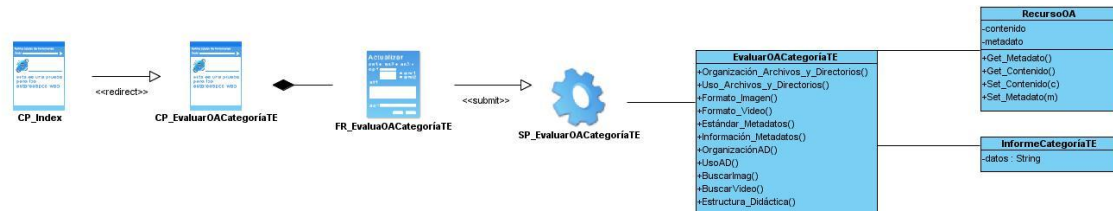


Ilustración 17: Diagrama de clases del análisis: Evaluar Categoría Técnica-Estética.

3.2.2 Diagramas de interacción (colaboración)

Se presentaran algunos de los diagramas de interacción de los casos de DC_Evaluar Categoría Contenido y DC_Evaluar Categoría Técnica-Estética. En el Anexo#1 se podrán encontrar los restantes diagramas de interacción.

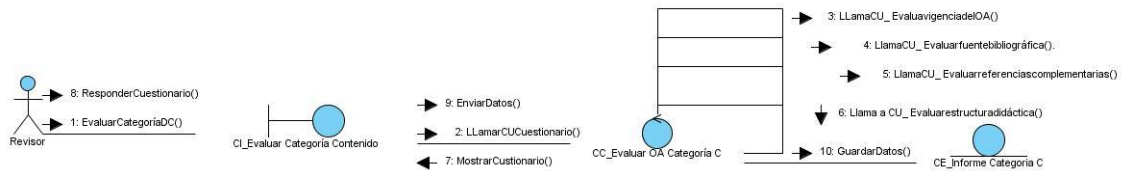


Ilustración 18: Diagrama de interacción: DC_Evaluar Categoría Contenido.

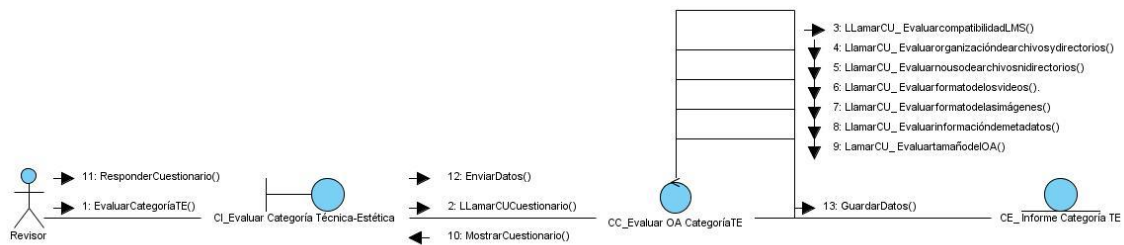


Ilustración 19: Diagrama de interacción: DC_Evaluar Categoría Técnica-Estética.

3.3 Diseño

El diseño es el centro de atención al final de la fase de elaboración y el comienzo de las iteraciones de construcción. El modelo de diseño está muy cercano al de implementación, lo que es natural para guardar y mantener el modelo de diseño a través del ciclo de vida completo del software. En el diseño se modela el sistema y determina su forma para que soporte todos los requerimientos, incluyendo los no funcionales. Una entrada esencial en el diseño es el resultado del análisis, o sea, el modelo de análisis, que proporciona una comprensión detallada de los requerimientos. Los propósitos del diseño se definen en transformar los requerimientos en un diseño de cómo el sistema debe ser, desarrollar una robusta arquitectura del sistema y adaptar el diseño para que se corresponda con el entorno de implementación, diseñando sus funcionalidades.

3.3.1 Diagrama de clases del diseño

En el diagrama de clases de diseño se muestran los atributos y métodos de cada clase y se representa de una forma sencilla la colaboración y las responsabilidades de las distintas clases que forman el sistema. Los diagramas de clases se utilizan para modelar la vista de diseño estática de un sistema, esto incluye modelar el vocabulario del sistema, modelar las colaboraciones o modelar esquemas. Los diagramas de clases son importantes no sólo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas ejecutables, aplicando ingeniería directa e inversa.

Para una mejor organización en el documento los diagramas de clases del diseño se recogen en el Anexo#3.

3.3 Modelo de datos

Un modelo de datos es un modelo abstracto que describe como deben ser representados los datos, sus relaciones y restricciones que deben cumplirse entre ellos; sirve para describir la estructura de la base de datos. La figura 21 muestra el modelo de datos de la propuesta de solución. (35)

Ilustración 20: Modelo de datos

3.4 Conclusiones parciales

En el presente capítulo se muestran algunos de los artefactos generados en el flujo de trabajo de análisis y diseño, además se incluye una breve descripción de los mismos, con el objetivo de que se comprenda perfectamente los requisitos del software, posibilitando la correcta transformación de los mismos a un diseño que indique como debe ser implementado el software posteriormente.

CONCLUSIONES

Mediante la investigación realizada en el presente trabajo, que contiene el estudio de los sistemas de evaluación de la calidad de OA existentes en el mundo, las herramientas y tecnologías a utilizar, además de temas relacionados con la investigación, este proceso fue de vital importancia para la solución que se propone, se obtuvieron resultados que permitieron arribar a las siguientes conclusiones:

- La solución propuesta permitió cumplir con el objetivo planteado, quedando listo todo para ser implementado el módulo Revisión de la calidad de los OA en los proyectos CRODA y RHODA.
- Se realizó un nuevo modelo de evaluación de la calidad de los OA basado en el análisis de algunos modelos y herramientas existentes en el mundo teniendo en cuenta algunas de sus características y ventajas.
- En el desarrollo del trabajo se realizó el levantamiento de los requerimientos funcionales determinados, definiendo así los casos de uso que responderían a cada funcionalidad del sistema, posibilitando sintetizar cada funcionalidad con el caso de uso que le daría cumplimiento.
- Se analizaron los casos de uso propuestos, se desarrolló los diagramas de clases del análisis, viabilizando el flujo de interacción entre clases el análisis.
- Se diseñaron los diagramas de clases para el módulo a desarrollar, así como una propuesta del modelo de datos que sustentara las clases persistentes del sistema.

RECOMENDACIONES

- Realizar los restantes flujos de trabajo que propone RUP como la guía del proceso de desarrollo hasta obtener el producto funcional en dichas aplicaciones.
- Utilizar la documentación generada en los flujos posteriores, dígase la implementación del sistema propuesto.
- Perfeccionar el modelado del módulo mediante la actualización de los cambios que sean necesarios durante el flujo de implementación y pruebas.
- Usar la documentación obtenida para ampliar las fronteras y realizar productos semejantes no sólo para el centro FORTES, sino también para todas las Universidades del país que desarrollen OA como producto.
- La utilización de otras propuestas de guías y modelos de evaluación de la calidad de OA para una mejor eficiencia en el proceso de revisión de los mismos.

TRABAJOS CITADOS

1. Rojas, Victor Fabian Novoa. **E-learning oportunidad y conocimiento.** [En línea] 2010. [Citado el: 25 de Noviembre de 2010.] <http://www.scribd.com/doc/32241380/E-Learning-and-Moodle>.
2. Gusmán, Clara López. Los Repositorios de Objetos de Aprendizaje como soporte a un entorno e-learning. *Tesis doctoral*. España : s.n., 2005.
3. Franco, Rodolfo. SISTEMAS DE INFORMACIÓN GEOGRÁFICA. [En línea] 02 de 08 de 2010. [Citado el: 20 de diciembre de 2010.] <http://gemini.udistrital.edu.co/comunidad/profesores/rfranco/metadatos.htm>.
4. Naharro, Susana Martínez, y otros. Los objetos de aprendizaje como recurso de calidad para la docencia: criterios de validación de objetos en la Universidad Politécnica de Valencia. España : Instituto de Ciencias de la Educación, Área de Sistemas de Información y Comunicaciones, 2009.
5. Morales, Erla, y otros. GESTIÓN DE OBJETOS DE APRENDIZAJE DE CALIDAD: CASO DE ESTUDIO. 2009.
6. Vidal, Christian L., Segura, Alejandra A. y Prieto2, Manuel E. Calidad en objetos de aprendizaje. Chile; España : s.n., 2007.
7. Otamendi, Ainhoa, y otros. Instrumento para la evaluación de objetos de aprendizaje (LORI_ESP). *Manual de usuario*. 2006.
8. Tamayo, Daymy Ávila. Elementos fundamentales sobre la concepción de los Objetos de Aprendizaje en la UCI. *Elementos fundamentales sobre la concepción de los Objetos de Aprendizaje en la UCI*. La Habana : s.n., 2009.
9. Mateos, Jorge E. Gil. “Estrategia de Gestión de Recursos Educativos Abiertos en forma de Objetos de Aprendizaje en la Universidad de La Habana”. Universidad de La Habana. La Habana, Cuba : s.n., 2010. Tesis Doctoral.
10. Warden, Shore and. *The Art of Agile Development* . s.l. : O’Reilly, 2007. nt Primera Edición.
11. Visual Paradigm. [En línea] UML CASE Tools - Free for Learning UML, Cost-Effective for Business Solutions, 20 de Diciembre de 2010. [Citado el: 12 de Enero de 2011.] <http://www.visual-paradigm.com/product/vpuml/>.
12. Hipertext Preprocessor. [En línea] 2008. [Citado el: 18 de Diciembre de 2010.] <http://www.php.net>.
13. ASP en castellano. [En línea] 2007. [Citado el: 18 de Diciembre de 2010.] <http://www.programacion.net/asp>.

14. González, Sergio Navarro. Grupo de Investigación en Procesamiento del Lenguaje Natural y Sistemas de Información. [En línea] 28 de Octubre de 2009. [Citado el: 18 de Diciembre de 2010.] http://gplsi.dlsi.ua.es/~slujan/asp/Introduccion_a_ASP.htm.
15. WebStyle. *JavaScript*. [En línea] 8 de Agosto de 2006. [Citado el: 25 de Enero de 2011.] <http://www.webestilo.com/javascript/>.
16. JAVASCRIPT. [En línea] 26 de Abril de 2006. [Citado el: 21 de Enero de 2011.] http://perso.wanadoo.es/javascript_12/.
17. Los diferentes lenguajes de programación para la web. *Maestros de la Web*. [En línea] 7 de Noviembre de 2007. [Citado el: 25 de Enero de 2011.] <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>.
18. Larman, Craig. *UML y Patrones*. 2003.
19. Herramienta CASE para el modelado UML. [En línea] 2008. [Citado el: 05 de Enero de 2011.] <http://www.visual-paradigm.com/product/vpuml/>.
20. Zaninotto, Francois y Potencier, Fabien. *Symfony, la guía definitiva*. 2005.
21. MySQL. 1.4. *Panorámica del sistema de gestión de base de datos MySQL*. [En línea] 2011. [Citado el: 25 de Enero de 2011.] <http://dev.mysql.com/doc/refman/5.0/es/what-is.html>.
22. PostgreSQL vs. MySQL. *MySQL*. [En línea] 2009. [Citado el: 17 de Diciembre de 2010.] http://danielpecos.com/docs/mysql_postgres/x57.html.
23. PostgreSQL. *PostgreSQL vs. MySQL*. [En línea] 2009. [Citado el: 17 de Diciembre de 2010.] http://danielpecos.com/docs/mysql_postgres/x15.html.
24. NetBeans. [En línea] 2011. [Citado el: 2011 de 1 de 24.] http://netbeans.org/features/php/index_es.html.
25. Amador, César Eduardo Velázquez, Arteaga, Jaime Muñoz y Rodríguez, Francisco Alvarez. Aspectos de la Calidad de Objetos de Aprendizaje en el Metadato de LOM. México : s.n., 2007.
26. Morales, Erla, y otros. Propuesta de Evaluación de Objetos de Aprendizaje. Salamanca España : s.n., 2006.
27. Damián, Verónica Patricia Molina y Martínez, Efrén Aguilar. SLOR: Simple Learning Object Repository, diseñando un repositorio de objetos de aprendizaje para la plataforma de teleeducación EducAD-UNACH. 2007.
28. Sánchez-Alonso, Salvador, Ovelar, Ramón y Sicilia, Miguel Ángel. Libro de Buenas Prácticas de e-learning. 2010.

29. Rational Unified Process(RUP). *Rational Unified Process(RUP)*. [En línea] 2009. [Citado el: 10 de Diciembre de 2010.] <http://www.utim.edu.mx/~mgarcia/DOCUMENTO/ADSI2/RUP.pdf>.
30. Blog Thaira.net. *Symfony, Beneficios e información*. [En línea] 2011. [Citado el: 20 de Febrero de 2011.] <http://www.thaira.net/blog/desarrolloweb/symfony-beneficios-e-informacion/>.
31. Bagüés, Ramiro Lago. Calidad Proactiva. *Framework Spring*. [En línea] Enero de 2008. [Citado el: 23 de Febrero de 2011.] <http://www.proactiva-calidad.com/java/spring/introduccionSpring.html>.
32. Zend Studio . *The Leading PHP IDE*. [En línea] 2010. [Citado el: 20 de Febrero de 2011.] <http://www.zend.com/en/resources/zend-documentation/>.
33. O, Dra. Lourdes Galeana de la. Objetos de Aprendizaje. *Objetos de Aprendizaje*. Colima : s.n., 2009.
34. XML. *XML*. [En línea] 2010. [Citado el: 22 de Diciembre de 2010.] xmldb-org.sourceforge.net/.
35. eXist-db Open Source Native XML Database. *eXist-db Open Source Native XML Database*. [En línea] 2011. <http://exist.sourceforge.net/> .
36. Zend Studio The leading PHP IDE. *Zend Studio The leading PHP IDE*. [En línea] 2010. [Citado el: 11 de Enero de 2011.] <http://www.zend.com/en/products/studio/>.
37. Roger Pressman. Ingeniería de Software. *Ingeniería de Software*. [En línea] 5ta Ed.
38. Belén, Marcos Emanuel. *Búsquedas Semánticas en Repositorios de Objetos de Aprendizaje*.
39. Gonzalez, Leonardo Rodríguez. Análisis y diseño de la Versión 3.0 de RHODA. Universidad de la Ciencias Informática : s.n., 2011.
40. Christian L. Vidal, Alejandra Segura, Manuel Prieto. Calidad en Objetos de Aprendizaje. 2008.

BIBLIOGRAFÍA

- (1) Garcerant, I. (10 de julio de 2008). Modelo de dominio. Recuperado el 2 de enero de 2011, de Tecnología y Synergix: <http://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>
- (2) ERS:especificacion de requerimientos del software. (5 de octubre de 2010). Recuperado el 2 de enero de 2011, de LUZ ARISPE: <http://luzarizpe.wordpress.com/2010/10/05/ers-especificacion-de-requerimientos-del-software/>
- (3) Quintin, S. C. (s.f.). Modelo del análisis. Recuperado el 4 de enero de 2011, de Scribd: <http://www.scribd.com/doc/7930106/Modelo-de-Analisis-Saul-Cuzcano-Quintin>
- (4) Gómez, A. V. (julio de 2006). Diagrama de clases del análisis. Recuperado el 5 de enero de 2011, de monografías.com: <http://www.monografias.com/trabajos37/diagnostico-pedagogico/diagnostico-pedagogico2.shtml>
- (5) Weitzenfeld, A. (2011). Ingeniería de software orientada a objetos con UML, Java e Internet. 4 de enero: 2005.
- (6) Los diferentes lenguajes de programación para la web. Maestros de la Web (7 de Noviembre de 2007 <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>).
- (7) Rational Unified Process(RUP). Rational Unified Process(RUP). 2009. <http://www.utim.edu.mx/~mgarcia/DOCUMENTO/ADSI2/RUP.pdf>.
- (8) Mateos, Jorge E. Gil. "Estrategia de Gestión de Recursos Educativos Abiertos en forma de Objetos de Aprendizaje en la Universidad de La Habana". Universidad de La Habana. La Habana, Cuba, 2010.
- (9) AulaGlobal. (2005). "Objetos de Aprendizaje - Ventajas de los Objetos de Aprendizaje.", <http://www.aulaglobal.net/ve/observatorio/articles.php?lng=es&pg=174>
- (10) Castro, C. V. (2008) Calidad en Objetos de Aprendizaje, <http://alarcos.inf-cr.uclm.es/doc/cmsi/trabajos/Christian%20Vidal%20%20-%20Calidad%20en%20OA%20-%20Pres.pdf>
- (11) González, R. E. R., J. M. Arteaga, et al. (2007) Evaluación de Objetos de Aprendizaje a través del Aseguramiento de Competencias Educativas. pp: 1-17, <http://ihm.ccadet.unam.mx/virtualeduca2007/pdf/211-RRG.pdf>
- (12) Mauri, T., J. Onrubia, et al. (2005). La calidad de los contenidos educativos reutilizables: diseño, usabilidad y prácticas de uso RED. Revista de Educación a

- Distancia (España). Número monográfico II, <http://www.um.es/ead/red/M2/> ,
http://www.um.es/ead/red/M2/mauri_onrubia.pdf
- (13)** Morales, E., F. J. García, et al. (2005) Propuesta de Evaluación de Objetos de Aprendizaje. Pp: 9, <http://www.uoc.edu/symposia/spdece05/pdf/ID06.pdf> , Universidad Salamanca, España
- (14)** Velázquez, C. E., J. M. Arteaga, et al. (2007) Aspectos de la Calidad de Objetos de Aprendizaje en el Metadato de LOM. Pp:18, <http://ihm.ccadet.unam.mx/virtualeduca2007/pdf/214-CVA.pdf>
- (15)** Ruiz, R., J. Muñoz, et al. (2008). "Formato para la Determinación de la Calidad en los Objetos de Aprendizaje.", http://www.laclo.org/index.php?option=com_docman&task=doc_view&gid=6
- (16)** Toll Palma, Yuniel Del Carmen, et al. (2011). " Guía de evaluación de la calidad de los Objetos de Aprendizaje producidos en la Universidad de las Ciencias Informáticas".
- (17)** Ávila, D. T. (2009). "Elementos fundamentales sobre la concepción de los Objetos de Aprendizaje en la UCI". Sesión Científica de la Dirección de Teleformación. Laboratorio de Teleformación, septiembre 2009.

GLOSARIO DE TÉRMINOS

API: Application Programming Interface, llevado al español, Interfaz de Programación de Aplicaciones. Es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usados generalmente en las bibliotecas.

Calidad: Totalidad de propiedades y características de un producto, proceso o servicio que le confiere su aptitud para satisfacer unas necesidades expresadas o implícitas.

CakePhp: Framework de desarrollo en PHP.

E-learning: medios electrónicos utilizados para elevar el nivel educativo y la capacitación de las comunidades de usuarios a nivel mundial.

Evaluación: permite señalar, estimar, apreciar o calcular el valor de algo.

Estándar: sirve como tipo, modelo, norma, patrón o referencia.

Framework: Es una estructura conceptual y tecnológica de soporte definida, normalmente, con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado.

HEODAR: Herramienta de Evaluación de Objetos De Aprendizaje Reutilizables.

Herramienta CASE: Ingeniería de Software Asistida por Ordenador. Varias aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero.

Hyperlinks: Son los elementos que asignan a un trozo de texto la facultad de que se pueda conectar a otro elemento, página u objeto.

IMS: Instructional Management System es un estándar para la definición de metadatos de objetos de aprendizaje.

LOM: Metadatos para los Objetos de Aprendizaje usado para describir un objeto de aprendizaje y otros recursos digitales similares usados para el apoyo al proceso de enseñanza-aprendizaje.

LTS: Es la abreviatura de “Long Term Support”, soporte a largo plazo en español.

Plugin: Es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

ROA: Repositorio de Objetos de Aprendizaje.

RUP: Es un proceso en el que se han unificado técnicas de desarrollo y metodologías, está orientado a objetos y preparado para desarrollar grandes y complejos proyectos.

SOA: Arquitectura Orientada a Servicios, traducido de sus siglas en inglés.

SQL: Lenguaje de consulta estructurado.

Taxonomía de Bloom: propone desarrollar un sistema de clasificación teniendo en cuenta tres aspectos: el cognitivo, el afectivo y el psicomotor.

TICs: Tecnologías de la Información y las Comunicaciones.

UML: Es un lenguaje de modelado para especificar o describir métodos o procesos.

XQuery: Es un lenguaje de consulta diseñado para consultar colecciones de datos

XML. Es semánticamente similar a SQL, pero incluye algunas capacidades de programación.

ANEXOS

Anexo#1: Diagramas de interacción.

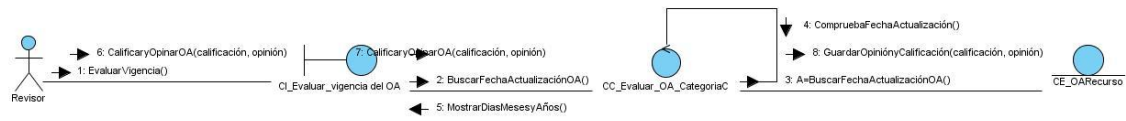


Ilustración 21: Diagramas de interacción: DC_Evaluar vigencia del OA.

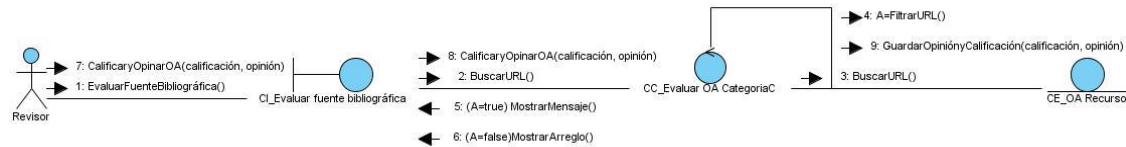


Ilustración 22: Diagramas de interacción: DC_Evaluar fuente bibliográfica.

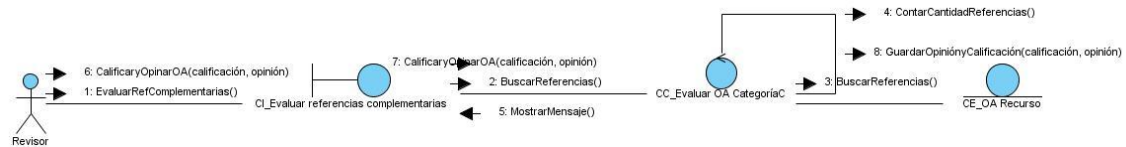


Ilustración 23: Diagramas de interacción: DC_Evaluar Referencias complementarias.

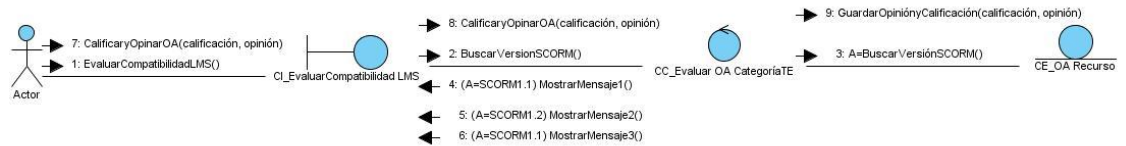


Ilustración 24: Diagramas de interacción: DC_Evaluar Compatibilidad LMS.



Ilustración 25: Diagramas de interacción: DC_Evaluar organización de archivos y directorios.

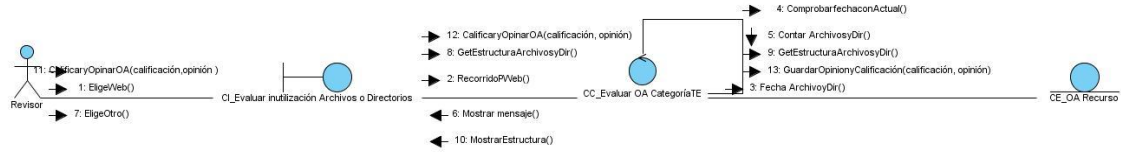


Ilustración 26: Diagramas de interacción: DC_Evaluar inutilización de archivos o directorios.

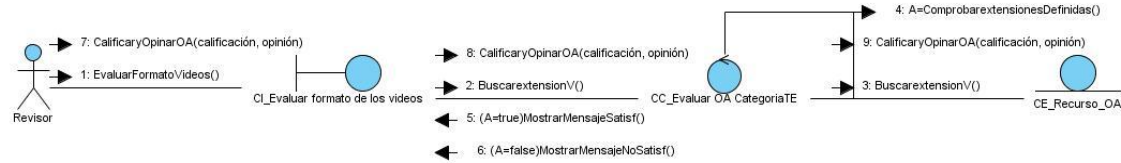


Ilustración 27: Diagramas de interacción: DC_Evaluar formato de los videos.

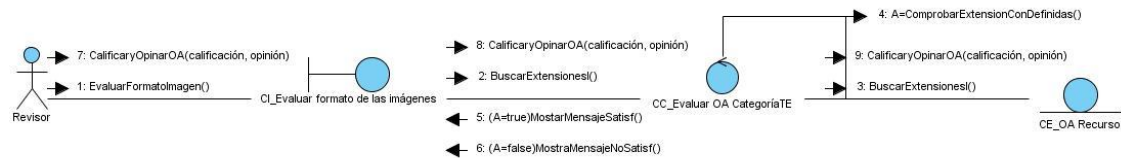


Ilustración 28: Diagramas de interacción: DC_Evaluar formato de las imágenes.



Ilustración 29: Diagramas de interacción: DC_Evaluar información de metadatos.

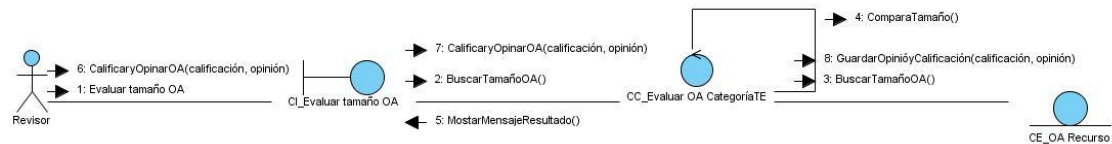


Ilustración 30: Diagramas de interacción: DC_Evaluar tamaño del OA.

Anexo#2: Descripción de los casos de uso.

Caso de uso: " Evaluar vigencia del OA"	
CU# 2	Evaluar vigencia del OA.
Propósito	Permitir evaluar el OA de acuerdo a la última vez en que fue actualizado.
Actores	Revisor
Resumen	El caso de uso es iniciado por el sistema cuando verifica la última fecha de actualización que tuvo el OA y finaliza cuando le muestra al usuario dicha fecha.
Referencias	RF2
Precondiciones	El usuario debe estar autenticado y sea un revisor.
Poscondiciones	-
Curso normal de los eventos	
Acción del actor	Respuesta del sistema
	1. El sistema muestra cuestionario "Evaluar OA en Categoría Contenido"
	2. El sistema busca en los metadatos del OA la última fecha en que fueron actualizados los archivos del mismo y devuelve la misma.
	3. El sistema comprueba que el año de la fecha devuelta se encuentre en un rango de 5 años de vigencia.
	4. El sistema devuelve como mensaje "El OA está actualizado hace \$ años, \$ meses y \$ días."
5. El usuario evalúa y comenta el OA.	6. El sistema guarda la evaluación y el comentario en los metadatos.

Curso Alterno "Rango de vigencia"	
	6.1 Devuelve como mensaje "El OA no está actualizado."
Volver paso 5.	

Tabla 4: Descripción del CU_Evaluar vigencia del OA.

Caso de uso: " Evaluar fuentes bibliográficas"	
CU# 3	Evaluar fuentes bibliográficas.
Propósito	Permitir evaluar el OA de acuerdo a la confiabilidad de las fuentes bibliográficas que se usa el mismo.
Actores	Revisor
Resumen	El caso de uso es iniciado por el sistema cuando verifica mediante un filtro de direcciones URL, aquellas definidas como fuentes no confiables y finaliza cuando el sistema muestra que está correcto o las fuentes no confiables que contiene el OA.
Referencias	RF4
Precondiciones	El usuario debe estar autenticado y sea un revisor.
Poscondiciones	-
Curso normal de los eventos	
Acción del actor	Respuesta del sistema
	1. El sistema muestra cuestionario "Evaluar OA en Categoría Contenido".
	2. El sistema busca en los metadatos del OA las fuentes

	<p>bibliográficas.</p> <p>3. El sistema compara las direcciones encontradas con las definidas en el filtro de direcciones URL como no confiables.</p> <p>4. El sistema muestra el mensaje "La fuente bibliográfica es confiable."</p>
5. El usuario da una evaluación y opina del OA.	6. El sistema guarda la evaluación y el comentario en los metadatos.
Curso Alterno "Fuente no confiable"	
	6.1 El sistema devuelve un arreglo con las direcciones de las fuentes no confiables.
Volver a paso 5.	

Tabla 5: Descripción del CU_Evaluar fuentes bibliográficas.

Caso de uso: " Evaluar referencias complementarias "	
CU# 4	Evaluar referencias complementarias.
Propósito	Permitir evaluar el OA de acuerdo a la cantidad de referencias complementarias que contenga según a quien va dirigido.
Actores	Revisor
Resumen	El caso de uso es iniciado por el sistema cuando verifica que la cantidad de referencias sea mayor al número definido por la plataforma para el nivel de enseñanza a quien vaya dirigido.
Referencias	RF4
Precondiciones	El usuario debe estar autenticado y sea un revisor.
Poscondiciones	-

Curso normal de los eventos	
Acción del actor	Respuesta del sistema
	1. El sistema muestra cuestionario "Evaluar OA en Categoría Contenido".
	2. El sistema comprueba que la cantidad de referencias complementarias que contenga el OA sea igual o mayor a la definida por la plataforma. 3. El sistema muestra el siguiente mensaje en un cuadro de texto "El OA posee \$ referencias para el nivel de enseñanza al que va dirigido."
4. El usuario evalúa y opina sobre el OA.	5. El sistema guarda la evaluación y comentario en los metadatos.

Tabla 6: Descripción del CU_Evaluar referencias complementarias.

Caso de uso: " Evaluar estructura didáctica "	
CU# 5	Evaluar estructura didáctica.
Propósito	Permitir evaluar el OA con respecto a que la estructura didáctica tenga un mínimo conjunto de <i>ítems</i> o que coincidan con una plantilla definida por el proyecto.
Actores	Revisor
Resumen	El caso de uso es iniciado por el sistema cuando verifica que el OA tenga un mínimo de ítem en la estructura didáctica o coincidan con una plantilla definida por la plataforma y finaliza cuando da respuesta al usuario.

Referencias	RF5
Precondiciones	El usuario debe estar autenticado y sea un revisor.
Poscondiciones	-
Curso normal de los eventos	
Acción del actor	Respuesta del sistema
	1. El sistema muestra cuestionario "Evaluar OA en Categoría Contenido."
	2. El sistema comprueba los <i>ítems</i> del OA o la estructura didáctica coincida con la plantilla definida por la plataforma.
	3. El sistema muestra el siguiente mensaje "Los ítems o la estructura didáctica del OA coinciden correctamente con la plantilla definida por la plataforma"
4. El usuario evalúa y comenta el OA.	5. El sistema guarda la evaluación y comentario en los metadatos.
Curso Alterno "Plantilla no coincide"	
	5.1 El sistema muestra el siguiente mensaje "Los ítems o la estructura didáctica del OA no coinciden con la plantilla definida por la plataforma"
Volver a paso 5.	

Tabla 7: Descripción del CU_Evaluar estructura didáctica.

Caso de uso: " Evaluar compatibilidad LMS "

CU# 7	Evaluar compatibilidad LMS.
Propósito	Permitir evaluar el OA de acuerdo a la compatibilidad que tiene con otros LMS usando empaquetamiento SCORM.
Actores	Revisor
Resumen	El caso de uso es iniciado por el sistema cuando verifica con que versión de SCORM está empaquetado el OA y que compatibilidad brinda a cualquier LMS y finaliza cuando informa con que versión está empaquetado el OA.
Referencias	RF7
Precondiciones	El usuario debe estar autenticado y sea un revisor.
Poscondiciones	-
Curso normal de los eventos	
Acción del actor	Respuesta del sistema
	1. El sistema muestra cuestionario "Evaluar OA en Categoría Técnico-Estético."
	2. El sistema busca en que versión está empaquetado el OA devolviendo el resultado. 3. El sistema mostrará el siguiente mensaje "El OA está empaquetado en la versión de SCORM 1.1"
4. El usuario evalúa y opina sobre el OA.	5. El sistema guarda la evaluación y el comentario en los metadatos.
Curso Alterno "Versión del OA"	
	5.1 El sistema muestra el siguiente

	<p>mensaje " El OA está empaquetado en la versión de SCORM 1.2"</p> <p>5.2 El sistema muestra el siguiente mensaje "El está empaquetado en la versión de SCORM 2004 ".</p>
<p>Volver a paso 4.</p>	

Tabla 8: Descripción del CU_Evaluar compatibilidad LMS.

<p>Caso de uso: " Evaluar organización de archivos y directorios"</p>	
<p>CU# 8</p>	<p>Evaluar organización de archivos y directorios.</p>
<p>Propósito</p>	<p>Permitir evaluar el OA de acuerdo al nivel de organización de los archivos y directorios rigiéndose por una plantilla definida por la plataforma.</p>
<p>Actores</p>	<p>Revisor</p>
<p>Resumen</p>	<p>El caso de uso es iniciado por el sistema cuando verifica la organización de la estructura de los archivos y directorios, y que la misma coincida con la plantilla definida por la plataforma y finaliza cuando muestra la información al usuario.</p>
<p>Referencias</p>	<p>RF8</p>
<p>Precondiciones</p>	<p>El usuario debe estar autenticado y sea un revisor.</p>
<p>Poscondiciones</p>	<p>-</p>
<p>Curso normal de los eventos</p>	
<p>Acción del actor</p>	<p>Respuesta del sistema</p>
	<p>1. El sistema muestra cuestionario "Evaluar OA en Categoría</p>

	Técnico-Estético.”
	<p>2. El sistema comprueba que los archivos y directorios del OA coincidan en organización con una plantilla definida por la plataforma.</p> <p>3. El sistema muestra el siguiente mensaje “Los archivos y directorios están organizados por la plantilla modelo”.</p>
4. El usuario evalúa y opina sobre el OA.	5. El sistema guarda la evaluación y el comentario en los metadatos.
Curso Alterno “Organización no coincide”	
	4.1 El sistema muestra el siguiente mensaje “La organización de los archivos y directorios no coinciden con la plantilla modelo”.

Tabla 9: Descripción del CU_Evaluar organización de archivos y directorios.

Caso de uso: “ Evaluar inutilización de archivos o directorios”	
CU# 9	Evaluar inutilización de archivos o directorios.
Propósito	Permitir evaluar el OA teniendo en cuenta si es una página web, si aparecen archivos y directorios que no se utilicen en el OA.
Actores	Revisor
Resumen	El caso de uso es iniciado por el sistema cuando verifica si los archivos y directorios del OA son utilizados cuando se ejecuta el mismo y finaliza el caso de uso dando respuesta al usuario.

Referencias	RF9
Precondiciones	El usuario debe estar autenticado y sea un revisor.
Poscondiciones	-
Curso normal de los eventos	
Acción del actor	Respuesta del sistema
	1. El sistema muestra cuestionario "Evaluar OA en Categoría Técnico-Estético."
2. El usuario elige la opción "Página web".	3. El sistema inicia un recorrido rápido del OA recorriendo la página web completa. 4. El sistema verifica que la fecha de modificación de los archivos y directorios del OA sea la actual. 5. El sistema cuenta los archivos y los directorios que están no se usan. 6. El sistema muestra el siguiente mensaje " El OA contiene \$ archivos y \$ directorios sin usar"
7. El usuario comenta y califica el OA.	8. El sistema guarda la evaluación y comentario sobre el OA en los metadatos.
Curso Alterno "Otro tipo"	
9. El usuario elige la opción " Otro".	10. El sistema muestra los archivos y directorios del OA.
Volver paso 7.	

Tabla 10: Descripción del CU_ Evaluar inutilización de archivos o directorios.