Universidad de las Ciencias Informáticas

Facultad 5

Título: Módulo para la evaluación de la calidad de los objetos de aprendizaje producidos en la Universidad de las Ciencias Informáticas.

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

Autores: Susana Enamorado Estrada
Yannier González Márquez

Tutores: MSc. Yuniet del Carmen Toll Palma.
Ing. Yohandri Ril Gil.

Ciudad de la Habana. 29 de mayo de 2012

"Año 54 de la Revolución"

Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la UCI los derechos patrimoniales de la misma, con carácter exclusivo.

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

Agradecimientos

Susana Enamorado Estrada

"Gracias a mi familia, en especial a mi mamá, mi hermana mi tía y mi prima Graciela y Dania. Valoro mucho todo lo que han hecho por mí y les agradezco aún más. Soy afortunada por la familia que tengo. A Eddy y su familia por ser excepcionales y maravillosos. A Yans porque con sus acciones se ha ganado mi cariño y admiración y hoy lo considero un hermano. Agradezco mucho a mis tutores y compañero de tesis Yuniet, Ril y Yannier porque han sido como una familia para mi. Por confiarme este proyecto y ayudarme con su realización. A mis viejas, nuevas amistades y compañeros por todas las experiencias que hemos compartido. A mis profesores, a todos aquellos que alguna ves me ayudaron y a la universidad por la oportunidad. Mucha gracias a todos."

Yannier González Márquez

"Agradezco especialmente a mi madre por creer en mí y por ser mi guía. A mi papá por siempre estar ahí cuando lo necesité. A mi otro padre Alexis, por apoyarme en todo. A mi tío Pedri y en general al resto de mi familia. A mis hermanos Marcos, Yailín y Yamisleidis por ser la alegría de mi vida. A mi compañera de tesis y a mis tutores. Agradezco a todas las personas que conocí antes y durante estos cinco años y que de una forma u otra me ayudaron y me apoyaron en especial a mis amigos de siempre: Eddy, Manuel, Peña, Eduardo, Delisle, Eilan entre otros muchos que no puedo mencionar pero que igual son muy importantes para mí. A Maren y a todas esas personas que me ayudaron a ver la universidad desde otro punto de vista. ¡En fin gracias a todos!"

Dedicatoria

Susana Enamorado Estrada

"Este trabajo va dedicado a las personas más especiales de mi vida. A mi mamá Yolanda por traerme al mundo y una vez en él luchar por mi incansablemente. Por enseñarme a ser disciplinada, perseverante y por su gran ejemplo. Por ser tu cuerpo mi refugio y tu voz mi aliento. A mi hermana Karen por ser la mejor hermana del mundo y para mi un regalo único y hermoso. Por todo su amor, sus consejos, su confianza y por lo bien que la pasamos cuando estamos juntas. A mi sobrino Adrián David, gracias bebé por la alegría que trajiste a esta pequeña familia. Espero que te conviertas en un gran hombre, te quiero y te deseo lo mejor del mundo. A mi novio Eddy, porque en todo lo que he logrado en estos últimos años estás tú. Por tanto apoyo, amor, dedicación y por querer siempre dibujarme en el rostro una sonrisa. A todos ustedes mi amor infinito, los quiero mucho."

Yannier González Márquez

"A mi mamá, por ser mi amiga, mi confidente, la persona que siempre está ahí para mí, por guiarme y por creer en mí aun en los momentos en que ni yo mismo lo hice. A mis abuelos que aunque la vida no les alcanzó para ver el concreto de mis sacrificios se que siempre están conmigo."

Resumen

En la presente investigación se realiza una amplia explicación de los conceptos asociados a los Objetos de Aprendizaje (OA). Se tratan elementos como su estructura, almacenamiento, estándares, evaluación manual y automatizada, haciendo especial énfasis en esta última. Actualmente el repositorio RHODA de la Universidad de las Ciencias Informáticas (UCI) cuenta con una variedad de OA que sirven como apoyo a la docencia. En él dichos OA deben pasar por un proceso de evaluación de la calidad. Para ello se cuenta con un grupo de revisores preparados para cumplir adecuadamente con esta actividad. Sin embargo las evaluaciones que se obtienen son manuales y proclives a la intromisión inconsciente de errores por parte de los revisores. En aras de contribuir a la mitigación de este problema el objetivo fundamental de esta investigación es implementar un módulo para automatizar el proceso de evaluación de la calidad de los OA. Para lograr dicho objetivo se realizó un estudio y selección de metodologías y herramientas que permitieran llevar un proceso de desarrollo con calidad y sencillez. Por ello se hizo uso del framework Symfony y de Rational Unified Process (RUP) para guiar el proceso de desarrollo. Los artefactos generados y las posibilidades que brinda el Symfony, hicieron posible obtener una aplicación web para el perfeccionamiento del proceso de evaluación de la calidad de los OA.

PALABRAS CLAVES: OA, Evaluación de la calidad, Repositorio, RHODA.

Índice general

Índice de figuras

Índice de tablas

Introducción

Las Tecnologías de la Información y las Comunicaciones (TIC) se denominan como “conjunto de tecnologías que permiten la adquisición, producción, almacenamiento, tratamiento, comunicación, registro y presentación de informaciones, en forma de voz, imágenes y datos contenidos en señales de naturaleza acústica, óptica o electromagnética”[1].

Las TIC y su introducción generalizada, intervienen en muchas esferas de la sociedad actual como: la medicina, el deporte, la educación, entre otras. La educación en cuestión, exhibe el proceso de enseñanza en sus inicios, empleando recursos físicos como maquetas, objetos, fotos, láminas, entre otros. Más adelante, con el surgimiento de las TIC, la educación obtuvo un notable impulso, y se estimuló la creación y uso de recursos didácticos digitales como son los Objeto de Aprendizaje (OA).

Existen diversas definiciones de OA, en términos generales este concepto se refiere a todo tipo de recurso digital que se emplee para apoyar la formación educativa, por ejemplo, archivos de audio, video, fotos y multimedias [2, 3, 4, 5, 6, 8].

Con la intención de gestionar estos medios electrónicos para la educación, surgen los Repositorios de Objetos de Aprendizaje (ROA) en los cuales se gestiona, almacena y publica una variedad de OA que pueden ser utilizados en cursos presenciales y no presenciales, aportando gran apoyo al profesor e investigador para contribuir a la educación y la capacitación de usuarios a nivel mundial. En estos repositorios cada OA debe pasar un proceso de evaluación de la calidad, el cual determina si este cumple los requerimientos para ser publicado a los usuarios.

Para que un OA obtenga una evaluación de calidad satisfactoria, es necesario que desde su creación se tengan en cuenta indicadores de evaluación que permitan garantizar la veracidad y la distribución equilibrada de los contenidos, calidad de imágenes, nivel de organización de los archivos y directorios, entre otros. Los OA deben garantizar el desarrollo de conocimientos, habilidades y capacidades en los estudiantes [7, 8].

Internacionalmente, los revisores evalúan la calidad de los OA teniendo en cuenta instrumentos de eva-

luación. Estos instrumentos ofrecen una guía de los aspectos a tener en cuenta para evaluar el OA así como los posibles valores que estos puedan alcanzar. Para el logro de un proceso de evaluación más robusto y fiable, resulta útil incorporar herramientas que sirvan para automatizar el proceso de revisión.

Cuba no se encuentra ajena a estos nuevos paradigmas los cuáles han revolucionado el proceso de enseñanza aprendizaje. La producción y uso de OA es amplio y en ascenso. En este sentido la tendencia actual es tratar los cursos basados en las TIC como OA. Involucradas en esta novedosa temática se encuentran reconocidas instituciones como el Instituto Superior Politécnico José Antonio Echeverría y la Universidad de las Ciencias Informáticas (UCI).

La UCI juega un papel protagónico en este sentido. Actualmente se encuentra incursionando en la creación de OA para utilizarlos como apoyo a la docencia, enfatizando en el aseguramiento de la calidad de los mismos. Para realizar la revisión de los OA, se aplica la Guía de evaluación de la calidad de los Objeto de Aprendizaje. Esta guía creada en la propia institución, tiene como objetivo guiar a los revisores en el proceso de evaluación, para garantizar la calidad de los OA antes de ser publicados. Además contribuye a facilitar el proceso evaluativo y permite evaluar todos los aspectos que se desean con rigurosidad, exactitud, de manera ordenada, facilitando la obtención de resultados satisfactorios y conformes[8].

Los mayores problemas dentro de la universidad para la evaluación de los OA, están dados cuando no se dispone de un grupo de revisores con conocimientos homogéneos en todas las aristas que comprende la evaluación, que conjuntamente con una herramienta que sirva de apoyo a este proceso puedan evaluar y determinar el resultado final generado. Esto, debido a que la evaluación que se obtiene es manual y no esta exenta de errores introducidos por la mano del hombre. Generalmente los resultados obtenidos son producto de la apreciación de los evaluadores, esto trae consecuencias negativas entre las que se destacan:

- Las evaluaciones están expuestas a la subjetividad en algunos indicadores que evalúen los especialistas de calidad.
- No se tiene una visión real de la calidad que acompaña a los OA que se ponen a disposición de la comunidad universitaria.
- Riesgo de publicar OA con bajos niveles de calidad.
- Divergencia entre los resultados obtenidos y los esperados durante la utilización de los OA como

apoyo a la docencia.

Debido a que los revisores acceden al repositorio donde se encuentran almacenados los OA y obtienen de cada uno de ellos una evaluación manual de la calidad. Es necesario automatizar su función mediante la guía, supervisión y registro de la labor que realizan. Teniendo en cuenta lo anteriormente expuesto, surge la necesidad de encontrar una solución para el siguiente **problema científico**: Las insuficiencias en el proceso de evaluación de los OA producidos en la UCI afecta la calidad de los mismos.

Por lo que el **objeto de estudio** de la investigación será: la herramientas para automatizar el proceso de evaluación de los OA. Donde el **campo de acción** está relacionado con la implementación de un módulo de revisión automatizada para evaluar la calidad de los OA en la UCI.

El **objetivo general** de esta investigación es implementar un módulo para automatizar el proceso de evaluación de la calidad de los OA en la UCI. El cual conllevó a definir como **objetivos específicos** los siguientes:

- Elaborar el marco teórico a partir de la investigación de los principales conceptos, características, estructura, clasificación y estado del arte de las herramientas de evaluación de la calidad de los OA.
- Seleccionar la metodología y herramientas de desarrollo de las que se hará uso.
- Diseñar el módulo de evaluación de la calidad de los OA.
- Implementar el módulo de evaluación de la calidad de los OA.
- Validar mediante pruebas de software el módulo de evaluación de la calidad de los OA.
- Integrar el módulo de evaluación de la calidad de los OA a RHODA.

Se defiende la siguiente **idea**: la implementación de un módulo para automatizar el proceso de evaluación de la calidad de los OA en la UCI, permitirá agilizar el proceso de evaluación y disminuir los errores introducidos durante este proceso.

Para darle cumplimiento a los objetivos anteriormente expuestos se definen una serie de **tareas de investigación**, las mismas son las siguientes:

1. Determinación de las principales características de los OA.
2. Determinación de la metodología y herramientas de desarrollo de las que se hará uso.
3. Determinación de los estándares de codificación de Symfony como framework de desarrollo.
4. Análisis y diseño del módulo de revisión para la evaluación de la calidad de los OA.
5. Implementación del módulo de revisión para la evaluación de la calidad de los OA.
6. Análisis de los resultados obtenidos de la aplicación del módulo de revisión para la evaluación de la calidad de los OA.
7. Validación mediante pruebas de software del módulo de revisión para la evaluación de la calidad de los OA.
8. Integración módulo de revisión para la evaluación de la calidad de los OA con el repositorio de OA de la UCI.

Entre los **métodos de investigación científica** existentes fueron utilizados los siguientes:

Métodos teóricos:

Análisis y síntesis principalmente en la precisión de los fundamentos teóricos relacionados con la calidad de los OA. Para apoyar el proceso de enseñanza aprendizaje a través de herramientas empleadas para automatizar el proceso de evaluación y el análisis e interpretación de los resultados obtenidos con la aplicación del módulo de revisión para evaluar la calidad de los mismos.

Inducción y deducción permitió analizar el fenómeno de la calidad de los OA, el proceso de evaluación del mismo y la influencia que tiene en otros contextos para arribar a conclusiones de acuerdo con lo investigado.

Métodos empíricos:

Observación se utilizó para conocer como se evalúa la calidad de los OA en la UCI.

Entrevista individual para enriquecer y completar la información obtenida. Además para conocer si el proceso de revisión se realiza de manera manual o automatizada.

Revisión de documentos con la finalidad de recopilar información de los OA, de la calidad y la manera en que se procede para la evaluación de los mismos, si es forma manual o automatizada.

Métodos del nivel estadístico:

Estadística descriptiva para el procesamiento de los datos que se obtengan en la aplicación del módulo de revisión para la evaluación de la calidad de los OA.

Estadística inferencial para la toma de decisiones sobre el rechazo o no de los OA sometidos a evaluación a partir de los datos que se obtengan de la evaluación de calidad realizada.

Estructuración del contenido y breve explicación de sus partes.

Capítulo 1. Fundamentación teórica. Se expone el marco conceptual de la propuesta de solución, introduciendo conceptos e ideas que se manejarán a lo largo de la investigación. Comprende el estudio del estado del arte de las herramientas que automatizan el proceso de evaluación de la calidad de los OA. Además de esto, se lleva a cabo un estudio de las herramientas, metodologías y lenguajes de desarrollo, para posteriormente realizar la selección de aquellas que se considere más conveniente su utilización en el desarrollo de la solución de la propuesta.

Capítulo 2. Características del sistema. Comprende la descripción de la solución propuesta. En un primer momento se enuncian las características del sistema para luego modelar el dominio del problema. Más adelante se realiza el modelado del sistema incorporando a la investigación elementos como requisitos funcionales, no funcionales y casos de uso. Se realiza el diseño del sistema, exponiendo estilos arquitectónicos y patrones de diseño empleados, diagramas de clases y el modelo de datos etc. En cada punto anteriormente mencionado se generan los artefactos correspondientes, según propone la metodología de desarrollo de software seleccionada en el capítulo anterior.

Capítulo 3. Implementación y pruebas del sistema: Comprende la implementación del sistema y se refleja mediante el diagrama de despliegue, la disposición física de los nodos vitales para el mismo con los componentes que radican en cada uno de ellos. Se valida la solución propuesta mediante pruebas de software las cuales constituyen una garantía de la calidad del mismo. Finalmente se evalúa la solución haciendo un resumen de los principales resultados obtenidos.

Capítulo 1

Fundamentación teórica

Introducción

En este capítulo se desarrollan cuatro puntos fundamentales, primeramente la descripción de los principales conceptos teóricos vinculados con el desarrollo de la investigación, para facilitar el entendimiento a partir de distintos enfoques, definiciones y puntos de vista, y emitir, en la medida de lo posible, la que más se adecue a las particularidades de este trabajo. O sea, una vez conformada esta primera parte se pretende elevar el dominio de elementos tan importantes como: ¿Qué son los OA?, ¿Cómo se estructuran?, ¿Cómo y dónde se almacenan?, ¿Bajo qué estándares es posible la creación y manipulación de OA? y no menos importante ¿Cómo es el proceso de evaluación de la calidad de los OA?.

En un segundo momento y como parte del estudio del proceso de evaluación de la calidad de los OA, se impone una revisión del estado del arte, donde se hará referencia a las principales prácticas que se han creado con la intención de facilitar el trabajo de los revisores y hacerlo más fiable y asequible. En este punto, con el apoyo de los resultados obtenidos en el estudio del estado del arte, es importante resaltar la necesidad que tiene la UCI de contar con un sistema que permita apoyar el proceso de evaluación de la calidad de los OA que se realiza en la misma.

En un tercer momento prevalece la realización de un cuidadoso estudio de las herramientas, metodologías y lenguajes para determinar cuáles se utilizarán en el desarrollo de la solución propuesta. Las implicaciones de hacer la mejor selección, la que más se adecue a las necesidades, se advierten tanto en la robustez del proceso de desarrollo del software como el producto final que se desea obtener. Para finalizar se ofrecen las conclusiones a las que se han podido arribar, condicionadas por el estudio previo realizado a lo largo del capítulo.

1.1. Objeto de Aprendizaje

Muchos autores han definido los OA desde diferentes ópticas, según sus puntos de vista, circunstancias y adoptando posiciones particulares. La presente investigación se acoge fundamentalmente a la dada por Leonardo Rodríguez, la cual define a los OA como “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 pueda ser reutilizado en entornos e-learning”[10].

No obstante, no es objetivo restringir este debatido concepto a una definición, sino que se persigue lograr un entendimiento más amplio, que permita conformar juicios propios y acertados, considerando una variedad de perspectivas que tal vez, sean mejor asimiladas ya sea porque posean mayor nivel de sencillez, generalidad, especificidad, etc.

Para el logro de este objetivo, se hace referencia a las siguientes definiciones de OA, una de ellas, desarrollada por la Corporación Universitaria para el Desarrollo de Internet (CUDI) quien describe, “Un OA es una entidad informativa digital que se corresponde (representa) con un objeto real, creada para la generación de conocimientos, habilidades, actitudes y valores, y que cobra sentido en función de las necesidades del sujeto que lo usa”[9].

La otra se expone con ayuda de las palabras de la Dra. Chan “La definición de OA más difundida hasta ahora, y al mismo tiempo, por su sencillez, más discutida y usada como base de nociones más elaboradas, es aquella que lo plantea como “cualquier recurso digital que puede ser reusado como soporte para el aprendizaje”.(Wiley 2000)” [2]. Seguidamente puntualiza que “El uso de términos como “cualquier” y “recurso” dejan abierta la definición, lo cual Wiley lo considera una cualidad importante, dado que permite considerar como recurso cosas de tamaño y función muy diversas” [2].

El objetivo fundamental que la UCI persigue con la utilización de los OA, es potenciar la calidad del proceso de enseñanza aprendizaje, ya que estos promueven en el estudiante el desarrollo de determinados componentes como la independencia y la flexibilidad. Constituyen un significativo soporte si se desea mejorar la apropiación de los conocimientos por los estudiantes.

1.1.1. Estructura de los Objeto de Aprendizaje

La estructura de los OA ha evolucionado gracias al análisis que la comunidad académica ha realizado con respecto al tema. Se han ofrecido muchas teorías y buenas prácticas manejadas por diferentes instituciones. En sentido general, con solo hacer un recorrido elemental por algunas de ellas, es posible apreciar que existen numerosas estructuras de OA, todas de valor y que expresan visiones distintas del mismo problema. A continuación algunos enfoques que evidencian esta afirmación.

Atendiendo a las conclusiones a las que los autores Moral y Cernea han arribado en su estudio sobre OA, se obtuvo lo siguiente:

”Teniendo en cuenta la importancia de presentar al alumno una información estructurada y esquematizada y basándonos en las contribuciones enunciadas, nos referimos a un OA como a un contenido organizado en introducción, módulos teóricos que a su vez tienen un subobjetivo, actividades y evaluación que pueden contener recursos como texto, audio, video, JavaScript, Flash, simulaciones, estudio de caso, etc.. Su estructura será flexible, cada uno de los módulos que lo componen siendo independiente a su vez y con potencial de reutilización en otros OA y adaptabilidad”[11].

Figura 1.1: La información estructurada que se presenta en un OA[11]

Otros autores han considerado en su trabajo una estructura que incluye los siguientes apartados [25]:

- Visión general: Este apartado incluye la introducción, justificación, importancia, objetivos, prerrequisitos, esquema, resumen y relaciones con otros materiales. Todos estos elementos contribuyen a conformar una representación general del OA en cuestión.
- Evaluación: Se definen las pautas de evaluación, según los objetivos y habilidades que se considere de importancia su dominio. Incluye pruebas de evaluación o autoevaluación cuando sea necesario.
- Contenido: Se hace la presentación de la información, donde generalmente se incluyen conceptos, datos, procesos, procedimientos, principios, entre otros.

- Actividad: Su objetivo es la aplicación de los conocimientos adquiridos en la interacción con el OA, para lo cual se definen ejercicios y/o tareas.

Para Ruiz y Muñoz los OA:

“Están estructurados por componentes, ellos son: los contenidos temáticos, que debe estar bien distribuido, claros y certeros para que el estudiante comprenda la información, el diseño estético, el objeto debe tener una organización tanto en los contenidos mostrados como en la interfaz, el diseño instruccional, es el orden secuencial que debe tener el contenido en los objetos y metadato estandarizado, el mismo debe estar bien descrito y regido por estándares permitiendo una búsqueda rápida y efectiva en los repositorios donde se encuentre” [12].

Examinada la óptica de diferentes autores, se aprecia que los OA se estructuran en dependencia de lo que cada institución productora de OA quiera lograr. Dichas instituciones definen la estructura más conveniente en cada caso, lo que no impide que de forma general estas estructuras estén estrechamente relacionadas.

Debido a que el principal objetivo de esta investigación es desarrollar un módulo de revisión para apoyar la evaluación de los OA en la UCI. Se asume la estructura que la UCI como institución productora de OA ha creado. Esta estructura está conformada por los objetos de información, los metadatos y la estructura didáctica (objetivos, contenidos y autoevaluación/reflexión sobre lo aprendido) [13].

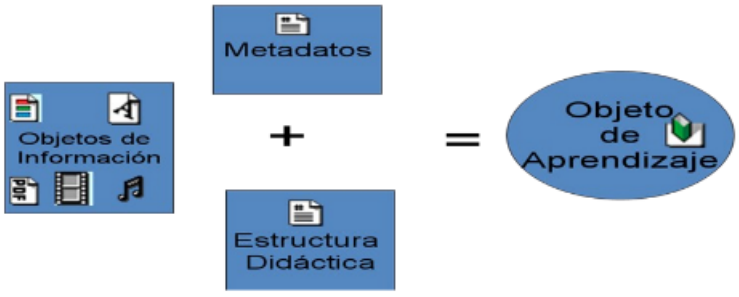

Figura 1.2: Componentes del OA utilizado en la UCI [14]

1.1.2. Almacenamiento de los Objeto de Aprendizaje

Con el aumento de los OA, se hizo necesario el almacenamiento de estos en un depósito que los mantenga centralizados para garantizar su homogeneidad, supervisión y gestión en sentido general, dando lugar a la aparición de los ROA.

Los ROA se han convertido en sistemas de amplia utilización y difusión, frecuentemente se presentan mediante una interfaz web, un mecanismo de búsqueda y determinados criterios para su clasificación. Para lograr tener un idea más clara “Los sistemas de repositorios son la infraestructura clave para el desarrollo, almacenamiento, administración, localización y recuperación de todo tipo de contenido digital” [16]. Más enfocados en el contexto educativo el JORUM+ project adopta la siguiente definición “un ROA es una colección de OA que tienen información (metadatos) detallada que es accesible vía Internet. Además de alojar los OA los ROA pueden almacenar las ubicaciones de aquellos objetos almacenados en otros sitios, tanto en línea como en ubicaciones locales”[17]. Aunque existen muchas definiciones para los ROA, estas, en sentido general no difieren mucho entre sí y dejan ver claramente que estos repositorios, son bases de datos o catálogos, creados para ser utilizados en un proceso de enseñanza.

A partir de las investigaciones realizadas, sería oportuno resaltar aquellas funcionalidades que se consideran más relevantes [15, 18, 16]:

- Administrar OA: Los repositorios de OA deben tener un cierto nivel de administración, en el que deben incluir los mecanismos apropiados para realizar funciones de manejo de derechos digitales, seguridad de acceso y manejo del ciclo de vida de los OA.
- Almacenar OA: Colocar un OA en el almacenamiento de datos con identificadores únicos que permitan que sea localizado posteriormente.
- Obtener metadatos: Se debe poder obtener metadatos de OA para poder hacer búsquedas ampliadas.
- Control de Calidad: Un sistema que asegure que los OA satisfagan requerimientos técnicos, educativos y de metadatos.
- Búsquedas: El repositorio de OA debe proveer la facilidad de localizar un OA apropiado de acuerdo

a distintos criterios de búsqueda. Esto puede incluir la funcionalidad de mostrar OA para luego hacer la selección.

- Mantener OA: El repositorio debe implementar un control de versiones adecuado.

A continuación se mencionan algunas iniciativas desarrolladas en el contexto de los ROA [20]:

- Multimedia Educational Resource for Learning and Online Teaching (MERLOT) es un repositorio centralizado que contiene sólo los metadatos y apunta a los objetos ubicados en sitios remotos. Es independiente y funciona como un portal de OA. Provee búsquedas y otros servicios como personalización, importación y exportación de objetos. Cualquier usuario puede tener acceso a todos los objetos contenidos en MERLOT y sólo los miembros contribuyen agregando objetos. La revisión por pares es una actividad que MERLOT utiliza para evaluar la calidad de los objetos agregados. [1]
- Campus Alberta Repository of Educational Objects (CAREO) es un repositorio centralizado de OA multidisciplinarios de profesores de Alberta (Canada). Es un repositorio independiente que da acceso a objetos remotos y locales a través de los metadatos contenidos en su colección. Cualquier usuario puede tener acceso a los objetos, pero los miembros tienen servicios adicionales, al igual que MERLOT ser miembro es gratis y abierto a cualquier persona. [2]
- Science, Mathematics, Engineering and Technology Education (SMETE) es un repositorio distribuido, que se presenta como una biblioteca digital que integra de forma federada las colecciones de varias bibliotecas de recursos educativos. El acceso es libre para la consulta. [3]

La UCI, también cuenta con su propio repositorio de OA RHODA el cual cumple con todas las cracterísticas de los ROA mencionados anteriormente y adicionalmente permite en el propio repositorio la creación de OA. [4] El RHODA constituye un espacio virtual que pone a la disposición de la comunidad académica una colección de OA reutilizables. Bajo la figura de recurso de aprendizaje de acceso abierto y soporta los estándares para estructura y empaquetado: Learning Object Metadata (LOM) y Sharable Content Object Reference Model (SCORM).

[1] Disponible en http://www.merlot.org/
[2] Disponible en http://www.careo.org/
[3] Disponible en http://www.smete.org/smete/
[4] Disponible en http://roa.uci.cu/

En RHODA, los usuarios finales pueden asumir varios roles, los cuales pueden ser creados dinámicamente, asignándole los permisos pertinentes para una necesidad específica. Propone por defecto, cinco roles (Invitado, Usuario, Autor, Revisor, Administrador), los cuáles se encuentran organizados jerárquicamente, donde cada uno hereda los privilegios del nivel inferior.

Figura 1.3: Organización de roles en RHODA.

Entre el amplio abanico de prestaciones que RHODA posee, es importante mencionar que da la posibilidad a los autores de crear OA tanto de forma individual como colaborativa, y establece tres etapas por las que debe pasar un OA:

- Edición: Es la etapa inicial, donde se crea el OA e involucra a los autores.
- Revisión: Finalizada la edición, el OA pasa a la fase que se puede catalogar como decisiva, debido a que es aquí, donde se estará realizando las revisiones necesarias, que evalúan su calidad. Esta etapa involucra a los Revisores, que son los autorizados y capacitados para realizar esta tarea.
- Publicado: Si el Revisor encuentra que el OA tiene la calidad requerida, la ficha del OA se acredita con el sello de calidad, que lo avala como un contenido de calidad, que ha pasado por la revisión de especialistas antes de ponerlo a disposición de las comunidades académicas. Y este pasa a ser accesible para todos los usuarios, incluyendo el Invitado.

Básicamente RHODA constituye un lugar de apoyo a la comunidad universitaria, para el trabajo metodológico colaborativo, orientado a elevar la calidad de los recursos didácticos y el proceso de enseñanza aprendizaje. Los OA que en él se encuentran pueden ser seleccionados para complementar cursos que se encuentran en el Entorno Virtual de Aprendizaje (EVA) o para la autosuperación de estudiantes y profesores.

1.1.3. Estándares para la composición y manejo de los Objetos de Aprendizaje

Un estándar no es más que una especificación dictada por organizaciones reconocidas por sus estudios y aportes, como son AICC, ADL, ARIADNE, IMS Global Consortium Inc., IEEEL LTSC, W3C, entre otras, para posibilitar que una amplia gama de productos coexistan dentro de un mercado [19].

Para los OA la filosofía es la misma, para estos también existen estándares que abarcan aspectos de gran relevancia como la estructura y el empaquetado, fundamentalmente. Tal es el caso de LOM propuesta por Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) y SCORM creado por el Departamento de la Defensa de Estados Unidos. Aunque los anteriores no son los únicos estándares que existen en este campo, son los que se van a tratar en este epígrafe debido a que el repositorio RHODA, ofrece soporte para ellos.

“En el año 2002 se emite el estándar 1484.12.1 (IEEE, 2002) que acredita al modelo de datos LOM como el estándar de metadatos para OA. LOM especifica la semántica y la sintáctica de un conjunto mínimo de metadatos necesario para, completa y adecuadamente, identificar, administrar, localizar y evaluar un OA. Su propósito es facilitar a profesores, alumnos y a sistemas automáticos la tarea de buscar, compartir e intercambiar OA, permitiendo el desarrollo de catálogos que contemplan la diversidad cultural e idiomática de los contextos en los que se puedan utilizar los objetos y sus metadatos”[20].

Para organizar los metadatos, LOM los agrupa en nueve categorías [20]:

- General: Información general que describe el OA como un todo.
- Ciclo de vida: Características relacionadas con la historia y el estado presente del OA y de aquellos que han afectado a este objeto durante su evolución.
- Meta-metadatos: Información sobre los mismos metadatos, no sobre el OA que se está describiendo.
- Técnica: Requisitos y características técnicas del OA.
- Educativa: Condiciones del uso educativo del recurso.
- Derechos: Condiciones de uso para la explotación del recurso.

- Relación: Relación del recurso descrito con otros OA.
- Anotación: Comentarios sobre el uso educativo del OA.
- Clasificación: Descripción temática del recurso en algún sistema de clasificación.

El modelo SCORM “es un conjunto de estándares y especificaciones para compartir, reutilizar, importar y exportar OA” [16]. Retomando las ideas de Guzmán.

“Este modelo describe cómo las unidades de contenidos se relacionan unas con otras a diferentes niveles de granularidad, cómo se comunican los contenidos con el Learning Management System (LMS), define cómo empaquetar los contenidos para importarse y exportarse entre plataformas, y describe las reglas que un LMS debe seguir a fin de presentar un aprendizaje específico” [20].

Los detalles de la especificación se encuentran en cuatro documentos a los que se da mantenimiento de manera independiente, reflejados en la Figura 1.4.

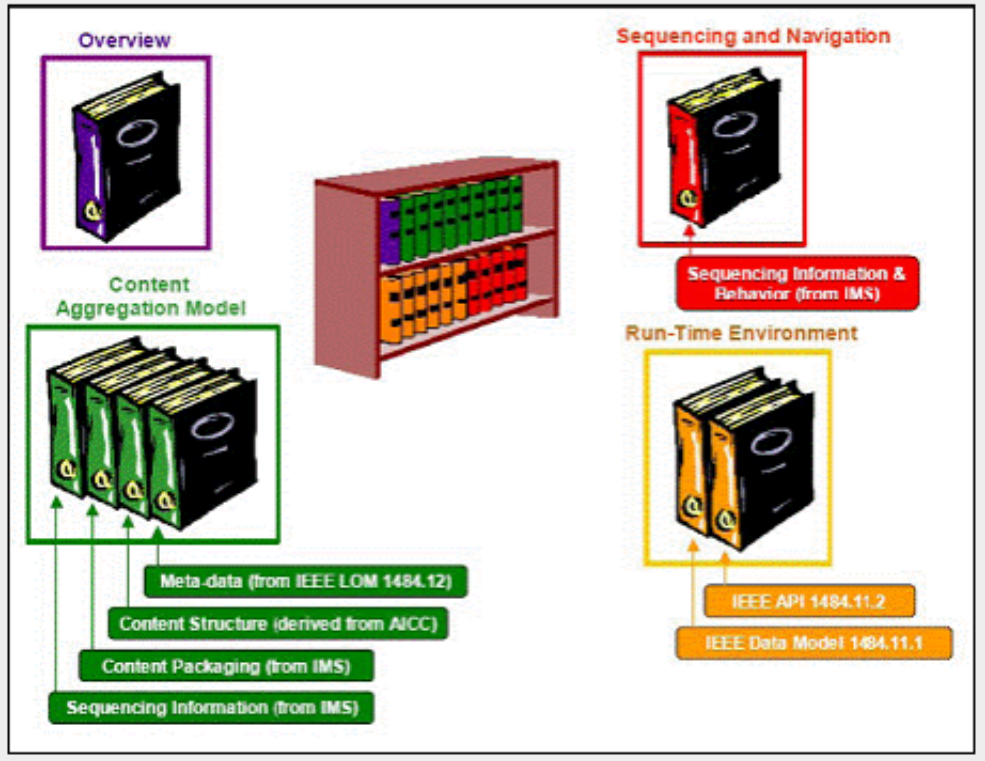

Figura 1.4: Los documentos que conforman el estándar SCORM [21]

1.2. Evaluación de los Objetos de Aprendizaje

La evaluación de los OA, es un factor decisivo ya que si estos al ser publicados carecen de la calidad necesaria podría provocar desmotivación en los usuarios que consumen este tipo de recurso didáctico, afectaría el proceso de enseñanza-aprendizaje y los resultados obtenidos producto de su aplicación divergirían a gran escala de lo esperado. Por esta razón la tendencia está dirigida a la constante evaluación de los OA mediante el empleo de modelos y guías desarrolladas especialmente para cubrir estas necesidades.

"... no se puede hablar de consolidación del eLearning sin contar con las suficientes garantías de calidad. La clave está en dejar en un segundo plano las modalidades y las herramientas para centrarse realmente en el objetivo de la formación y del aprendizaje", establece García Peñalvo y añade que "contar con acreditaciones y/o sellos de calidad otorgados por entidades externas e internacionalmente reconocidas constituye una cierta garantía, y es a la vez una forma efectiva de que las instituciones pongan los medios oportunos para lograr y mantener una calidad en sus productos formativos" [22].

1.2.1. Evaluación manual de los Objetos de Aprendizaje

Hasta el momento no se ha encontrado un modelo especialmente diseñado y estandarizado para las evaluaciones de calidad de los OA, en lugar de esto las organizaciones que dedican su trabajo al uso y manejo de OA han definido sus propios modelos y guías de evaluación, entre las que se destacan:

- Learning Object Review Instrument (LORI). Proporciona un marco de evaluación de OA basado en el análisis de nueve dimensiones. Las nueve dimensiones de los OA evaluadas por este instrumento son: Calidad de Contenido, Alineamiento de los objetivos de aprendizaje, Retroalimentación y adaptación, Motivación, Diseño de Presentación, Usabilidad en la interacción, Accesibilidad, Reusabilidad, Cumplimiento de Estándares [23].
- Reeves. Consta de 14 dimensiones pedagógicas basadas en teorías y conceptos de aprendizaje. Estas dimensiones han sido usadas para evaluar cursos en ambientes de e-learning y, si se considera a un curso como un OA con alto nivel de agregación, este modelo puede usarse para evaluar

la calidad desde el punto de vista pedagógico. Estas dimensiones son: epistemológica, filosofía pedagógica, sustento psicológico, orientación a objetivos, validez experimental, rol del instructor, flexibilidad de programa, valor del error, motivación, adaptación a diferencia a individuales, control de aprendizaje, actividades de usuario, aprendizaje cooperativo y sensibilidad cultural [24].

- Guía de evaluación de la calidad de los OA. Creada en la UCI para la evaluación de los OA que se encuentran el repositorio RHODA la cuál mide tres aspectos a evaluar (Formativos, Diseño y presentación y Tecnológicos) donde cada uno está compuesto por un número de indicadores el cuál puede variar en dependencia de lo que se quiera lograr [8].

Si bien se puede afirmar que los modelos y guías son una base para obtener evaluaciones de los OA lo más verídicas y objetivas posibles, estas, en su mayoría son aplicadas manualmente, condición propicia para la introducción inconsciente de errores por parte de los revisores y por consiguiente, obtención de resultados irreales.

1.2.2. Evaluación automatizada de los Objetos de Aprendizaje

Como resultado de un estudio de las herramientas existentes para la evaluación de los OA, se obtuvo que, para facilitar el trabajo de los revisores se han desarrollado herramientas como: la Herramienta de Objetos Didácticos de Aprendizaje Reutilizables (HEODAR) y la plataforma e-learning Adaptive Hypermedia Knowledge Management E-learning Platform (AKHME), de la Universidad de Salamanca, España. La Plataforma Asistencia para la Gestión de Objetos Reusables de Aprendizaje (AGORA) y el framework FLOE-T, una herramienta para la evaluación y estudio de Learning Object en los portales, de la Universidad Tecnológica de Panamá. La Tabla 1.1 muestra las características de cada una de ellas.

Tabla 1.1: Herramientas para la evaluación de Objeto de Aprendizaje

Nombre	Tipo	Descripción

HEODAR	Herramienta de software	La herramienta propone dos aspectos a evaluar (pedagógicos y técnicos). Es flexible en cuanto a que permite elegir que indicadores, dentro de los aspectos establecidos, se tendrán en cuenta para realizar la evaluación del OA. Carga los OA desde la plataforma Moodle a la cual se encuentra integrada. Establece una proporción a los aspectos definidos a evaluar expresado en porciento. Evalúa los aspectos definidos cuantitativamente cuyas escalas se dan en porciento y da también da una evaluación cualitativa. Permite evaluar recursos con la especificación SCORM.
AKHME	Plataforma e-learning	Es una plataforma e-learning que esta dividida en cuatro subsistemas (Administración de OA y Subsistema de Diseño de Aprendizaje, Subsistema de Administración de Conocimientos, Subsistema Adaptativo y Visualización y Subsistema de Presentación). La plataforma almacena OA y estos son evaluados por el subsistema de Administración de Conocimientos donde se encuentran accesibles dos herramientas evaluadoras, una es colaborativa y la otra es un agente inteligente que evalúa el OA automáticamente, ambas guiadas por criterios que se ajustan a cuatro categorías (Psicopedagógica, Didáctico-Curricular, Técnica-Estética y Funcional) con sus respectivos pesos dados en porciento. El estándar de empaquetado de OA es Instructional Management System (IMS).

AGORA	Plataforma e-learning	La plataforma está conformada por un conjunto de módulos que se integran en una aplicación Web. Cuenta con un repositorio para el almacenamiento de OA, desde donde se cargan estos para el proceso de evaluación. Los usuarios pueden expresar sus opiniones con respecto a la calidad del contenido de un recurso educativo. Las evaluaciones están basadas en un modelo de calidad denominado Modelo para la Evaluación de la Calidad en Objetos de Aprendizaje (MECOA) el cual define seis aspectos a evaluar. 1. Contenido del OA. 2. Representación del OA. 3. Competencias logradas a partir de la interacción con el OA. 4. Autogestión en la interacción con el OA. 5. Significación del OA. 6. Creatividad que desarrolla el OA. Los metadatos se presentan en una estructura XML acorde al estándar IEEE-LOM para la descripción del OA. Los recursos educativos colocados en el repositorio de AGORA pueden consultarse para ver sus evaluaciones de calidad y determinar si resultan apropiados para formar parte de una actividad de aprendizaje.
FLOE-T	Framework	FLOE-T es una aplicación cliente que se descarga desde el portal del proyecto FLOE-T. La herramienta carga OA desde repositorios ubicados en Internet. Tiene definidos cuatro criterios (Calidad positiva, Calidad negativa, Entrega y Calidad formativa) para la implementación del modelo de evaluación. Los revisores expertos ya registrados son los únicos autorizados en cargar la Uniform Resource Locator (URL) del OA que se desea evaluar y los datos generados del proceso de evaluación son almacenados en un servidor de bases de datos.

Cada una de las herramientas vistas apoya el proceso evaluativo de los OA, propiciando agilidad y efecti-

vidad en el mismo. Sin embargo, tienen la desventaja de ser inflexibles en cuanto a criterios para evaluar la calidad y a la integración con repositorios de OA. Además los instrumentos de evaluación de la calidad que utilizan, difieren en gran medida con el instrumento empleado en la UCI. Por lo que se propone implementar una herramienta de evaluación de la calidad de los OA que aplique la Guía de la calidad de los OA empleada en la UCI y se integre a RHODA.

1.3. Metodología y herramientas

A continuación se definen las metodología, herramientas y tecnologías que han de usarse para el desarrollo de esta investigación.

1.3.1. Metodología de desarrollo de software Rational Unified Process

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software [36]. Estas se adoptan con el objetivo de obtener un producto de mayor calidad y un proceso de desarrollo de software con características tales como: existencias de reglas bien definidas, verificaciones intermedias, planificación y control, comunicación efectiva y utilización sobre un abanico amplio de proyectos.

Rational Unified Process (RUP) es un proceso de Ingeniería de Software (ISW) propuesto por Rational Software Corporation para la construcción completa del ciclo de ISW. Permite la productividad en equipo y la realización de mejores prácticas de software a través de plantillas y herramientas que lo guían en todas las actividades de desarrollo del software. Es un producto que unifica las disciplinas en lo que a desarrollo de software se refiere, incluyendo modelado de negocio, manejo de requerimientos, componentes de desarrollo, ingeniería de datos, manejo y configuración de cambios, y pruebas, cubriendo todo el ciclo de vida de los proyectos basado en la construcción de componentes y maximizando el uso del Unified Modeling Language (UML). Se resume en tres aspectos definitorios: dirigido por casos de uso, iterativo e incremental y centrado en la arquitectura, lo que hace único al proceso. RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor empeño en las distintas actividades. Para el desarrollo de RHODA se

utilizó esta metodología de desarrollo de software por lo que se continua su uso en el desarrollo del módulo para la evaluación de la calidad de los OA.

1.3.2. Herramienta de modelado Visual Paradigm Suite

Es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software. Ayuda a una rápida construcción de aplicaciones con calidad, mejores y a un menor esfuerzo. Modela diagramas de clases, código inverso y genera código desde diagramas de bases de datos. Permite la transformación de diagramas de Entidad-Relación en tablas de base de datos. Brinda la posibilidad de Integración con Visio para el dibujo de diagramas UML y posee un Generador de informes para generación de documentación. Se propone emplear la herramienta Ingeniería de Software Asistida por Computadora (CASE) Visual Paradigm Suite versión 6.4 para modelar los artefactos que se obtienen en el ciclo de vida del módulo para la evaluación de la calidad de los OA por las características antes mencionadas y por haberse utilizado en la modelación de los módulos anteriores de la plataforma RHODA.

1.3.3. Lenguaje de programación Hypertext Preprocessor

Es un lenguaje de programación web usado normalmente para la creación de contenido para sitios web con los cuales se puede programar las páginas HTML y los códigos de fuente. Es libre, por lo que se presenta como una alternativa de fácil acceso para todos. Permite las técnicas de Programación Orientada a Objetos. Es muy rápido y su integración con la base de datos PostgreSQL y el servidor Apache, le permite constituirse como una de las alternativas más atractivas del mercado [37]. RHODA está desarrollada en el lenguaje Hypertext Preprocessor (PHP) versión 5.3, que porta gran facilidad de uso, de aprendizaje y posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.

1.3.4. Gestor de Base de Datos PostgreSQL

Es un sistema de gestión de base de datos relacional orientada a objetos y libre que esta diseñado para administrar grandes cantidades de datos. Es robusto, confiable y mantiene la integridad de los datos. Se

ejecuta en la mayoría de los Sistemas Operativos más utilizados en el mundo incluyendo Linux, varias versiones de UNIX y en Windows. Además soporta funciones donde la salida puede tratarse como un conjunto de valores que son tratados igual a una fila retornada por una consulta. Son escritos en varios lenguajes, con la potencia que cada uno de ellos ofrece, desde las operaciones básicas de programación, tales como bifurcaciones y bucles, hasta las complejidades de la programación orientada a objetos o la programación funcional. Se utiliza PostgreSQL versión 8.4, pues es un software de código abierto, provee servicio y procesamiento a múltiples usuarios simultáneamente, permite a una aplicación realizar varias tareas concurrentemente y realiza un acertado trabajo junto a la programación PHP.

1.3.5. Symfony

Es un completo framework diseñado para optimizar el desarrollo de aplicaciones web. Emplea el patrón de diseño Modelo Vista Controlador (MVC) para separar las distintas partes que forman una aplicación web, la lógica de negocio, la lógica de servidor y la presentación. Además proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Es importante destacar que automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El desarrollo de la plataforma RHODA en su versión 1.0 se realizó con este framework, debido a que está preparado para aplicaciones empresariales y es adaptable a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo y poseer una gran cantidad de documentación, lo que propicia un mejor aprendizaje por parte de los desarrolladores, por lo que para el desarrollo del módulo para la evaluación de la calidad de los OA se debe utilizar dicho framework en su versión 1.3.8.

1.3.6. Entorno de Desarrollo Integrado NetBeans

Es un entorno de desarrollo integrado, una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el IDE NetBeans. Es un producto libre y gratuito sin restricciones de uso. Brinda la posibilidad al programador de administrar las interfaces y las configuraciones del usuario.

1.3.7. Servidor web Apache

Es una tecnología gratuita de código abierto; servidor altamente configurable de diseño modular que permite ampliar sus capacidades. Trabaja con gran cantidad de lenguajes como: Perl, PHP y otros como Java y páginas jsp, teniendo todo el soporte que se necesita para tener páginas dinámicas. Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor, posibilitando configurarlo para que ejecute un determinado script cuando ocurra un error concreto. Tiene una alta configurabilidad en la creación y gestión de logs, como medida del administrador para un mayor control sobre lo que sucede en el servidor [37].

Conclusiones

A lo largo de la investigación realizada se pudo adquirir la base teórica necesaria para lograr cumplir con los objetivos trazados en desarrollo de la solución propuesta. Para ello, se abordan los principales conceptos relacionados con la temática a tratar.

Partiendo de la necesidad que posee el repositorio de OA de la universidad, de contar con un sistema de software para la evaluación de la calidad de los OA, se realizó un estudio de sistemas similares. El objetivo principal de este estudio, fue precisamente encontrar un sistema empleado con el mismo fin y que este, a su vez, pudiera ser reutilizado, lo cuál resultó imposible.

De acuerdo a las características que poseerá el sistema y el entorno donde este será desplegado, se determinaron cuidadosamente las metodologías y herramienta a utilizar. Estas últimas, fueron seleccionadas en función de imprimir la mayor calidad posible al software y su proceso de desarrollo. Se tomaron en consideración todos los factores que influyen en el mismo ya ses de forma directa o indirecta.

Capítulo 2

Características del sistema

Introducción

En el presente capítulo se hace una descripción de la solución propuesta como parte del análisis y diseño del sistema perteneciente a la fase de elaboración en concordancia con los flujos de trabajo propuestos por RUP.

Para lograr los objetivos trazados, se obtiene el modelo de dominio con el objetivo de lograr un mayor entendimiento del dominio del problema. Para ello se identifican las clases que interactúan en la realidad física del sistema a desarrollar y sus relaciones. Se identifican los requerimientos funcionales y no funcionales. A partir de este punto se generan los Caso de Usos (CUs) para lograr ver con mayor facilidad las funcionalidades que el sistema debe poseer, describiendo claramente su flujo de eventos, de forma tal que estos sean claros y entendibles. Todo esto proporciona las bases necesarias para generar el modelo de análisis y diseño con sus respectivos artefactos generados.

Se aplica la metodología RUP y se hace uso de la herramienta Visual Paradigm y UML como lenguaje de modelado. Se generan diagramas con los que se amplía el conocimiento y entendimiento del sistema, los cuales permiten obtener una concepción general aterrizada y ciertamente más clara de lo que se pretende desarrollar.

2.1. Características del sistema

El Módulo para la evaluación de la calidad de los OA surge a partir de la necesidad de automatizar el proceso de evaluación de la calidad de los OA. El RHODA, cuenta con un amplio abanico de funcionalidades que permiten que su uso sea sencillo y provechoso. Los OA que se crean no se publican directamente, sino que son sometidos a un riguroso proceso de revisión que evalúa su calidad.

En el proceso de revisión juegan un papel protagónico los instrumentos de evaluación, los cuáles sirven de guía al revisor. Entre los instrumentos más relevantes se encuentra la Guía de evaluación de la calidad de los OA. Esta guía está especialmente concebida para su empleo en la UCI y tiene como base teórica el estudio de otros instrumentos reconocidos en el mundo. Para que las evaluaciones que se realicen sean lo más precisas y cómodas posibles, se realiza la automatización del proceso con la construcción del módulo de evaluación a incorporar en el RHODA.

El módulo de revisión tiene como base la Guía de evaluación de la calidad de los OA. Esto no quiere decir que sea el único instrumento que se pueda emplear, pero sí el más recomendado. Cada OA es asignado a un revisor para su evaluación, este según sea su nivel de dominio de los aspectos a evaluar, puede agregar a otros revisores a su revisión. Cada cambio hecho sobre cada evaluación se queda almacenado en la base de datos pudiendo ser retomada en cualquier momento. Una vez completada la evaluación de los aspectos propuestos, el sistema calcula la evaluación final. Esta evaluación final influye en la toma de decisiones con el OA respecto a su publicación en el repositorio.

Una vez informatizado todo el proceso de revisión el sistema garantizará:

- Importar y evaluar los OA que se encuentran bajo los estándares SCORM y LOM.
- Visualizar los metadatos y estructura didáctica definidas por el usuario que construyó el OA.
- Proveer una base de almacenamiento de los cambios en la evaluación para realizar análisis posterior y retomar las evaluaciones no finalizadas.
- Realizar evaluaciones cualitativas y cuantitativas del OA.
- Proveer un sistema de configuración definidos por políticas de acceso.
- Exportar los resultados de las evaluaciones a otros formatos para posteriores análisis.
- Gestionar el instrumento de evaluación a emplear, permitiendo el restablecimiento del original.

Todo esto incidirá en una eficiente y rápida toma de decisiones de un OA determinado, permitiendo realizar comparaciones a partir de datos históricos y análisis correlacionales entre varios registros de revisiones realizadas. A continuación mediante el análisis y diseño del sistema, se ofrece una descripción más detallada y que complementa con otros elementos lo visto hasta el momento.

2.2. Conceptos asociados al dominio del problema

En el ámbito asociado al dominio de la investigación, se identificó que no se manejan los procesos de negocio. El motivo es que resulta difícil captar claramente estos procesos, imposibilitando determinar el conjunto estructurado de las actividades que se desarrollan en el negocio, toda la organización, las relaciones etc. Es por ello que se determinó el uso del modelo de dominio para lograr comprender los conceptos con los que se trabajan en el desarrollo y puesta en práctica de la aplicación.

El modelo de dominio es uno de los artefactos generados como resultado del desarrollo de la disciplina de análisis y que alcanza mayor preeminencia en la fase de elaboración. El análisis es el paso que precede al diseño del sistema, tal vez por esta razón resulta tan importante detallar bien cada artefacto y documento generado en esta etapa. Básicamente, el modelo de dominio se representa como uno o un conjunto de diagramas de clases y engloba los conceptos de la propia realidad física en la que ejerce el sistema. Con la construcción del modelo de dominio, se persigue plasmar el conocimiento adquirido en una determinada área. Es utilizado como una útil herramienta para comprender el sector industrial o de negocios al cual el sistema va a servir. Una vez completado este paso es probable que se desarrolle una amplia visión del sistema y el entorno en que este será desplegado posteriormente.

Más concretamente, ¿qué elementos quedarán representados en el diagrama de dominio?, a continuación se listan los que se consideran más relevantes[26]:

- Conceptos u objetos del dominio del problema: clases conceptuales.
- Asociaciones entre las clases conceptuales.
- Atributos de la clase conceptuales.

Esto no quiere decir que representar cada uno de los elementos antes mencionados sea un procedimiento estricto. Es posible capturar un mayor grado de detalle en dependencia de lo que se quiera lograr. Por esta razón está abierta la decisión de cuanto detalle va a ser necesario o no incluir en cada modelo. El objetivo es captar lo suficiente para entender el medio en el cual el sistema que estamos diseñando va a funcionar y esto demanda una cantidad distinta de detalles cada vez.

El entorno donde está presente el problema se puede ver en modelo de domino de la Figura 2.1

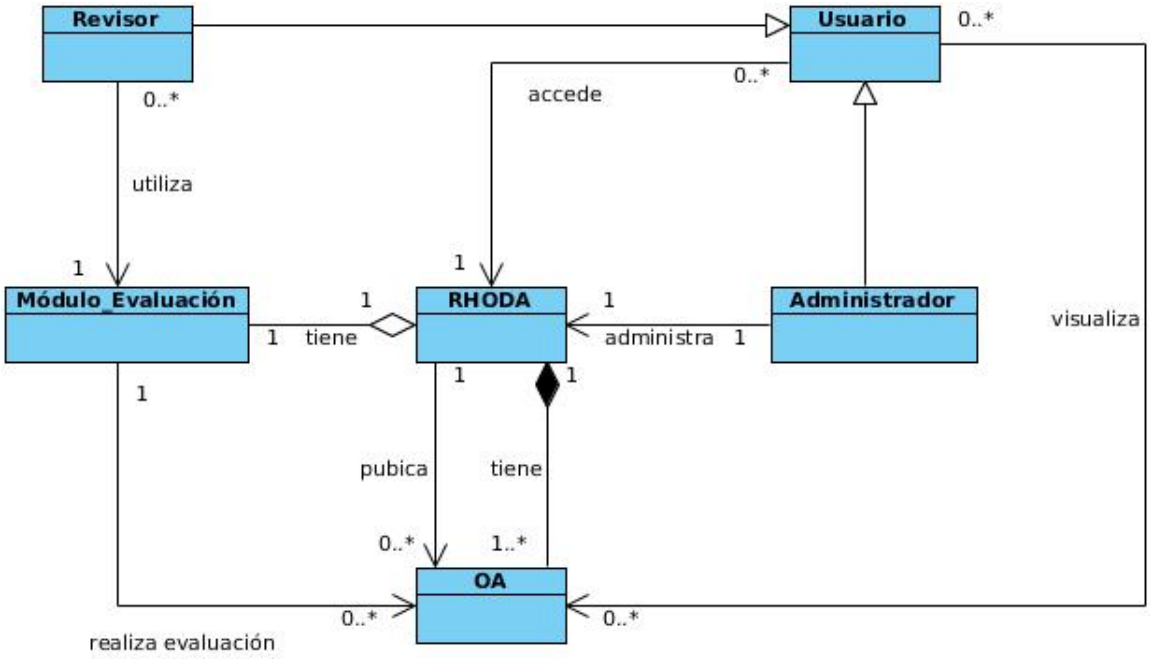

Figura 2.1: Modelo de Dominio

A continuación se detallan cada una de las clases presentes en el modelo de dominio propuesto.

- **Administrador**: En casos especiales es necesario eliminar o añadir indicadores y aspectos a la evaluación, para esto existe un administrador encargado de esta y otras tareas vinculadas a la administración del repositorio en sentido general.
- **Módulo_Evaluación**: Es el módulo que se adjunta al RHODA desde el cual es posible realizar la evaluación de los OA, cuya evaluación es tenida en consideración para decidir si los OA se encuentran con la calidad requerida para ser publicados.
- **OA**: Son los recursos didácticos que se almacenan en el RHODA que pueden ser accedidos por los usuarios en apoyo al aprendizaje. Los OA pueden ser accedidos siempre que estos hayan obtenido una evaluación de calidad satisfactoria.
- **Revisor**: La revisión se lleva a cabo por un equipo de revisión. El equipo de revisión está compuesto por uno o varios revisores, los cuales constituyen el personal capacitado y con conocimientos en los aspectos que mide la evaluación.

- **RHODA**: Es el lugar donde se crean, modifican, eliminan, evalúan y publican los OA. Para esto incluye un conjunto de módulos encargados de cumplir tareas específicas.
- **Usuario**: Cualquier persona que interactúe con el repositorio, ya sea para crear, revisar o simplemente visualizar los OA que en él se encuentran.

2.3. Especificación de requisitos del software

Los requisitos de software son sin duda un punto clave en el proceso de desarrollo de software. Si la captura, definición, control de cambios y administración de estos no se realiza con la precisión que lleva, el proyecto podría llegar al punto de fracasar completamente. He aquí la importancia de esta actividad.

Existen numerosas definiciones dadas por muchos autores reconocidos en cuanto al término de requerimiento de software. He aquí algunas definiciones que dadas por la IEEE e Ian Somerville:

- Condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo [35].
- Condición o capacidad que debe estar presente en un sistema o componentes de un sistema para satisfacer un contrato, estándar, especificación u otro documento formal [35].
- Declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de éste [39].

A continuación se enuncian las características de un requerimiento propuestas por el autor Michael Arias Chaves en su artículo “La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software”.

Es importante no perder de vista que un requerimiento debe ser:

- Especificado por escrito: Como todo contrato o acuerdo entre dos partes.
- Posible de probar o verificar: Si un requerimiento no se puede comprobar, entonces ¿cómo se sabe si se cumplió con él o no?

- Conciso: Un requerimiento es conciso si es fácil de leer y entender. Su redacción debe ser simple y clara para aquellos que vayan a consultarlo en un futuro.
- Completo: Un requerimiento está completo si no necesita ampliar detalles en su redacción, es decir, si se proporciona la información suficiente para su comprensión.
- Consistente: Un requerimiento es consistente si no es contradictorio con otro requerimiento.
- No ambiguo: Un requerimiento no es ambiguo cuando tiene una sola interpretación. El lenguaje usado en su definición, no debe causar confusiones al lector.

2.3.1. Requisitos funcionales del software

Los requisitos funcionales del software son aquellos que describen las funciones que el sistema debe ser capaz de realizar. Usualmente, la mayoría de ellos se convierten en CU del sistema y posteriormente en algoritmos que dan vida al sistema desarrollado.

Sin perder de vista todo lo visto hasta el momento, se realizó un análisis de los procesos del negocio definiendo las actividades que se deben automatizar. Estas actividades constituyen la base para identificar los requisitos funcionales del sistema. A continuación se presentan los requerimientos identificados durante el análisis realizado al Módulo de evaluación de la calidad de los OA que presentan una prioridad alta para el cliente. El resto de los requerimientos identificados de encuentra en el Anexos A.

RF1. **Evaluar OA cuantitativamente:** El sistema debe ser capaz de emitir la evaluación cuantitativa del OA dada en porciento.

RF2. **Evaluar OA cualitativamente:** El sistema debe ser capaz de emitir la evaluación cualitativa del OA la cual se encuentra dividida en cuatro posibles categorías (No adecuado, Poco adecuado, Adecuado y Muy adecuado).

RF3. **Visualizar OA:** El sistema debe ser capaz de visualizar el contenido del OA al mismo tiempo que este se evalúa, de forma tal que el revisor pueda determinar con mayor grado de exactitud la evaluación de la calidad del mismo.

RF4. **Guardar estado de la evaluación:** El sistema debe ser capaz de guardar el estado de la evaluación almacenando en la base de datos los últimos cambios realizados.

RF5. **Cargar estado de la evaluación:** El sistema debe ser capaz de cargar el estado de la evaluación de un OA que previamente haya sido almacenada en la base de datos. El revisor puede continuar la revisión del OA a partir del último cambio realizado.

RF6. **Restaurar instrumento de evaluación:** El sistema debe ser capaz de restaurar los aspectos e indicadores oficiales que propone el instrumento: Guía de evaluación de la calidad de los OA.

2.3.2. Requisitos no funcionales del software

Los requerimientos no funcionales, como su nombre sugiere, son aquellos requerimientos que no se refieren directamente a las funcionalidades específicas que proporciona el sistema, sino a las propiedades emergentes de este [39].

Las propiedades emergentes se refieren al comportamiento de los sistemas en su entorno operativo. A menudo son factores críticos para sistemas informáticos, ya que un fallo mínimo en estas propiedades puede hacer inutilizable al sistema. Algunos usuarios puede que no necesiten ciertas funciones de sistema, por lo que este puede ser aceptable sin ellas. Sin embargo, un sistema no fiable o demasiado lento es probablemente rechazado por todos los usuarios [39].

Para su correcto funcionamiento, el sistema propuesto, precisa el cumplimiento de los requerimientos no funcionales que se citan a continuación:

Usabilidad

RnF1. **Acceso al sistema:** El sistema debe brindar simplicidad y facilidad a su acceso. De manera que un usuario estándar pueda acceder a él sin mayores problemas.

RnF2. **Manejo del sistema:** Las funcionalidades y disposición de las mismas en el sistema deben se

comprensibles al usuario, de modo que este desarrolle habilidades para manejar el sistema con fluidez.

RnF3. **Sistema de ayuda:** El sistema contará con una ayuda que permite solucionar dudas durante su utilización.

Eficiencia

RnF1 **Capacidad de interacción con usuarios:** El sistema debe permitir la interacción con uno o varios usuarios, según sea la necesidad y será accedido vía web.

Soporte

RnF1 **Mantenimiento a través de versiones:** Se debe proporcionar mantenimiento al sistema con el objetivo de adicionar funcionalidades y/o corregir errores a través de versiones posteriores.

Restricciones de diseño

RnF1 **Lenguaje de programación PHP:** El sistema debe ser desarrollado empleando el lenguaje de programación PHP debido a que se trata de un módulo adjunto a un repositorio ya desarrollado bajo esta misma restricción. Se adopta esta restricción, con el objetivo de lograr que la integración al repositorio sea óptima.

RnF2 **Uso de Symfony como framework de desarrollo:** El sistema debe ser desarrollado empleando el framework de desarrollo Symfony debido a que se trata de un módulo adjunto a un repositorio ya desarrollado bajo esta misma restricción. Se adopta esta restricción, con el objetivo de lograr que la integración al repositorio sea óptima.

RnF3 **Patrón arquitectónico Modelo Vista Controlador:** El sistema sigue el patrón arquitectónico MVC siguiendo la arquitectura empleada por Symfony para lograr la mayor adherencia al framework y explotar al máximo sus potencialidades.

Interfaz

RnF1 **Estructura y diseño:** El sistema debe presentar adherencia máxima al sistema RHODA en cuanto a estructura y diseño, de modo que este exista concordancia y homogeneidad a apreciación del usuario.

RnF2 **Navegación:** La navegación por el sistema debe estar estructurada intuitivamente, sin cambios bruscos que puedan agobiar o desorientar al usuario.

2.4. Definición de los Casos de Uso del Sistema

Cada CU cumple con una porción de las funcionalidades que el sistema debe ofrecer para aportar un resultado de valor para los actores. El empleo de los CU en la construcción del sistema, tiene como objetivo especificar la secuencia de acciones que se requiere que el sistema realice como resultado de la interacción con los actores del mismo. Cada secuencia que trae consigo cada CU, puede incluir uno o más alternativas dentro de la secuencia a las que comúnmente se les conoce como flujo alterno.

2.4.1. Descripción de los actores del sistema

Al hacer alusión a los actores del sistema, se hace referencia a cualquier entidad externa, ya sean persona u otros sistemas externos que intercambian información con el sistema. De este modo cada actor puede ejecutar tantas tareas como se requieran durante la interacción actor–sistema [40].

Para la construcción del módulo de revisión fueron identificados los siguientes actores:

Tabla 2.1: Descripción de los actores del sistema.

Actor	Descripción
Revisor	Personal capacitado, responsable de realizar las evaluaciones de los OA. Posee suficientes conocimientos de los aspectos e indicadores que se miden en el proceso de evaluación.

Administrador	Persona responsable de realizar las funciones administrativas del sistema, como agregar, editar o eliminar indicadores o aspectos de acuerdo a las necesidades existentes.

2.4.2. Diagrama de Casos de Uso del sistema

Los CU son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario [32].

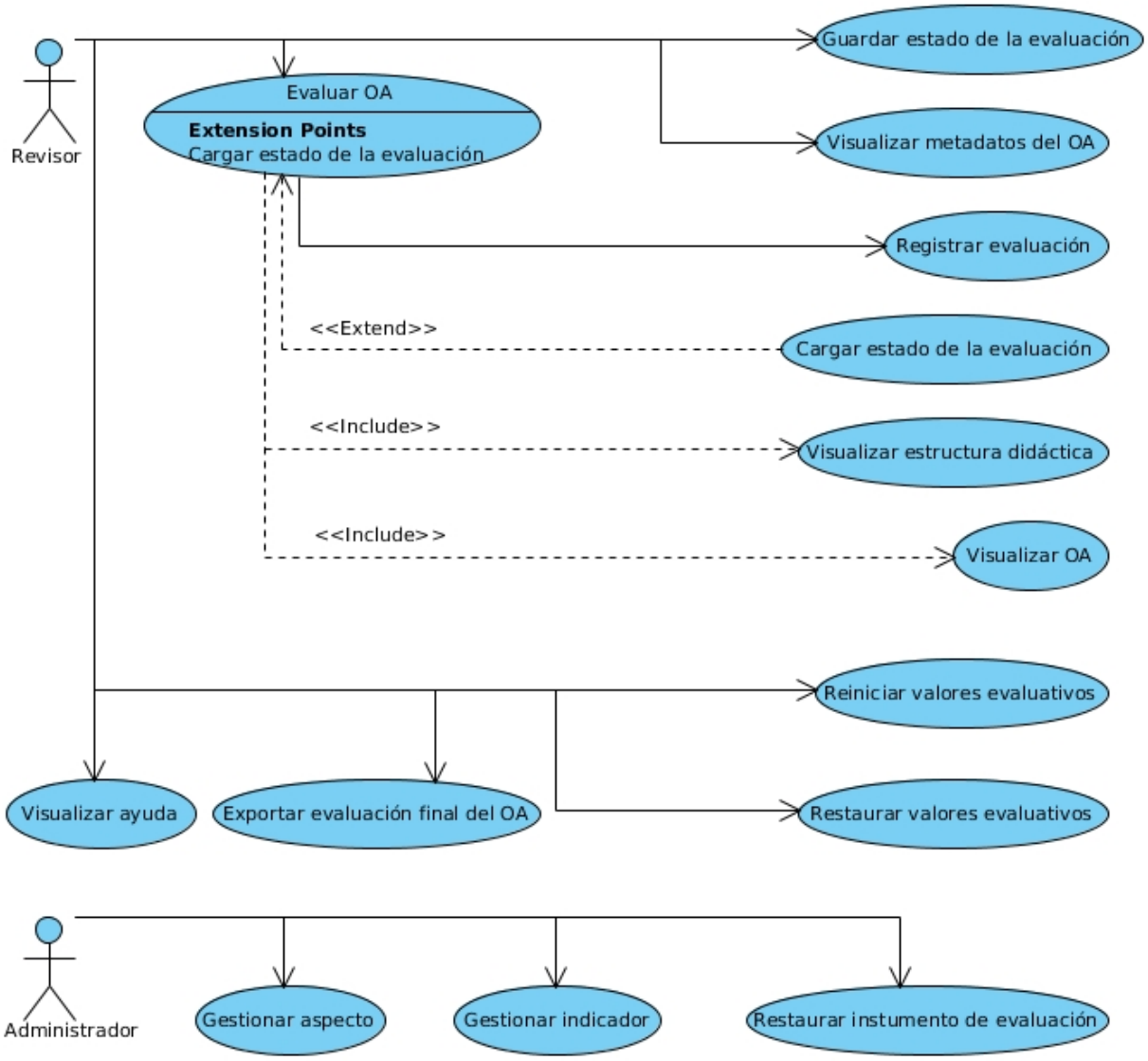

Figura 2.2: Diagrama de Casos de Usos

2.4.3. Descripciones de Casos de Uso

A continuación se expone el flujo de eventos que compone a los CU que fueron identificados cómo críticos a partir del análisis que propone el documento 0108_Evaluación de Casos de Uso del expediente de proyecto del programa de mejoras. Quedan plasmados en esta sección los CU identificados como críticos, el resto se puede consultar en el Anexo B.

Tabla 2.2: Descripción del CU Evaluar OA

Objetivo	Evaluar el OA emitiendo calidad alcanzada por este de forma cualitativa y cuantitativa.	
Actores	Revisor: (Inicia) Visualiza el OA para ganar en rigurosidad durante la evaluación, una vez terminada esta, se exportan los resultados a otros formatos en caso de ser necesario.	
Resumen	El CU inicia cuando el revisor procede a evaluar un OA, realiza la evaluación y el sistema calcula el resultado obtenido.	
Complejidad	Media	
Prioridad	Crítico	
Precondiciones	▪ El revisor se ha autenticado en el sistema. ▪ El revisor ha sido asignado a evaluar el OA.	
Postcondiciones	Se ha obtenido exitosamente el resultado de la evaluación del OA.	
Flujo de eventos		
Flujo básico “Evaluar OA”		
	Actor	Sistema
1.	1.1 Inicia el CU cuando accede a la sección de revisión del repositorio y selecciona un OA para evaluar.	1.2 Muestra la interfaz que contiene los indicadores evaluativos.
2.	2.1 Asigna valores a todos los indicadores presentes en la interfaz para la evaluación del OA.	

3.	3.1 Hace clic en el botón “Evaluar”. (Alterno1)	3.2 Guarda los cambios y realiza los cálculos aplicando la fórmula propuesta en el instrumento de evaluación, obteniendo el nivel de calidad del OA expresado en porciento (evaluación cuantitativa).
4.		4.1 Atribuye una evaluación cualitativa al OA a partir del resultado obtenido en cálculo de la evaluación cuantitativa.
5		5.1 Muestra una interfaz con el resultado obtenido de la evaluación (cuantitativa y cualitativa) general y por aspectos.
6.		6.1 Permite dos posibles acciones a realizar: 1. Volver a la evaluación. Ver (Sección1): “Volver atrás”. 2. Salir de la evaluación. Ver (Seccion2): “Salir de la evaluación”.
7.		7.1 Termina el CU.
Flujos alternos		
Nº 1 Evento “Faltan indicadores por evaluar”		
	Actor	Sistema
1.		1.1 Comprueba que restan indicadores por evaluar.
2.		2.1 Muestra el mensaje de error “Faltan indicadores por evaluar”.
3.		3.1 Regresa al paso 2.1 del flujo básico.
Sección 1: “Volver atrás”		
Flujo básico “Evaluar OA”		
	Actor	Sistema
1.	1.1 Selecciona la opción “Volver atrás”.	2.1 Regresa al paso 2.1 del flujo básico.
Sección 2: “Salir de la evaluación”		
Flujo básico “Evaluar OA”		

	Actor	Sistema
1.	1.1 Selecciona la opción “Salir”.	1.2 Redirecciona al usuario a la interfaz de revisión del repositorio.
2.		2.1 Regresa al paso 8.1 del flujo básico.

Relaciones	CU Incluidos	CU 2. Visualizar OA. CU 11. Visualizar estructura didáctica.
	CU Extendidos	CU 3. Cargar estado de la evaluación.

Tabla 2.3: Descripción CU Guardar estado de la evaluación

Objetivo	Guardar el estado de la evaluación del OA con los últimos cambios realizados en la evaluación del mismo.
Actores	Revisor: (Inicia) Visualiza el OA para ganar en rigurosidad durante la evaluación, una vez terminada esta, se exportan los resultados a otros formatos en caso de ser necesario.
Resumen	El CU inicia cuando se está evaluando un OA, seleccionando luego la opción “Guardar” para registrar las últimas modificaciones realizadas.
Complejidad	Media
Prioridad	Crítico
Precondiciones	El revisor ha seleccionado un OA para evaluar.
Postcondiciones	Se encuentra almacenado en la base de datos el estado de la evaluación del OA.
Flujo de eventos	
Flujo básico “Guardar estado de la evaluación”	

	Actor	Sistema
1.	1.1 Inicia el caso de uso cuando selecciona la opción “Guardar” que se encuentra en la interfaz que contiene los indicadores evaluativos.	1.2 Almacena en la base de datos la evaluación del OA con los últimos cambios realizados hasta ese momento.
2.		2.1 Termina el CU.

Relaciones	CU Incluidos	
	CU Extendidos	

2.5. Modelo de Análisis

En este punto el sistema cuenta con los requerimientos funcionales bien definidos aunque estos serán repasados posteriormente para corregir o agregar aquellos que se consideren necesarios. Además de esto las funcionalidades potenciales del sistema han sido agrupadas en los CU. El desarrollo del sistema de encuentra en la fase adecuada para mediante el Modelo de Análisis representar más detalladamente lo que este debe hacer.

Una definición formal del Modelo de Análisis se enuncia en “El proceso unificado de desarrollo de software”:

Constituye la primera aproximación al Modelo de Diseño, es el resultado del análisis de los CU y está conformado por las clases del análisis (interfaz, control y entidad), las cuales encapsulan las diferentes funcionalidades que representan los CU [27].

Una realización de CU – análisis es un colaboración dentro del Modelo de Análisis que describe cómo se lleva a cabo y se ejecuta un CU determinado en términos de clases del análisis y de sus objetos del análisis en interacción; proporciona por tanto una traza directa hacia un CU concreto del Modelo de Casos de Uso [27].

Como resultado de las actividades que se realizan en esta etapa como refinar los requisitos obtenidos anteriormente y la profundización en el dominio de la aplicación se pretende obtener mayor comprensión del problema para lograr modelar la mejor solución a la problemática planteada.

2.5.1. Diagrama de clases del análisis

Los diagramas de clases del análisis son artefactos que constituye una vista estática de las clases que conforman el modelo del análisis y las asociaciones entre las mismas. Es una vista de la futura composición de clases de software [33].

A continuación se presentan los Diagramas de Clases del Análisis de los CU críticos en el desarrollo del

sistema, ver Figuras 2.3 y 2.4. Para ver el resto de los diagramas de clases debe consultar el Anexo C.

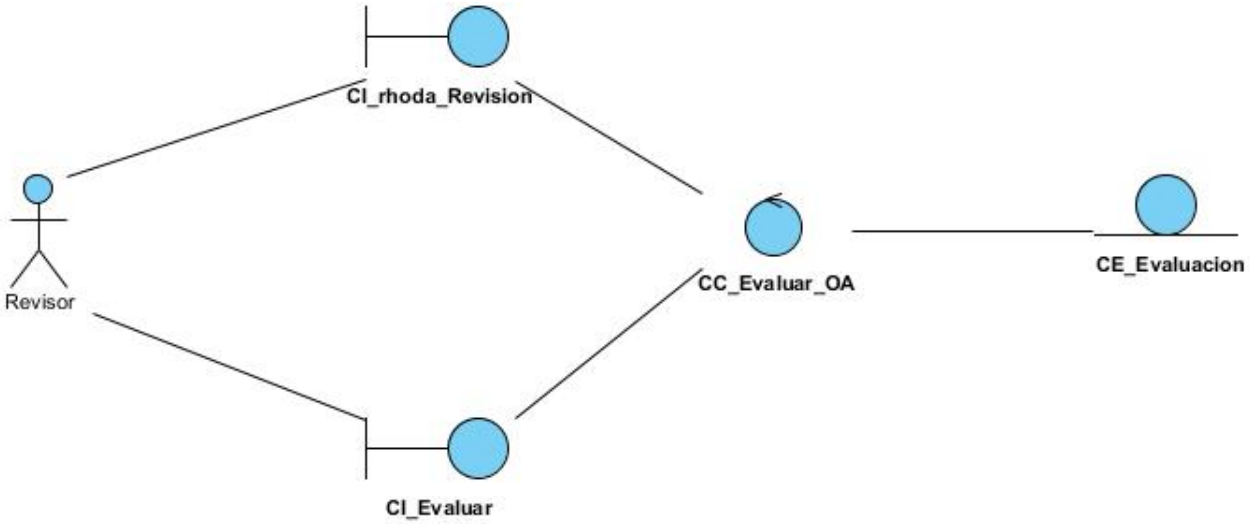

Figura 2.3: Diagrama de clases del análisis: CU Evaluar OA.

Figura 2.4: Diagrama de clases del análisis: CU Guardar estado de la evaluación.

2.5.2. Diagramas de interacción

Muestran una interacción concreta: un conjunto de objetos y sus relaciones, junto con los mensajes que se envían entre ellos [27].

Para la representación de los diagramas de interacción destacan dos elementos fundamentales ellos son:

- Diagramas de secuencia: Resaltan la ordenación temporal de los mensajes que se intercambian [27].
- Diagrama de colaboración / comunicación: Resaltan la organización estructural de los objetos que intercambian mensajes [27].

Los CU identificados para satisfacer los requerimientos del sistema, han sido modelados con diagramas

de interacción. Y puesto que el mayor interés radica en identificar responsabilidades y requisitos sobre los objetos, se propone su representación mediante diagramas de colaboración. Sin embargo no representaría mayores problemas la selección de uno u otro puesto que los diagramas de secuencia y de comunicación son isomorfos [27]:

- Un diagrama de secuencia se puede transformar mecánicamente en un diagrama de comunicación.
- Un diagrama de comunicación se puede transformar automáticamente en un diagrama de secuencia.

A continuación se presentan los Diagramas de colaboración de los CU críticos en el desarrollo del sistema, ver Figuras 2.5 y 2.6. Para ver el resto de los diagramas de clases debe consultar el Anexo D.

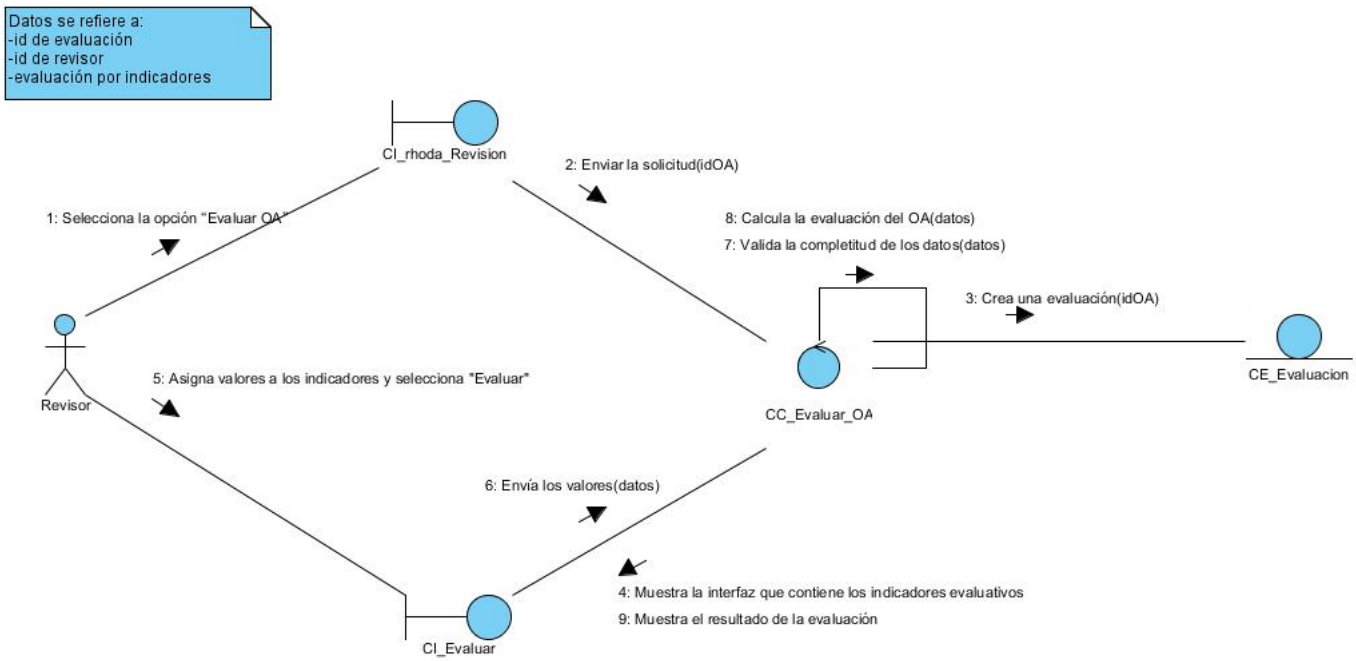

Figura 2.5: Diagrama de colaboración: CU Evaluar OA.

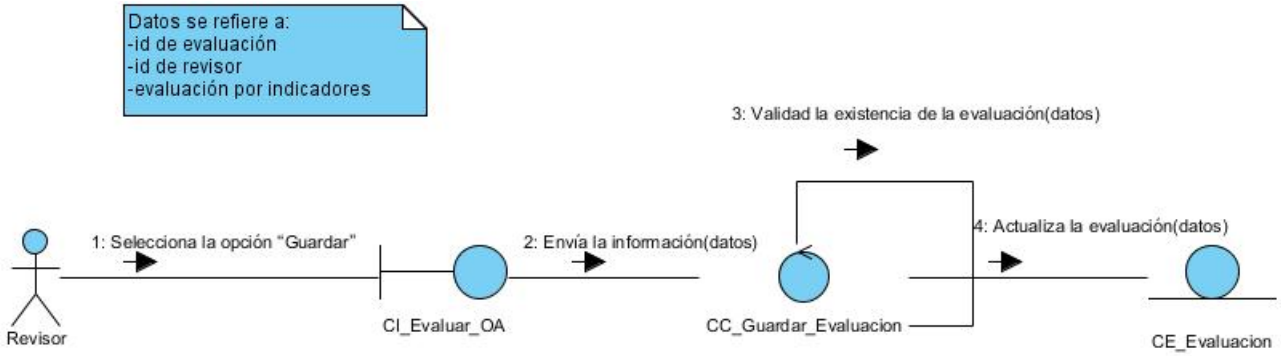

Figura 2.6: Diagrama de colaboración: CU Guardar estado de la evaluación.

2.6. Definición de estilos arquitectónicos

Symfony está basado en un patrón clásico del diseño web, conocido como arquitectura MVC, que separa la lógica del negocio (modelo) y la presentación (vista), consiguiendo aplicaciones más sencillas de mantener [38].

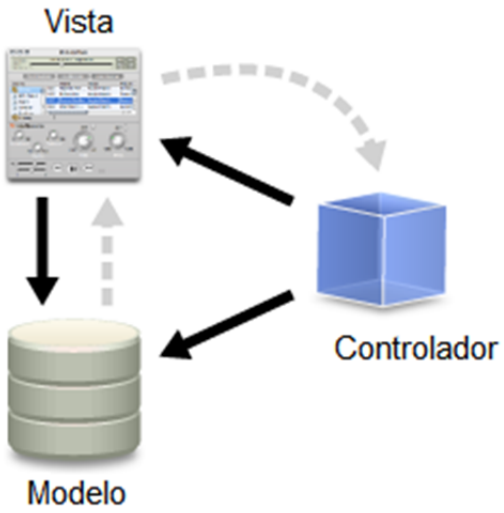

Figura 2.7: Arquitectura Modelo Vista Controlador [30]

El patrón considera tres roles [34]:

- **El modelo**: Representa información relacionada con el dominio de la aplicación.
- **La vista**: Es la representación del modelo en interfaces de usuarios. Solo se encarga de mostrar la información, cualquier cambio es manipulado por el controlador.
- **El controlador**: Interpreta las acciones del teclado o el ratón informando al modelo y/o la vista para que se actualicen según resulte apropiado.

La separación de la vista y el modelo trae ventajas; es posible tener diferentes representaciones de la misma información, haciendo uso del mismo código dentro del modelo. Es posible además programar el código del modelo, abstrayéndose de la representación visual que se le dará a la información.

Symfony toma lo mejor de la arquitectura MVC y la implementa de forma que el desarrollo de aplicaciones sea rápido y sencillo [38]. La capa referente al controlador dentro de la arquitectura MVC, Symfony la divide en un controlador frontal y acciones, ubicando el controlador frontal en el directorio web del proyecto.

La capa del modelo está dividida en dos capas, una para la abstracción del gestor de bases de datos y la otra es la capa de acceso a los datos, con lo que logra aplicaciones independientes a los sistemas gestores de base de datos. Una gran ventaja que tiene Symfony con respecto a muchos otros frameworks de desarrollo radica en el empleo de tareas automatizadas a través de la interfaz de líneas de comandos y el uso de la librería Propel para el Mapeo de Objetos Relacionales (ORM), encargada de la generación automatizada de las clases necesarias para el acceso a los datos, a partir del esquema de la base de datos escrito en Lenguaje de Marcas Extensible (XML) o YAML no es Otro Lenguaje de Marcado (YAML).

Por otra parte la capa de vista, está separada en tres partes: los elementos comunes a todas las páginas de la aplicación, denominada layout; las plantillas, encargadas de visualizar las variables definidas en el controlador; y la lógica de las vistas, definida a través de los slots y componentes (fragmentos de vistas que contienen lógica de la aplicación y pueden ser reutilizables).

2.7. Modelo de Diseño

El Modelo de Diseño es un modelo de objetos que centra su atención en la parte física de nuestro sistema. Se enfoca particularmente en el impacto que provoca en el sistema los requerimientos funcionales y no funcionales para describir la realización física de los CU atendiendo además a otras restricciones que se puedan presentar. Se considera una base para la posterior actividad en el ciclo de vida de desarrollo de software de RUP, la implementación. Es un artefacto considerado de enorme utilidad en esta etapa gracias a que construye una arquitectura estable y sirve de abstracción al modelo de implementación [41].

Una realización de CU–diseño es una colaboración en el Modelo de Diseño que describe cómo se realiza un CU específico, y cómo se ejecuta, en términos de clases de diseño y sus objetos; proporciona por tanto una traza directa a una realización de CU–análisis en el Modelo de Análisis [27];

En este epígrafe se recogerán los elementos más destacados dentro del modelo de diseño como los patrones más relevantes utilizados, el diagrama de clases del diseño y el modelo de datos que se propone con el objetivo de lograr el mejor funcionamiento del módulo.

2.7.1. Definición de patrones de diseño

¿Qué es un patrón de diseño?

Cada patrón, describe un problema que ocurre una y otra vez en nuestro entorno, y describe el núcleo de la solución al problema, de manera que puede utilizarse un millón de veces, sin hacer lo mismo dos veces [31].

Los patrones de diseño resuelven problemas comunes a partir de soluciones probadas mediante un diseño orientado a objeto facilitando la reutilización de clases ya existentes. Dentro de los elementos más importantes que componen un patrón, se destacan el nombre del patrón, el problema que resuelve y la solución al mismo. El framework Symfony implementa una serie de patrones de diseño que hacen que su arquitectura sea suficientemente robusta y a la vez flexible como para adaptarse a los casos más

complejos. A continuación se especifican algunos patrones y ejemplos de su uso dentro de la solución propuesta:

- **Abstract Factory (Fábrica abstracta):** "Proporciona una interfaz para crear familias de objetos que dependen entre sí, sin especificar sus clases concretas" [43]. Esto hace transparente el tipo de familia concreta que se esté utilizando y permite trabajar con objetos de distintas familias de manera que estas no se mezclen entre sí, por ejemplo: cuando el marco de trabajo necesita crear un nuevo objeto, busca en la definición de la factoría el nombre de la clase que se debe utilizar para esta tarea. Se recomienda este patrón cuando se hace necesario adicionar nuevas familias de objetos a un sistema.
- **Singleton (Instancia única):** "Garantiza que una clase sólo tenga una instancia, y proporciona un punto de acceso global a ella" [43], ejemplo de esto es la llamada a la función sfContext:getInstance() en el caso del controlador frontal para garantizar el acceso a la misma instancia. Este patrón establece la creación de una instancia única de un objeto, para ser accedida luego por un método invocado, además es muy flexible al realizar modificaciones para controlar el número de las mismas.
- **Decorator (Envoltorio):** El mismo "añade dinámicamente nuevas responsabilidades a un objeto, proporcionando una alternativa flexible a la herencia para extender la funcionalidad" [43], es decir, cuando se trata de añadir funcionalidades, el patrón proporciona una alternativa a la especialización mediante herencia. Por ejemplo: la plantilla global o el archivo layout.php, contienen el código HTML que es común a todas las páginas de la aplicación para no tener que repetirlo en cada página, integrándose el contenido de la plantilla en el Layout para su decoración.

Patrones General Responsibility Assignment Software Patterns (GRASP)

1. Describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades.
2. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de un objeto esencial.
3. Aplica el razonamiento para el diseño de una forma sistemática, racional y explicable.

Creador: “El patrón Creador aporta un principio general para la creación de objetos, una de las actividades más frecuentes en programación” [42]. Como su nombre lo indica es el que crea y guía la asignación de responsabilidades relacionadas con la creación de objetos, encontrando un creador que debemos conectar con el objeto producido en cualquier evento. Este patrón da soporte al bajo acoplamiento.

En el módulo para la evaluación de la calidad de los OA, la clase actions.class.php es el creador, pues todos los módulos del sistema la contienen para hacer funcional el sistema con sus acciones. Estas últimas se encargan de crear los objetos de las clases que representan las entidades.

Experto: “El patrón Experto posibilita una adecuada asignación de responsabilidades facilitando la comprensión del sistema, su mantenimiento y adaptación a los cambios con reutilización de componentes” [42]. Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada, lo cual es un principio básico que suele ser útil en el diseño orientado a objetos. El uso de este patrón brinda beneficios como la conservación del encapsulamiento y el soporte de un bajo acoplamiento y una alta cohesión.

Propel es una librería externa que utiliza Symfony para realizar una capa de abstracción al modelo de datos, encapsulando toda la lógica de los datos y generando las clases con funcionalidades comunes en las entidades. El patrón Experto se evidencia en la aplicación puesto que cada clase creada por Propel a partir de una entidad es experta en manejar su información.

Controlador: Este patrón es un evento generado por actores externos y se asocia con operaciones del sistema, como son el método de la operación que le corresponde realizar, las respuestas a los eventos del mismo, asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. Además garantiza que las operaciones del sistema se manejen en la capa de dominio de los objetos y no en la de presentación, “sirve como intermediario entre la interfaz de usuario y las clases que encapsulan los datos” [42]. Normalmente un controlador delega en otros objetos el trabajo que se necesita hacer; coordina o controla la actividad.

Las peticiones web, en Symfony, son manejadas por un único punto de entrada de toda la aplicación en un entorno determinado: el controlador frontal (frontend.php). Este utiliza el sistema de enrutamiento para asociar el nombre de una acción y el nombre de un módulo con la URL entrada por el usuario cuando recibe una petición.

Alta Cohesión: “El patrón Alta Cohesión es una medida que determina cuán relacionadas y adecuadas están las responsabilidades de una clase, de manera que no realice un trabajo colosal; una clase con baja cohesión realiza un trabajo excesivo, haciéndola difícil de comprender, reutilizar y conservar” [42]. Cada elemento del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable. Este patrón mejora la claridad y facilidad con que se entiende el diseño, simplifica el mantenimiento y las mejoras de funcionalidad, genera un bajo acoplamiento y soporta mayor capacidad de reutilización.

El marco de trabajo Symfony permite la asignación de responsabilidades con alta cohesión. Un ejemplo de esto es la colaboración de la clase actions.class.php con otras clases para realizar diferentes operaciones y crear objetos, la misma tiene la responsabilidad de definir las acciones sobre las plantillas y proporciona al software flexibilidad frente a grandes cambios por poseer una estrecha relación en sus funcionalidades.

Bajo Acoplamiento: “El patrón Bajo Acoplamiento es una medida de la fuerza con que una clase se relaciona con otras, porque las conoce y recurre a ellas; una clase con bajo acoplamiento no depende de muchas otras, mientras que otra con alto acoplamiento presenta varios inconvenientes: es difícil entender cuando está aislada, es ardua de reutilizar porque requiere la presencia de otras clases con las que esté conectada y es cambiante a nivel local cuando se modifican las clases a fines” [42]. Este patrón es un principio que asigna la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, etc.). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión.

Para lograr un bajo acoplamiento de clases en Symfony, la clase actions.class.php hereda solamente de sfActions.

2.7.2. Diagramas de clases del diseño

Los diagramas de clases del diseño representan la interacción de las clases de diseño y sus objetos en la realización de los CU, de forma que de cada CU se desprende un diagrama de clases. Para el sistema que se está proponiendo se presenta a continuación en las Figuras 2.8 y 2.9 el diagrama de clases del diseño con estereotipos web correspondientes a los CU críticos del sistema. El concepto de estereotipos

web no cambia en lo absoluto nada de lo mencionado hasta el momento sino que especifica que se está desarrollando un sistema cuyas prestaciones se darán vía web. Para ver el resto de los digramas consultar el Anexo E.

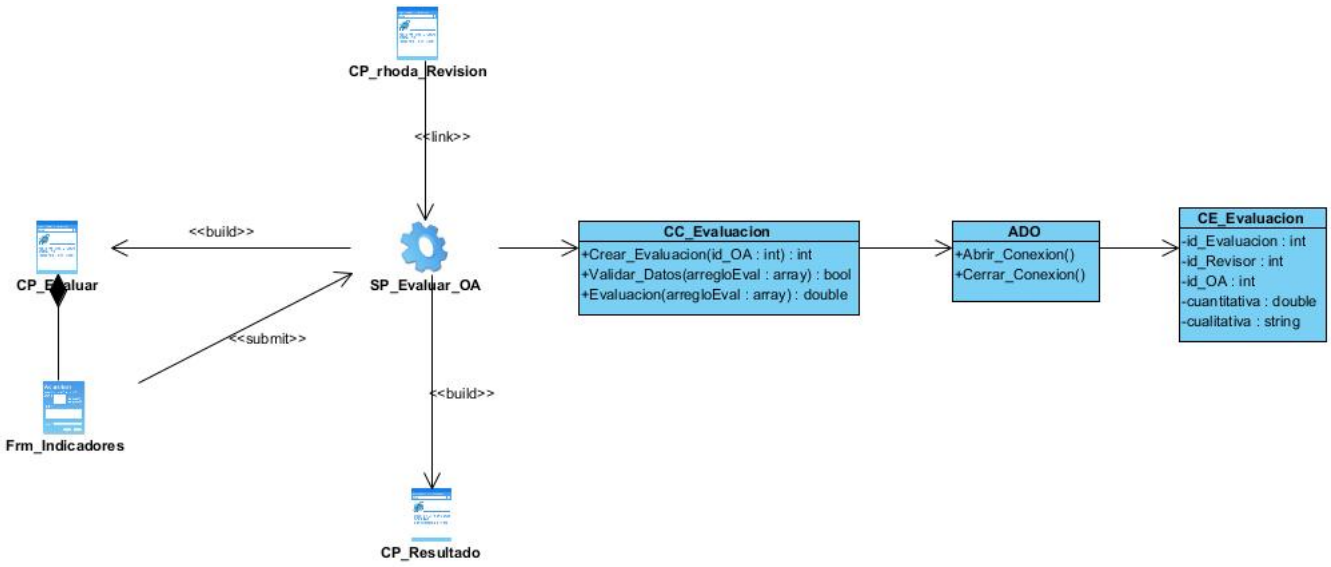

Figura 2.8: Diagrama de clases del diseño: CU Evaluar OA.

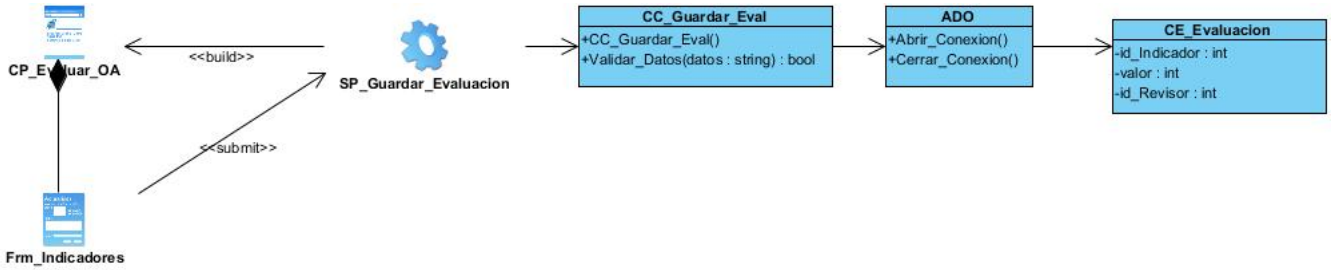

Figura 2.9: Diagrama de clases del diseño: CU Guardar estado de la evaluación.

2.8. Modelo de datos

El modelo de datos es la descripción básica de un contenedor de datos en el cual se describen las tablas, relaciones y métodos que se utilizan para almacenar la información en el contenedor. Los modelos de datos no son elementos físicos, son abstracciones que permiten la implementación de un sistema

eficiente de base de datos [44].

Se presenta en la Figura 2.10 el Diagrama entidad relación definido para describir la representación lógica y física de la información persistente que se maneja en el sistema, se describen los elementos que intervienen en el módulo de evaluación de la calidad de los OA y la forma en que estos se relacionan.

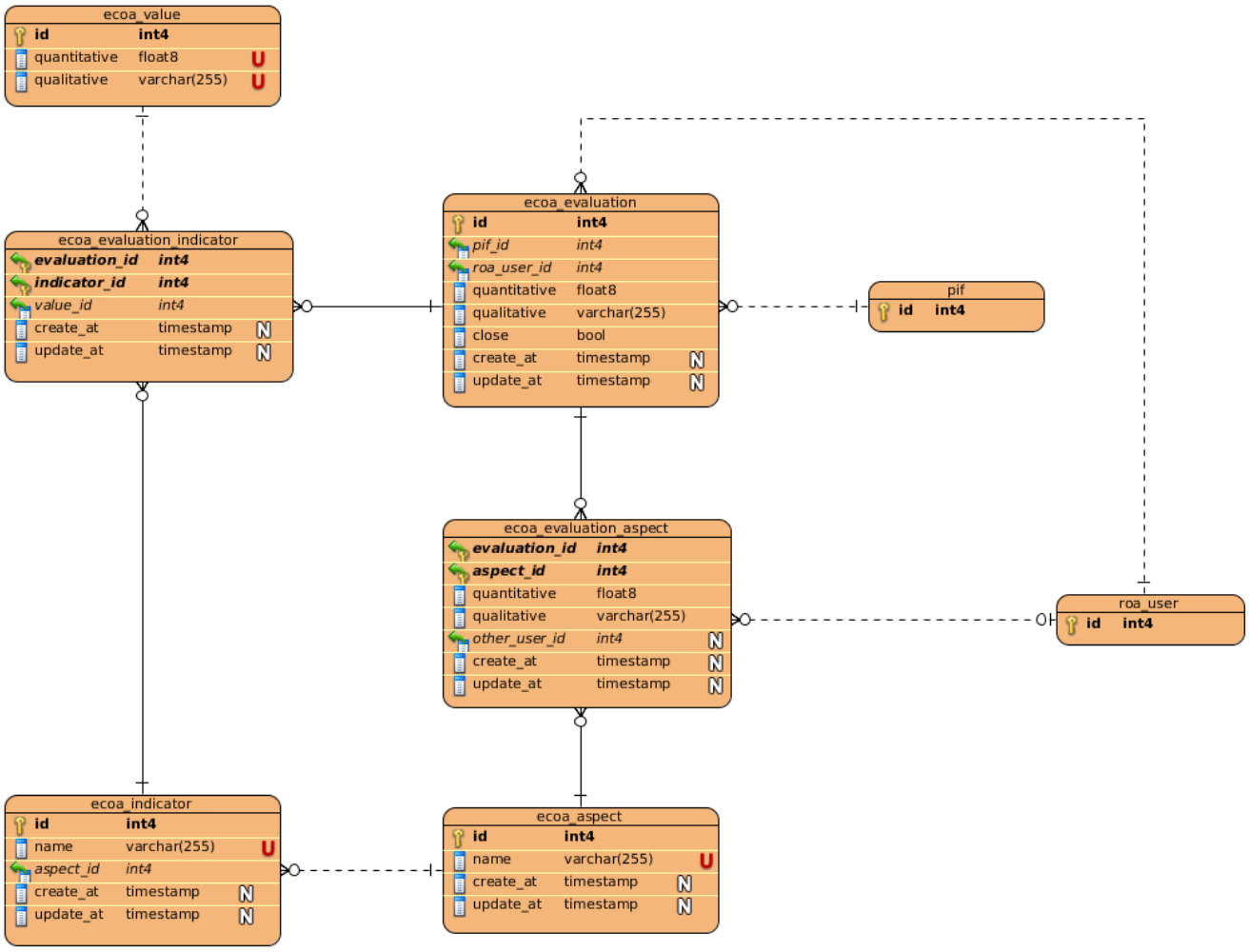

Figura 2.10: Diagrama Entidad-Relación

Conclusiones

Hasta el momento, la investigación realizada ha llegado al punto donde la utilización de RUP ha guiado satisfactoriamente el desarrollo del software, encontrándose este, listo para la implementación. Se

siguió un conjunto de pasos para obtener una concepción del sistema que cumpla con los objetivos y expectativas marcados. De forma tal que se incluyen en esta sección los artefactos generados en correspondencia con los flujos de trabajo desarrollados. Por tanto, se cuenta actualmente con el modelo de dominio del sistema. Más adelante quedan recogidos los requerimientos identificados, los cuáles constituyen una base para la identificación de los CU. Se genera el diagrama de CU y se describe cada CU para lograr comprensión absoluta de los mismos. En el modelo de análisis y de diseño se generan sus respectivos diagramas de clases y quedan estos explícitamente modelados. También quedó modelado el diagrama entidad relación a partir del cual quedará constituida la base de datos del sistema. Y no menos relevante se expone un resumen de los estilos arquitectónicos y patrones de diseño utilizados.

Capítulo 3

Implementación y pruebas

Introducción

A continuación queda reflejado el flujo de actividades que conforman la implementación y las pruebas del sistema. Esta vez el desarrollo del capítulo estará dado en dos momentos fundamentales. Por tanto, se presenta organizado de la siguiente manera.

Implementación, donde se relejará la distribución del los nodos físicos, especificando protocolos de comunicación y sus relaciones en un diagrama. También se reflejan las relaciones entre los componentes que conforman la aplicación, como resultado de traducir modelos generados en fases previas.

Pruebas, estas adquieren un elevado nivel de importancia debido a que en esta etapa se prueba el software. El objetivo que se persigue aquí, es descubrir y corregir el mayor número de errores antes de que comiencen las pruebas del cliente, ya que cada vez que este usa el software está llevando a cabo una prueba. Se generan los artefactos que resultan de aplicar al sistema pruebas de caja negra.

3.1. Modelo de implementación

El modelo de implementación describe cómo los elementos del modelo de diseño, como las clases, se implementan en términos de componentes, como ficheros de código fuente, ejecutables, etc. El modelo de implementación describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados y cómo dependen los componentes unos de otros [27].

3.1.1. Modelo de despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. El modelo de despliegue se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño [27].

Podemos observar lo siguiente en el modelo de despliegue [27]:

- Cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar.
- Los nodos poseen relaciones que representan medios de comunicación entre ellos, tales como Internet, intranet, bus y similares.
- El modelo de despliegue puede describir diferentes configuraciones de red, incluidas las configuraciones para pruebas y para simulación.
- La funcionalidad (los procesos) de un nodo se definen por los componentes que se distribuyen sobre ese nodo.
- El modelo de despliegue en sí mismo representa una correspondencia entre la arquitectura de software y la arquitectura del sistema (el hardware).

3.1.2. Diagrama de despliegue

Para Craig Larman en su libro UML y patrones “Los diagramas de despliegue muestran a los nodos procesadores la distribución de los procesos y de los componentes [28].

A continuación en la Figura 3.1 se puede observar el diagrama de despliegue del módulo para la evaluación de la calidad de los OA. En él se exhibe la disposición física de los nodos vitales para el sistema en tiempo de ejecución y los componentes que radican en cada uno de ellos. Por otro lado es posible apreciar los links comunicación de los nodos, los cuáles se presentan estereotipados con sus respectivos protocolos de comunicación.

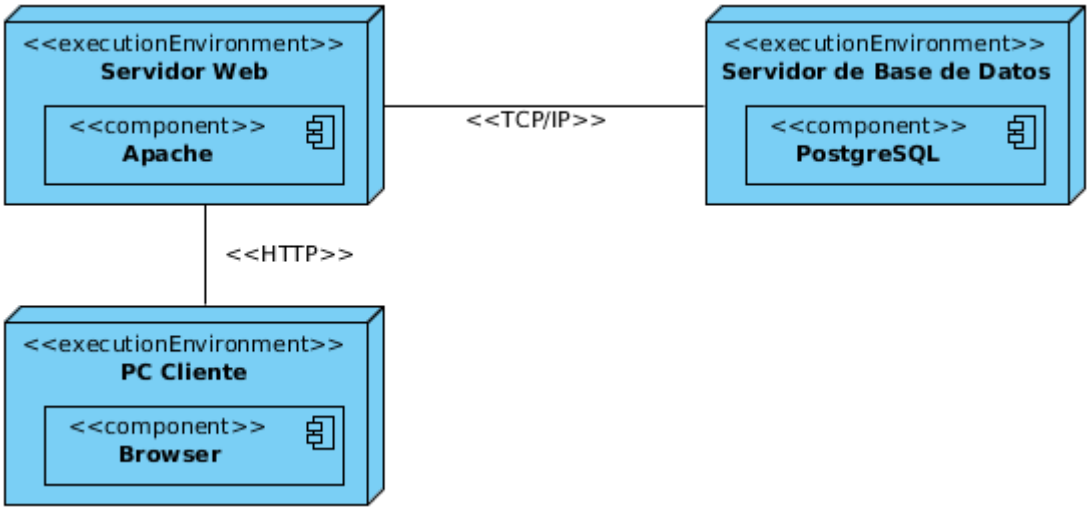

Figura 3.1: Diagrama de Despliegue

Por tanto, cualquier usuario que desee acceder al sistema debe servirse de una computadora. A continuación, haciendo uso del navegador se conectará al servidor web de código abierto A Patchy Server, Apache a través del protocolo seguro Hypertext Transfer Protocol (HTTP). En el servidor Apache se mostrará el repositorio que tendrá incorporado el módulo de evaluación de la calidad de los OA. De aquí en adelante se establece la conexión a través del protocolo Transmission Control Protocol/Internet Protocol TCP/IP hacia el servidor de base de datos PostgresSQL.

3.1.3. Diagrama de componentes

En el mismo libro anteriormente mencionado, su autor, Craig Larman, de hace alusión a los diagramas de componentes exponiendo que “Los diagramas de componentes muestran las dependencias del compilador y del “runtime” entre los componentes del software; por ejemplo los archivos de código fuente y los DLL”. En las Figuras 3.2 y 3.3 se modela los diagramas de componentes de los CU críticos de la aplicación. Para ver el resto de los diagramas consultar el Anexo F.

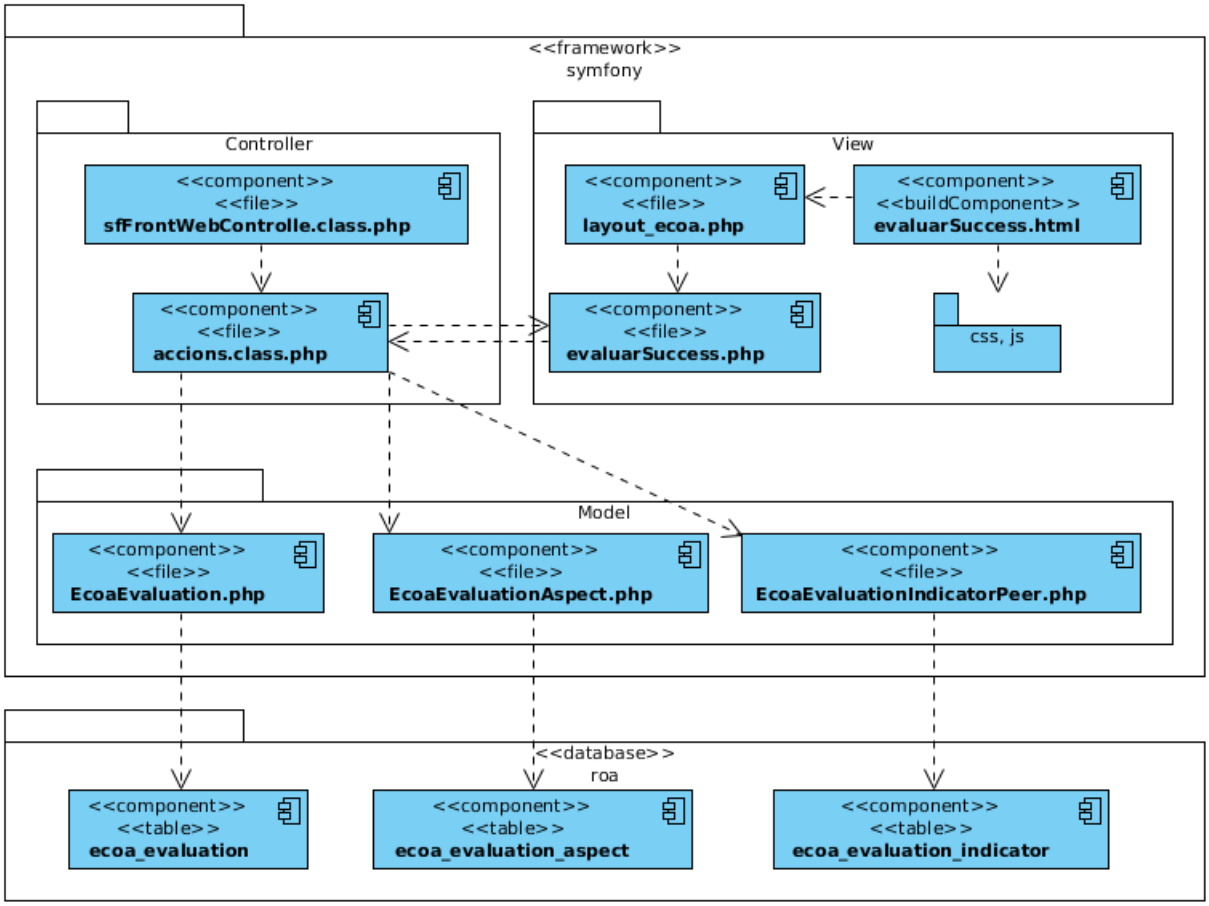

Figura 3.2: Diagrama de componentes: CU Evaluar OA

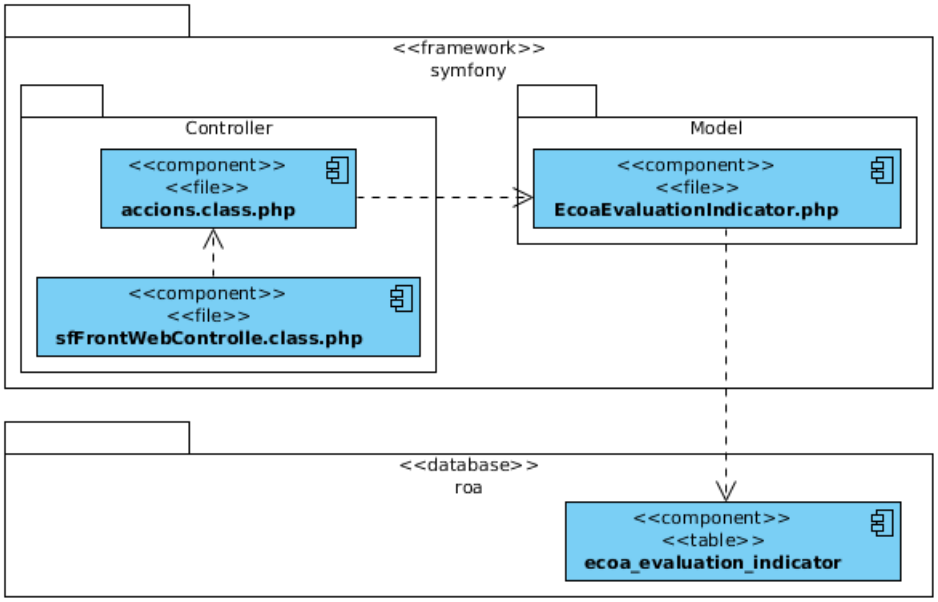

Figura 3.3: Diagrama de componentes: CU Guardar estado de la evaluación

3.2. Pruebas de software

Las pruebas de software son un elemento crítico para la garantía de la calidad del software y representan una revisión final de las especificaciones, del diseño y de la codificación. Una vez generado el código fuente, el software debe ser probado para describir (y corregir) el máximo de errores posibles antes de su entrega al cliente [29].

Mediante las pruebas de software se pretende cambiar el punto de vista hacia el módulo de evaluación de la calidad de los OA que se propuso y que se ha vendido desarrollando a lo largo del proceso de desarrollo de software. O sea, proyectar los próximos esfuerzos a la desestabilización del propio sistema, de forma tal que se logre encontrar el mayor número posible de defectos. De esta forma, las posibilidades de localizar problemas en el sistema para su control y solución es mucho mayor. Y así entregar al cliente un sistema más robusto. Condición que sería mucho menor que si se prescindiera de las pruebas.

Nivel de prueba

Para la realización de las pruebas al módulo desarrollado se realizó un análisis para determinar la que más se ajustara a las necesidades existentes. De los cinco niveles de prueba existentes, unidad, integración, sistema, aceptación y liberación, se escogió el de sistema. Este nivel de pruebas de sistema se lleva a cabo durante la construcción del sistema. Su objetivo primordial es probar a fondo el sistema comprobando su funcionalidad e integridad globalmente, en un entorno lo más parecido posible al entorno final de producción. Para ello se prueba el sistema como lo hará el usuario final [45].

Tipo de prueba

Dentro de los tipos de prueba se seleccionó las pruebas de funcionalidad por sobre otras como: controles, esfuerzo, volumen etc. Para ello se tomó en consideración el hecho de que este tipo de pruebas examina si el sistema cubre sus necesidades de funcionamiento, acorde a las especificaciones de diseño. En ellas se debe verificar si el sistema lleva a cabo correctamente todas las funciones requeridas, se debe verificar la validación de los datos y se deben realizar pruebas de comportamiento ante distintos escenarios [46].

Técnica de prueba

Se selecciona la aplicación de la técnica de caja negra debido a que estas se enfocan en los requerimientos establecidos y en la funcionalidad del sistema. Y no considera la codificación dentro de sus parámetros a evaluar, ya que no están basadas en el conocimiento del diseño interno del programa. Para ello toda la atención se enfoca en suministrar datos como entrada y estudiar su respectiva salida, despreciando lo que pueda estar haciendo el módulo por dentro en su implementación. Se trata de un enfoque que intenta descubrir diferentes tipos de errores que no se encuentran con los métodos de caja blanca [46, 47].

Método de prueba

Existen varias técnicas para realizar las pruebas de caja negra como: el Análisis de Valores Límites, Grafos de Causa-Efecto y la Partición de Equivalencia de la cual se hará uso durante esta etapa. Se hace la selección de esta técnica debido a que es considerada muy efectiva ya que permite el análisis de datos válidos e inválidos para las entradas existentes en el software. El diseño de casos de prueba para la partición equivalente se basa en una calificación de las clases de equivalencia para una las condiciones de entrada existentes. Una clase de equivalencia representa un conjunto de estados válidos o inválidos para estas condiciones [47].

3.2.1. Diseño de casos de prueba

Un caso de prueba específica una forma de probar el sistema, incluyendo la entrada o resultado con la que se ha de probar y las condiciones bajo las que ha de probarse. Un caso de prueba puede derivarse de, y por tanto debe seguir la traza de, un caso de uso en el modelo de casos de uso o de una realización de caso de uso en el modelo de diseño [27]. A continuación se presentan los diseños de casos de prueba para los CU críticos. Para ver el resto consultar el Anexo D.

Diseño de Caso de Prueba para el CU Evaluar OA

Descripción general

El CU inicia cuando el revisor selecciona evaluar un OA, luego realiza la evaluación y el sistema calcula el resultado.

Condiciones de ejecución

1. El revisor se ha autenticado en el sistema.
2. El revisor ha sido asignado a evaluar el OA.

Tabla 3.1: Caso de Prueba: CU Evaluar OA

Escenario	Descripción	Evaluación	Respuesta del sistema	Flujo central
EC 1.1 Evaluar OA satisfactoriamente	Permite evaluar un OA	V Exelente V Bien V Regular V Mal	Muestra una interfaz con el resultado obtenido de la evaluación general y por aspectos	Acceder a la sección de revisión del repositorio. Seleccionar un OA para evaluar. Asigna valores a todos los indicadores. Seleccionar la opción “Evaluar”
EC 1.2 Regresar a la evaluación	Permite regresar a la interfaz que contiene los indicadores evaluativos	NA	Muestra al revisor la interfaz que contiene los indicadores evaluativos	Acceder a la sección de revisión del repositorio. Seleccionar un OA para evalúa. Asignar valores a todos los indicadores. Seleccionar la opción “Evaluar”. Seleccionar la opción “Volver atrás”

EC 1.3 Salir de la evaluación	Permite salir de la evaluación sin tener que registrarla antes	NA	Redirecciona al revisor a la sección de revisión del repositorio	Acceder a la sección de revisión del repositorio. Seleccionar un OA para evaluar. Asignar valores a todos los indicadores. Seleccionar la opción “Evaluar”. Seleccionar la opción “Salir”.
EC 1.4 Faltan indicadores por evaluar	Permite regresar a la interfaz que contiene los indicadores evaluativos para completar la evaluación	NA	Comprueba que restan indicadores por evaluar, notifica el error y redirecciona al revisor a la interfaz que contiene los indicadores evaluativos.	Acceder a la sección de revisión del repositorio. Seleccionar un OA para evaluar. Asigna valores a todos los indicadores. Seleccionar la opción “Evaluar”.

Tabla 3.2: Variables del Caso de Prueba 3.1

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Evaluación	Lista desplegable	No	Evaluación que se le otorga al OA, según los indicadores por evaluar propuestos por el instrumento de evaluación utilizado, (Obligatorio).

Diseño de Caso de Prueba para el CU Guardar estado de la evaluación

Descripción general

El CU inicia cuando se está evaluando un OA, seleccionándose luego la opción “Guardar” para registrar las últimas modificaciones.

Condiciones de ejecución

1. El revisor se ha autenticado en el sistema.
2. El revisor ha sido asignado a evaluar el OA.
3. El revisor ha seleccionado la opción “Evaluar” del OA que se encuentra en el repositorio.

Tabla 3.3: Caso de Prueba: CU Guardar estado de la evaluación

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Guardar estado de la evaluación exitosamente.	Permite guardar el estado de la evaluación del OA.	Se almacena en la base de datos el estado de la evaluación del OA.	Acceder a la sección de revisión del repositorio. Seleccionar un OA para evaluar. Asignar valores a todos los indicadores. Seleccionar la opción “Guardar”.

3.2.2. Resultados de las pruebas

Una evaluación de prueba es una evaluación de los resultados de los esfuerzos de prueba, tales como la cobertura del caso de prueba, la cobertura del código y el estado de los defectos [27].

Una semana fue el plazo establecido para la realización de las pruebas al software, para ello se propor-

cionaron instrucciones precisas de cómo realizarlas al personal involucrado. Se explicó detalladamente el objetivo de esta importante actividad y los pasos a seguir para llevarla a cabo con calidad. Una vez vencido el plazo se obtuvieron los siguientes resultados:

A continuación se presenta el desglose de las no conformidades encontradas por CU. Las técnicas de prueba de Caja Negra que se llevaron a cabo arrojaron un total de 10 no conformidades en la primera iteración. Si se observa la Figura 3.4 se puede observar como el CU 1 (Evaluar OA) es en el que se detectaron las no conformidades más significativas, las cuáles podían comprometer el correcto funcionamiento del módulo de evaluación. Por otra parte en las pruebas a los CU 5, 8 y 9 (Restaurar instrumento de evaluación, Gestionar aspecto, Gestionar indicador) se plasmaron recomendaciones enfocadas principalmente al perfeccionamiento del diseño. Por último, la mayor cantidad de no conformidades encontradas se clasificaron como no significativas y estas estuvieron dirigidas principalmente al mejoramiento de la comunicación del usuario con el módulo.

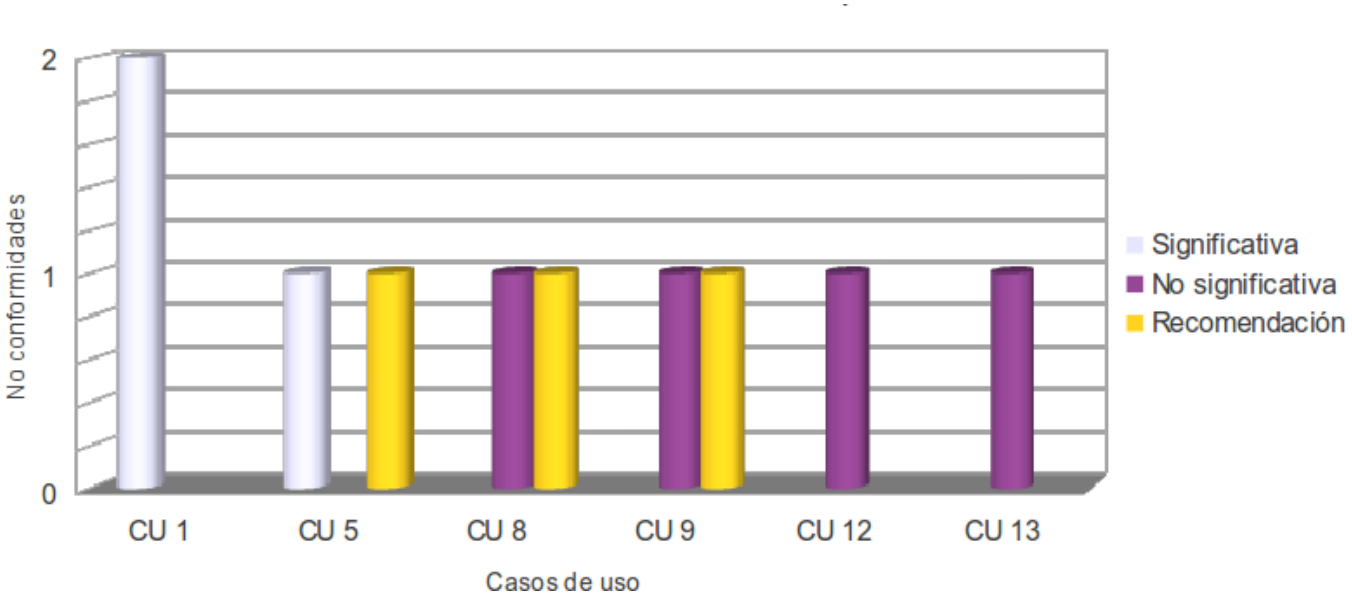

Figura 3.4: No conformidades detectadas por CU

Posterior a esta iteración se estableció en conjunto con los desarrolladores un plazo de 72 horas para la resolución de las no conformidades detectadas y el comienzo de las segunda iteración de pruebas. Para el término de esta iteración las no conformidades detectadas con anterioridad se corrigieron correctamente y los responsables de las pruebas no encontraron nuevas fallas.

Finalizando este ciclo se le pidió a un grupo de usuarios ajenos al sistema que navegaran por el mismo con el objetivo de descubrir debilidades y fortalezas del mismo. Para ello se seleccionó un grupo de atributos, ellos son: (Amigable, Legibilidad, Eficiencia, Satisfacción, Reversibilidad, Interfaz gráfica, Facilidad de acceso). Luego de utilizar el sistema los usuario seleccionados debía calificarlo atendiendo a estos atributos y atribuyendo una evaluación del 1 al 5, uno para la más baja y cinco para la más alta.

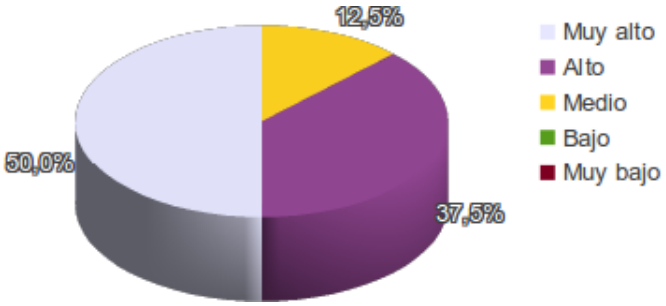

Figura 3.5: Amigable

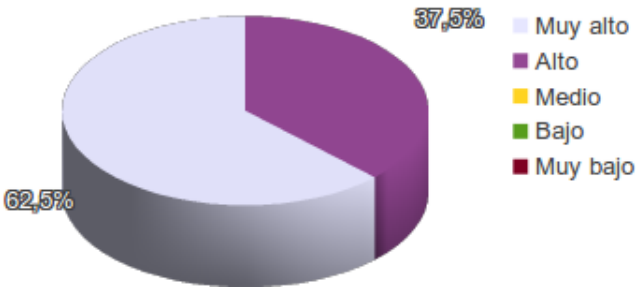

Figura 3.6: Legibilidad

Amigable: Se mide la facilidad de interacción del sistema con el usuario sin tener que consultar un manual o ayuda en línea. Ver Figura 3.5.

Legibilidad: Se mide el color de los textos, el contraste de los mismos con el fondo, el tipo de fuente y el tamaño de la misma, que debe ser legible por la mayoría de los usuarios. Ver Figura 3.6.

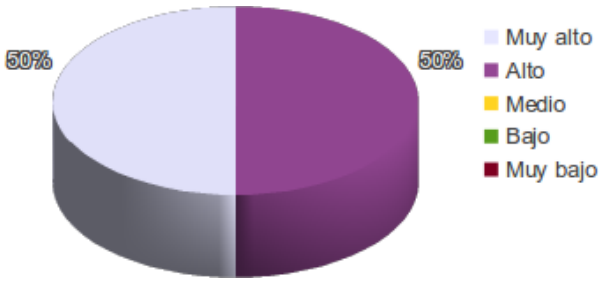

Figura 3.7: Eficiencia

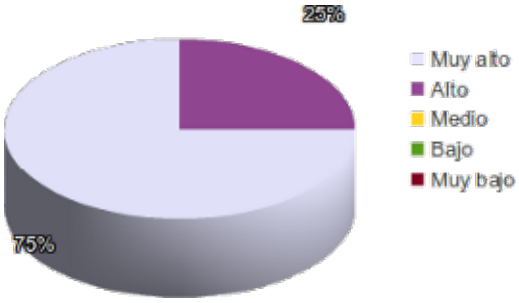

Figura 3.8: Reversibilidad

Eficiencia: Se mide si las tareas que se llevan a cabo, pueden ser realizadas rápida y fácilmente. Ver Figura 3.7.

Reversibilidad: Se mide la capacidad del sistema para deshacer las acciones realizadas. Ver Figura 3.8.

Interfaz gráfica: Se mide que tan placentero resulta la navegación en el sistema gracias a la interfaz

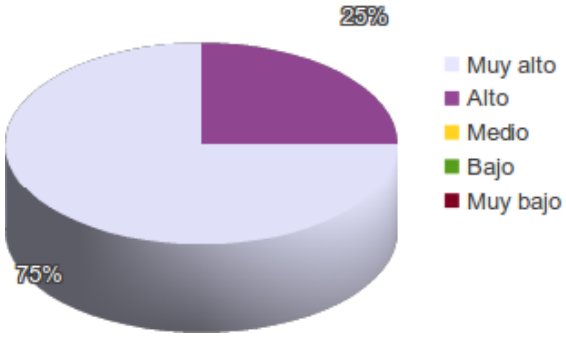

Figura 3.9: Interfaz gráfica

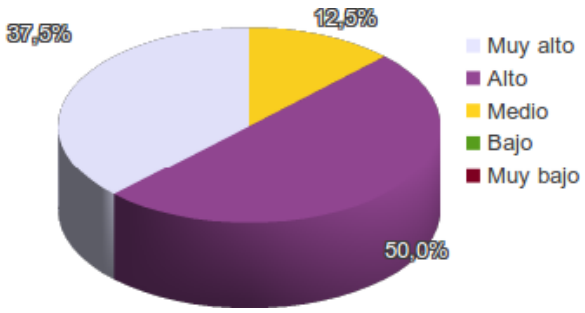

Figura 3.10: Acceso al sistema

gráfica. Esto incluye las imágenes, colores y disposición de los elementos que conforman el sistema. Ver Figura 3.9.

Acceso al sistema: Se mide la facilidad con la que se puede acceder al sistema. Ver Figura 3.10.

Como se observa en las gráficas anteriores, la aplicación de la encuesta arrojó que los usuarios consideran que de forma general el módulo posee un alto nivel de acuerdo a los atributos planteados. Y se destacan entre ellos la legibilidad, reversibilidad e interfaz gráfica las cuales resultaron de gran aceptación.

Conclusiones

Al finalizar el desarrollo de este capítulo se obtuvo un módulo completamente funcional. En este punto se encuentran implementados los elementos del diseño cumpliendo satisfactoriamente con los requerimientos definidos. Se generaron nuevos artefactos como los diagramas de componentes y el modelo de despliegue.

Se integró el módulo para la evaluación de la calidad de los OA al repositorio y se hizo la selección de las pruebas a aplicar. Se corrigieron las no conformidades detectadas durante las pruebas las cuales restaban calidad al módulo desarrollado. Finalmente, se logró identificar las principales fortalezas a partir del análisis de los resultados arrojados de la aplicación de encuestas a usuarios con diferentes niveles de conocimiento en la temática desarrollada.

Conclusiones

Luego de analizar la posibilidad de utilizar alguna solución existente a nivel nacional e internacional para dar respuesta a la problemática planteada, se decidió desarrollar el módulo para la evaluación de la calidad de los OA. Al concluir el desarrollo del mismo se plantean las siguientes conclusiones:

- Se logró con el estudio de las herramientas para automatizar el proceso de evaluación de los OA identificar las principales funcionalidades para el módulo desarrollado.
- Se realizó el diseño e implementación del módulo para la evaluación de la calidad de los OA a partir de los requerimientos funcionales y no funcionales identificados.
- Se integró el módulo desarrollado con el RHODA para el apoyo a las evaluaciones de la calidad de los OA que en él se almacenan.
- A partir de las pruebas de funcionalidad usando la técnica de caja negra a las que se sometió el módulo, se encontraron 10 no conformidades que fueron resueltas por los desarrolladores del mismo.

Ante los elementos anteriores puede afirmarse que se cumplieron los objetivos propuestos y se verificó la validez de la idea a defender, lo cual se materializa en el módulo obtenido. Actualmente el módulo está integrado al repositorio de la universidad sirviendo de apoyo a los revisores para evaluar la calidad de los OA.

Recomendaciones

Aunque la concepción del módulo permite integrarse con cualquier repositorio que cumpla con los estándares SCORM y LOM; se recomienda una aplicación que genere automáticamente en los repositorios la estructura de datos necesaria para el funcionamiento del módulo.

El módulo permite exportar a los formatos (.csv y . html) los resultados de las evaluaciones. Se recomienda implementar la funcionalidad de importar.

Lista de acrónimos

AGORA Asistencia para la Gestión de Objetos Reusables de Aprendizaje

AKHME Adaptive Hypermedia Knowledge Management E-learning Platform

CAREO Campus Alberta Repository of Educational Objects

CASE Ingeniería de Software Asistida por Computadora

CU Caso de Uso

CUDI Corporación Universitaria para el Desarrollo de Internet

EVA Entorno Virtual de Aprendizaje

GRASP General Responsibility Assignment Software Patterns:
Patrones Generales de Asignación de Responsabilidad de Software

HEODAR Herramienta de Objetos Didácticos de Aprendizaje Reutilizables

IDE Entorno de Desarrollo Integrado

IEEE Instituto de Ingenieros Eléctricos y Electrónicos

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

ISW Ingeniería de Software

LMS Learning Management System:
Traducción al español: Sistema de Gestión de Aprendizaje, software basado en un servidor Web que provee módulos para los procesos administrativos y de seguimiento que se requieren para un sistema de enseñanza aprendizaje

LOM Learning Object Metadata

LORI Learning Object Review Instrument

MECOA Modelo para la Evaluación de la Calidad en Objetos de Aprendizaje

MERLOT Multimedia Educational Resource for Learning and Online Teaching

MVC Modelo Vista Controlador

OA Objeto de Aprendizaje

ORM Mapeo de Objetos Relacionales

PHP Hypertext Preprocessor

ROA Repositorios de Objetos de Aprendizaje

RUP Rational Unified Process

SCORM Sharable Content Object Reference Model

SMETE Science, Mathematics, Engineering and Technology Education

TIC Tecnologías de la Información y las Comunicaciones

UCI Universidad de las Ciencias Informáticas

UML Unified Modeling Language

URL Uniform Resource Locator

XML Lenguaje de Marcas Extensible

YAML YAML no es Otro Lenguaje de Marcado

Referencias Bibliográficas

[1] J. Rosario. La tecnología de la información y la comunicación (tic). su uso como herramienta para el fortalecimiento y el desarrollo de la educación virtual. *Disponible en el ARCHIVO del Observatorio para la CiberSociedad, Recuperado el*, 17, 2005.

[2] M. Chan. Objetos de aprendizaje: una herramienta para la innovación educativa. *Revista Apertura, Innova. Universidad de Guadalajara*, 2002.

[3] R. Morales and A.S. Agüera. Capacitación basada en objetos reusables de aprendizaje. *Boletín IIE*, 26(1):23, 2002.

[4] D.A. Wiley. Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. *Learning Technology*, 2830(435):1–35, 2000.

[5] L.A. Poveda and FPU Becaria. Objeto de aprendizaje.

[6] C. Velázquez, J. Muñoz, F. Êlvarez, and C. Arevalo. La importancia de la definición de la calidad del contenido de un objeto de aprendizaje. *Avances en Ciencias de la Computación (ENCTOA 2005)*, pages 329–333, 2005.

[7] E. Morales, F. García, A. Barrón, A. Berlanga, and C. López. Propuesta de evaluación de objetos de aprendizaje. In *II Simposio Pluridisciplinar Sobre Diseño, Evaluación y Descripción de Contenidos Educativos (SPEDECE)*, 2005.

[8] Y.C. Toll. Guía de evaluación de la calidad de los objetos de aprendizaje producidos en la universidad de las ciencias informáticas. Master's thesis, Universidad de la Ciencias Informáticas, 2011.

[9] Corporación Universitaria para el Desarrollo de Internet A. C. Comité académico de objetos de aprendizaje, 2002. Integrantes de la Comisión: Lourdes Galeana, Universidad de Colima,María Elena Chan, Universidad de Guadalajara y Marisol Ramírez, ITESM.

[10] L. Rodríguez. Análisis y diseño de la versión 3.0 de rhoda. Master's thesis, Universidad de las Ciencias Informáticas, 2011.

[11] M.E. Del Moral and D.A. Cernea. Diseñando objetos de aprendizaje como facilitadores de la construcción del conocimiento. In *Actas del II Simposio Pluridisciplinar sobre Diseño, Evaluación y Descripción de Contenidos Educativos Reutilizables, SPDECE*, volume 5, 2005.

[12] R. Ruíz, J. Muñoz, and F. Álvarez. Formato para la determinación de la calidad de los objetos de aprendizaje. 10, 2008.

[13] Y.C.P. Toll, L.O. Ruiz, Y.C. Trujillo, and Y.G Ril. La calidad de lo objetos de aprendizaje producidos en la universidad de las ciencias informáticas. In *Décima Edición de la Semana Tecnológica*, 2010.

[14] D.T. Ávila. Elementos fundamentales sobre la concepción de los objetos de aprendizaje en la uci. *Sesión Científica de la Dirección de Teleformación*, 2009.

[15] P. Higgs, S. Meredith, and T. Hand. Technology for sharing: Researching learning objects and digital rights management. *Flexible Learning Leader Report*, 2003, 2002.

[16] T. Looms and C. Christensen. Emerging and enabling technologies for the design of learning object repositories report. *Advanced Distributed Learning*, 2002.

[17] JORUM+ Project. The jisc online repository for [learning and teaching] materials. 2004.

[18] EduTools. Compare management systems. 2005.

[19] E. Castro. Estándares en los sistemas de gestión de aprendizaje. *Instituto Politécnico Nacional, Dirección de Tecnología Educativa. Biblioteca Nacional de Ciencia y Tecnología*, 2002.

[20] C. Guzmán. *Los Repositorios de Objetos de Aprendizaje como soporte a un entorno e-learning*. PhD thesis, Tesis doctoral, Universidad de Salamanca, 2005.

[21] A.L. Etxeberría and España) Asociación Nacional de Centros de E-Learning y Distancia (Madrid. *Buenas prácticas de e-learning*. Asociación Nacional de Centros de E-Learning y Distancia, 2007.

[22] Dr. Francisco José García Peñalvo. Dos temas controvertidos en eLearning: objetos de aprendizaje y calidad. http://www.learningreview.es/e-learning-279/articulos-elearning/447-dos-temas-controvertidos-en-elearning-objetos-de-aprendizaje-y-calidad, 2011.

[23] JC Nesbit, K. Belfer, and T. Leacock. Learning object review instrument (lori). *E-learning research and assessment network*, 2003.

[24] C.L. Vidal, A.A. Segura, and M.E. Prieto. Calidad en objetos de aprendizaje. In *V Simposio Pluridisciplinar sobre Diseño y Evaluación de Contenidos Educativos Reutilizables (SPDECE 2008), Salamanca*, 2008.

[25] M.P.P. Espinosa, F.M. Sánchez, and I.G. Porlán. Producción de materiales didacticos: Los objetos de aprendizaje. 2008.

[26] Modelo de dominio. *Tecnología y Synergix*.

[27] I. Jacobson, G. Booch, and J. Rumbaugh. El proceso unificado de desarrollo de software. 2000.

[28] C. Larma. Uml y patrones. *Introducción al análisis y diseño orientado a objetos. I Edición. Editorial Prentice Hall. México*, 1999.

[29] R.S. PRESSMAN. Ingeniería del software un enfoque practico. quinta edición. madrid. macgraw-hill, 2002.

[30] Aitor Rigada. Model view controller. *Aitor Rigada*, October 2008.

[31] C. Alexander and S. Ishikawa. M. silverstein with m. jacobson, i. fiksd ahl-king and s. angel, a pattern language, 1977.

[32] Dra. Anaisa Hernández González. Curso de UML for actividad2 diagrama de casos de uso del negocio y del sistema.

[33] Flujo de trabajo análisis y diseño - EcuRed. http://www.ecured.cu/index.php/Flujo_de_Trabajo_An%C3%A1lis

[34] M. Fowler. *Patterns of enterprise application architecture*. Addison-Wesley Professional, 2003.

[35] A. September. Ieee standard glossary of software engineering terminology. *Office*, 121990(1), 1990.

[36] Apuntes. ingenieria del software. sistemas informaticos. nivel de madurez sofware. informatica aplicada a la gestión pública. 2005/06-2. universidad de murcia. rafael barzanallana.

[37] A.E. de Cuba. Oficina nacional de estadísticas. *Varios años, La Habana*, 1997.

[38] F. Potencier and F. Zaninotto. Symfony la guía definitiva. *Apress. Año*, 2007.

[39] I. Sommerville. Software engieniering. octava edición. ed, 2007.

[40] A. Weitzenfeld. *Ingeniería de software orientada a objetos con Java e Internet*. Cengage Learning Latin America, 2005.

[41] N. Rojas and Y. Hecheverría. Sistema para la gestión de los eventos científicos en la universidad de las ciencias informáticas. Master's thesis, Universidad de las Ciencias Informáticas, 2008-2009.

[42] R. Botero Tabares. Patrones grasp y anti-patrones: Un enfoque orientado a objetos desde la lógica de programación. *Entre ciencia e ingeniería*, (8):161–173, 2011.

[43] J. Gracia. Patrones de diseño. análisis y diseño. ingenieria del software. http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php.

[44] R. Guia Chávez. Desarrollo de un sistema de gestión de eventos científicos para salud pública, 2011.

[45] J. Gutiérrez. Introducción al proceso de pruebas.

[46] J.L. Moreno Alvarez. *Aplicacion de un Sistema Experto para el desarrollo de Sistema Evaluador del modelo Capability Maturity Model (CMM) niveles dos y tres*. PhD thesis, May 2004.

[47] E.J. Rojas, J. y Barrios. Investigación sobre estado del arte en diseño y aplicación de pruebas de software.

Bibliografía

[AGM] J.A. Aguilar, I. Garrigós, and J.N. Mazón. Modelos de weaving para trazabilidad de requisitos web en a-ooh. In *DSDM: Actas del VII Taller sobre Desarrollo de Software Dirigido por Modelos, JISBD, Congreso Espanol de Informatica (CEDI), Valencia, Espana, SISTEDES*, pages 146–155.

[BRJ+99] G. Booch, J. Rumbaugh, I. Jacobson, J.S. Martínez, and J.G. Molina. *El lenguaje unificado de modelado*. Addison Wesley Madrid, 1999.

[CALC07] J. Cabero Almenara and M.C. Llorente Cejudo. La interacción en el aprendizaje en red: Uso de herramientas, elementos de análisis y posibilidades educativas. *RIED: revista iberoamericana de educación a distancia*, 10(2):97–123, 2007.

[CCLM06] I. Castillo, R. Caldera, F. Losavio, and A. Matteo. Caracterización de sistemas fiables basada en un modelo estándar de calidad. In *Proceeding XXXII Conferencia Latinoamericana de Informática CLEI*, 2006.

[dlA] P. de la Asignatura. Ingeniería del software (3º itis-itig).

[DT04] P. Dodds and SE Thropp. Advanced distributed learning initiative-sharable content object reference model–2004 overview, 2004.

[EBBEMM02] F. Esteban Bara, M.R. Buxarrais Estrada, and M. Martínez Martín. La universidad como espacio de aprendizaje ético. *Revista Iberoamericana de educación*, (29):17–44, 2002.

[GEMR06] J.J. Gutiérrez, M.J. Escalona, M. Mejías, and A.M. Reina. Modelos de pruebas para pruebas del sistema. *Taller de Desarrollo de Software Dirigido por Modelos. XIII Jornadas sobre Ingeniería del Software y Bases de Datos JISBD*, 2006.

[GST06] A.M. Galvez, F.T. Serrano, and F. Tirado. *Sociabilidad en pantalla: un estudio de la interacción en los entornos virtuales*. Uoc Sl Editorial, 2006.

[Lar99b] C. Larman. *UML y patrones*. Pearson, 1999.

[LMM06] M.G. Lopez, V. Miguel, and N. Montaño. Prototipo del repositorio de objetos de aprendizaje de un sistema generador de ambientes de enseñanza-aprendizaje basados en objetos de aprendizaje (ambar). In *III Simposio Pluridisciplinar sobre Objetos de Aprendizaje y Diseños de Aprendizaje apoyados en las tecnologías (OD@ 06). Oviedo, España*, 2006.

[MLJLG05] F. Montero, V. López-Jaquero, M. Lozano, and P. González. Idealxml: un entorno para la gestión de experiencia relacionada con el desarrollo hipermedial. *ADACO: Ingeniería de la usabilidad en nuevos paradigmas aplicados a entornos web colaborativos y adaptativos, Proyecto Cicyt TEN2004-08000-C03-03, Taller celebrado en Granada*, 2005.

[MOCC05] T. Mauri, J. Onrubia, C. Coll, and R. Colomina. La calidad de los contenidos educativos reutilizables: diseño, usabilidad y prácticas de uso. *RED–Revista de educación a distancia. número monográfico II*, 2005.

[Som05] I. Sommerville. *Ingeniería del software*. Pearson Educación, 2005.

[tut] Tutorial de UML - modelo de clases. http://www.dcc.uchile.cl/˜psalinas/uml/modelo.html.

[Var02] M.L. Varas. Repositorios de objetos de aprendizaje. *URL www. alejandria. cl/recursos/documentos/documento_varas. doc. visto el*, 29, 2002.

[Zap05] M. Zapata. Secuenciación de contenidos y objetos de aprendizaje. *RED. Revista de Educación a Distancia. Número monográfico II. Disponible en: http://www. um. es/ead/red/ M*, 2, 2005.