

Universidad de las Ciencias Informáticas

**Facultad 4**



**Título:**

## **Análisis y Diseño de la versión 3.0 de RHODA**

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

**Autor:**

Leonardo Rodríguez González

**Tutores:**

Ing. Javier Soler Martín

Ing. Roxana Cañizares González

Ciudad de La Habana, X de mayo de 2011

“Año 53 de la Revolución”

## Declaración de Autoría

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de la Ciencias Informáticas a hacer uso del mismo en su beneficio.

Y para que así conste firmo la presente a los 24 días del mes de Mayo de 2011.

Autor:

---

Leonardo Rodriguez González

Tutores

---

Ing. Javier Soler Martín

---

Ing. Roxana Cañizares González



*“Y si alguna de las cosas que decimos las explota el enemigo  
y nos producen profunda vergüenza,  
¡¡bienvenida sea la vergüenza!!...  
¡¡bienvenida sea la pena!!,  
si sabemos convertir la vergüenza en fuerza,  
si sabemos convertir la vergüenza en espíritu de trabajo,  
si sabemos convertir la vergüenza en dignidad,  
si sabemos convertir la vergüenza en moral”  
Fidel Castro Ruz.*

## *Dedicatoria*

*Dedico la realización de este trabajo:*

*A todos aquellos que han hecho posible y han contribuido con su desarrollo.*

*A mis padres.*

*A mis amigos.*

*A Fidel y la Revolución cubana.*

## *Agradecimientos*

*A mis padres, que han hecho hasta lo imposible para enseñarme como hacer las cosas en la vida, las primeras personas que confiaron en mí, quienes se desvelaron por verme crecer, quienes han sufrido la ausencia de su único hijo fuera de casa, quienes constituyen mi guía, mi desvelo, los destinatarios de mi gran amor. A quienes no me alcanzará la vida para agradecerles.*

*A mis amigos, en los que siempre he encontrado ayuda, los que nunca me han dejado solo, los que han depositado su confianza en mí, aquellos que se han convertido en los hermanos que no tuve.*

*A mi familia, quienes siempre han estado ahí dispuestos a ayudar, los que siempre han confiado en mí. Quienes me han hecho llorar y reír. Los que han estado unidos a mí en todo momento.*

*A los maestros y profesores que han hecho posible mi educación, los responsables de que tantos años de esfuerzos obtengan sus frutos.*

*A la incalculable y valiosa ayuda de mis tutores Javier y Roxana, personas a las que siempre tendré que agradecer.*

*A Fidel y la Revolución cubana, la obra noble y generosa que me ha permitido convertirme en un hombre de bien.*

*A Cuba la tierra que me vio nacer la que me permitió crecer.*

*A los que durante el transcurrir de los años me han criticado.*

*A la vida, que me ha dado tanto.*



## Resumen

Los Repositorios de Objetos de Aprendizaje son la infraestructura clave para el almacenamiento, administración, localización y recuperación de Objetos de Aprendizaje. Varias iniciativas existen a nivel mundial (Agrega, Merlot, Connexions, entre otros) y algunos centros de la educación superior en Cuba también cuentan con sistemas con este objetivo.

En la Universidad de las Ciencias Informáticas desde el año 2009 se instaló la versión uno del Repositorio de Objetos de Aprendizaje RHODA, como sistema de apoyo a los procesos de enseñanza – aprendizaje existentes. Fue desarrollado por el centro Tecnologías para la Formación de la propia universidad. Durante la realización de un diagnóstico a esta versión se pudo detectar un grupo de deficiencias que constituyen limitaciones para una correcta gestión de sus procesos. Con el objetivo de perfeccionar estos procesos se realiza la presente investigación, que plantea a partir del estudio del estado del arte, el análisis y diseño de una nueva versión que incorpora nuevas funcionalidades y mejora algunas de las existentes. Para su desarrollo se utilizó la metodología RUP, que incluye a UML como lenguaje de modelado para la generación de los artefactos de las fases de inicio y elaboración.

**Palabras clave:** análisis, diseño, estándares, e-learning, metadatos, Objeto de Aprendizaje, Repositorio de Objetos de Aprendizaje.

## INDICE DE CONTENIDO

Introducción .....	1
Capítulo 1. Fundamentación Teórica .....	1
1.1 Los entornos <i>e-learning</i> .....	1
1.2 Los Repositorios de Objetos de Aprendizaje .....	3
1.3 Tipos de Repositorios de Objetos de Aprendizaje.....	5
1.4 La creación de Repositorios de Objetos de Aprendizaje .....	6
1.5 Soluciones conocidas, nacional e internacionalmente .....	6
1.6 Herramientas colaborativas .....	10
1.7 Herramientas de Control de Versiones .....	12
1.8 Estándares y especificaciones.....	13
1.9 Herramientas y tecnologías a utilizar .....	17
Capítulo 2. Análisis y Diseño .....	27
2.1 Modelo de dominio .....	27
2.1.1 Definición de conceptos fundamentales.....	27
2.2 Requerimientos.....	30
2.3 Identificación de roles, actores y casos de uso del sistema .....	36
2.4 Diagramas de casos de uso.....	41
2.5 Descripción de casos de uso .....	44
2.6 Modelo de análisis .....	52
2.6.1 Diagramas de clases del análisis .....	52
2.6.2 Diagramas de colaboración .....	54
2.7 Modelo de diseño .....	57
2.7.1 Diagramas de clases del diseño (DCD). .....	58
Conclusiones .....	60
Recomendaciones .....	61
Referencias Bibliográficas.....	62
Glosario de Términos.....	66



# INTRODUCCIÓN

## Introducción

El progresivo desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) que durante varias décadas se ha venido produciendo, ha sido la razón principal para que el hombre se provea de nuevas herramientas que ayuden y flexibilicen el trabajo. Este vertiginoso desarrollo ha conseguido que se expanda a todas las esferas la Informática como una ciencia aplicada de mucha utilidad y gran auge sobre todo en las últimas décadas del pasado siglo XX y el presente siglo XXI.

El desarrollo de la Informática, utilizado como medio de apoyo en campos particulares como la educación, ha contribuido a mejorar sus procesos, tanto en las modalidades presenciales como a distancia. Esto ha permitido que sean utilizadas las TIC como medio imprescindible para la creación de espacios virtuales, que apoyen la formación, surgiendo y desarrollándose de esta manera el e-learning.

El término e-learning describe el *“conjunto de tecnologías, aplicaciones y servicios orientados a facilitar la enseñanza y el aprendizaje a través de Internet/Intranet, que facilitan el acceso a la información y la comunicación con otros participantes”* (Red TNet España, 2006) y ha generado un acercamiento al plano de la enseñanza de los distintos individuos que tenían como limitante, estar en contacto continuo con los tradicionales procesos de formación.

Los ambientes e-learning tienden a ser complejos en cuanto a su operación, razón por la cual se han llevado a cabo y continúan realizándose notables esfuerzos para desarrollar especificaciones y estándares con la intencionalidad de crear una forma común de identificar, definir y comunicar a los elementos involucrados en estos tipos de entornos (Sistemas de Gestión del Aprendizaje, Herramientas de Autor para la Creación de Objetos de Aprendizaje, Repositorios de Objetos de Aprendizaje, entre otras) (IMS GLC, 2004).

Dentro de los esfuerzos por la estandarización, se encuentra la creación de los contenidos educativos digitales como Objetos de Aprendizaje (OA), éstos deben cumplir con características y funcionalidades que les permitan ser reutilizados y compartidos, entre personas y sistemas, así como ser transportables e intercambiables entre distintas aplicaciones.

Desde 1994 en que Wayne Hodgins define por primera vez los OA hasta la actualidad, se ha manejado un amplio número de conceptos, algunos como Wiley aseguran que son elementos que apoyan la enseñanza asistida por computadora, la IEEE asevera que son objetos digitales que pueden ser utilizados y

## INTRODUCCIÓN

referenciados durante el aprendizaje asistido con tecnologías, Chiariani afirma que son entidades digitales utilizadas por los diseñadores para la construcción de pequeñas piezas de componentes instruccionales. Debido a la gran diversidad de conceptos existentes y en coincidencia con los criterios antes expuestos el autor considera como un OA para la presente investigación: recurso digital con una granularidad apropiada y una marcada intención formativa, compuesto por uno o varios objetos de información, con un único objetivo, descrito con metadatos y con un comportamiento secuenciado que asegure el correcto enlace entre los elementos de su estructura didáctica y que puede ser reutilizado en entornos e-learning.

Una actividad natural en el campo de los OA es maximizar el número de contextos donde estos puedan ser utilizados, rigiéndose por el paradigma orientado a objetos (Leyva Leyva, et al., 2008) al igual que la tendencia a concentrarlos o recopilarlos en contenedores que los organicen y mantengan disponibles para diversos usos. Estos contenedores son conocidos en la actualidad como Repositorios de Objetos de Aprendizaje (ROA).

El acceso y disponibilidad de los OA son factores claves para el desarrollo y mejor desempeño de los entornos e-learning (Pujol, 2005). La falta de una correcta organización y de una herramienta apropiada para la gestión de contenidos, propicia que los recursos se extravíen, se duplique la información y se pierda tiempo en su localización. Los repositorios son la infraestructura clave para el desarrollo, almacenamiento, administración, localización y recuperación de todo tipo de contenido digital (ADL, 2002). Contar con herramientas como los ROA contribuye además de lo antes expuesto; al mantenimiento, acceso y distribución de los recursos educativos que se encuentran disponibles en la red.

Las características de los ROA están básicamente definidas por la de los OA, los más conocidos (Merlot, CAREO, Agrega) operan de forma independiente, son aplicaciones con una interfaz web, un mecanismo de búsqueda y listado con algún tipo de clasificación.

Para la construcción de los ROA varias organizaciones y consorcios se han dado la tarea de crear especificaciones y estándares para el *e-learning*, dentro de ellos se puede citar los esquemas de metadatos especializados (Dublin Core, Learning Object Metadata (LOM), IMS-MD), siendo una tendencia la utilización del LOM o algún esquema derivado de este. ADL (2002) propone un conjunto de funcionalidades a cumplir por los ROA con el objetivo de dar acceso a los OA en un ambiente seguro.

Respecto a este campo Cuba no se ha quedado rezagada, ha incorporado a los métodos tradicionales educativos, principalmente en la educación superior, herramientas educativas, de apoyo a las

## INTRODUCCIÓN

modalidades presenciales, semipresenciales o las totalmente a distancia. En este ámbito varias instituciones han comenzado a crear sus propios repositorios, tal es el caso del Repositorio de Recursos Educativos de la Universidad Virtual de la Salud (RREUVS), de la red telemática Infomed; el repositorio de la Universidad Agraria de la Habana (RUNAH), el repositorio de recursos digitales de la Universidad de la Habana, el creado en el Centro Universitario Jose Antonio Echeverría entre otros. Por lo general estos repositorios no tienen comunicación con otros elementos del entorno donde han sido instalados, no soportan un sistema de revisiones, igualmente carecen de un sistema de empaquetado que permita sus recursos puedan ser utilizados en otros repositorios

La Universidad de las Ciencias Informáticas<sup>1</sup> (UCI); donde se ha hecho imprescindible el uso de entornos e-learning debido al modelo de formación centrado en el aprendizaje, con una incorporación progresiva a la enseñanza semipresencial en entornos virtuales de aprendizaje y una intensa vinculación entre la investigación y la producción; ha obtenido un notable desarrollo en este campo. Se cuenta con la utilización de un Entorno Virtual de Aprendizaje (EVA), creado a partir de una personalización del LMS Moodle<sup>2</sup>, creación de la Herramienta de Autor CRODA (Creando Objetos de Aprendizaje), el Entorno Virtual de Postgrado y la creación de un Repositorio de Objetos de Aprendizaje, a cargo de un equipo de desarrollo conformado por estudiantes y profesores de la propia universidad. Encontrándose operable y en pleno uso por los docentes de la universidad la versión 1.0.

Para el desarrollo de RHODA se tuvo en cuenta el uso de estándares: Learning Object Metadata (LOM) como esquema de metadatos, Simple Query Interface (SQI) para la interoperabilidad y las especificaciones básicas dictadas por la guía Sharable Content Object Reference Model (SCORM) 1.2 y 2004 para el empaquetamiento de los OA. Al igual que el resto de los repositorios antes mencionados, (RREUVS, RUNAH, UH, CUJAE, CAREO, MERLOT, Agrega) RHODA posee una interfaz web amigable y permite realizar un conjunto de funcionalidades distribuidas en varios módulos: Gestión de Usuarios, Gestión de OA, Foro, Estadísticas y Administración.

Este repositorio posee un panel de administración con funciones básicas implementadas, que necesitan ser perfeccionadas y de la inclusión de algunas otras funciones que disminuyan la complejidad de

---

<sup>1</sup> Universidad creada en año 2003 donde se encontraba localizada la base de radio-escucha: LOURDES. Adjunta al Ministerio de Informática y las Comunicaciones de Cuba. Más información en el sitio: [www.uci.cu](http://www.uci.cu)

<sup>2</sup> CMS dedicado a la creación de entornos virtuales para el aprendizaje, para más información ver: <http://moodle.org/>

## INTRODUCCIÓN

determinadas acciones; como el control de Logs; en la plataforma que permiten una correcta administración y control de la de seguridad y configuración.

Aunque posee un sistema de mensajería interna, este no mantiene informados oportunamente a los usuarios de los eventos que en él se producen, pasando inadvertidos y privando a la comunidad de un gran cúmulo de información que puede ser de gran ayuda en la gestión de sus OA.

Aunque RHODA presenta un entorno de colaboración en el proceso de creación de un OA, no cuenta con el uso de herramientas que propicien e incentiven la formación de grupos de usuarios para desarrollar determinadas acciones y procesos, estos aspectos son necesarios con el fin de proveer un ambiente de socialización del conocimiento e intercambio de ideas en la confección y publicación de los OA.

RHODA permite la creación de OA tanto de forma individual como colaborativa, y posibilita la generación de OA con un bajo nivel de complejidad, permitiendo que estos puedan ser editados, durante su creación o una vez almacenados. Al realizar la edición de un OA en el sistema actual, se pierde la información correspondiente a la versión anterior, sustituyéndola por la editada.

La recuperación de información es una de las razones fundamentales de la existencia de cualquier repositorio. RHODA actualmente permite realizar 2 tipos de búsquedas, simple y avanzada, la búsqueda avanzada se puede realizar atendiendo a los criterios de “categoría”, “subcategoría”, “autor” o a través de la búsqueda por metadatos, estableciendo valores como criterios de búsqueda, para un subconjunto especificado de éstos. El sistema actual no cuenta con otro tipo de recuperación de información que permita la localización eficiente y efectiva de los OA.

RHODA aunque tiene un gran volumen de funcionalidades, su comunicación con otros sistemas es un tanto limitada, característica de mucha importancia en este tipo de plataformas, ya que potencia la reutilización de OA y permite que ambos sistemas puedan intercambiar información y trabajar conjuntamente.

Por todo lo antes expuesto se plantea como **problema a resolver** las deficiencias presentes en la versión 2.0 de RHODA constituyen limitaciones en la gestión de sus procesos.

Se define como **objeto de estudio** los procesos de gestión en los Repositorios de Objetos de Aprendizaje.

## INTRODUCCIÓN

**Objetivo general** realizar el análisis y diseño de una nueva versión de RHODA para perfeccionar la gestión de sus procesos.

Teniendo como **campo de acción** la concepción, análisis y diseño de repositorios de objetos de aprendizaje.

### Idea a defender

Con la realización del análisis y diseño de una nueva versión de RHODA es posible obtener una adecuada entrada a la implementación, que garantice un perfeccionamiento en la gestión de los procesos del repositorio.

### Objetivos específicos

- Elaborar el marco teórico de la investigación.
- Desarrollar los artefactos correspondientes al análisis y diseño de la nueva versión de RHODA.

Para dar cumplimiento a los objetivos propuestos se recurrió a la utilización de **métodos de investigación** con el objetivo de proveer a la investigación las bases necesarias, que tributen a su éxito.

Entre los métodos de investigación teóricos utilizados se encuentran: **Analítico-Sintético**, que permitió identificar los conceptos y definiciones más importantes relacionados con el desarrollo de ROA, y permitió generar luego, una propuesta de análisis y diseño adecuada a la situación planteada y las tecnologías estudiadas; el **Histórico-Lógico** que permitió realizar un estudio de los diferentes ROA existentes, sus características, ventajas así como su evolución. **Modelación**, el cual posibilita el esbozo de los diferentes diagramas y modelos del proceso de análisis y diseño de la aplicación. También se utiliza de los **métodos empíricos** la **Observación**, para detallar el comportamiento de los sistemas existentes y permite extraer información del modo en que estos gestionan los OA.

### Estructura capitular

Para el desarrollo del presente trabajo se compone de 2 capítulos. Cada uno de ellos aborda diferentes temáticas y logran una mejor distribución del contenido. Los capítulos están estructurados de la siguiente manera.

## INTRODUCCIÓN

**Capítulo I.** Describe el marco teórico de la investigación. Se realiza el análisis del estado del arte y evolución de los ROA. Se enuncian, los principales conceptos relacionados a ellos y la utilización de los estándares establecidos por la comunidad e-learning para la construcción de sistemas informáticos como los ROA. Se expone la tecnología utilizada para el modelado de la aplicación. Además se justifican las herramientas, lenguaje de modelado y metodología utilizada.

**Capítulo II.** Se realiza el análisis y diseño de la aplicación. Se exponen además, los artefactos generados durante la realización de este flujo de trabajo.

Este documento contiene, Conclusiones, Recomendaciones, Referencias Bibliográficas, Anexos y Glosario de Términos.

# CAPÍTULO 1. Fundamentación teórica

## Capítulo 1. Fundamentación Teórica

### Introducción

Se exponen un conjunto de fundamentos con el objetivo de crear el marco teórico relacionado con los aspectos tratados en el objeto de estudio y campo de acción, a fin de crear las bases teóricas y metodológicas que propician generar la solución propuesta.

Se abordan las definiciones que se dan a los entornos *e-learning*, y se profundiza conceptualmente respecto los ROA. De igual manera se tratan las características más importantes relacionadas a los OA, así como las definiciones respecto a ellos. Se realiza el análisis de varios ROA para conocer el estado del arte en este campo. Se fundamenta la selección de las herramientas, metodología y tecnologías a utilizar para realizar el modelado de la versión 3.0 de RHODA

### 1.1 Los entornos *e-learning*

El avance y el desarrollo de las TIC, la llegada de la era de la información, el surgimiento de Internet y con él, el nacimiento y desarrollo de la web constituyen indudablemente factores que han revolucionado la manera en que llega el conocimiento a los distintos individuos. Todos estos cambios han permitido una cuantiosa mejora en los sistemas de educación a distancia, en este sentido se afirma que *“en realidad la educación a distancia ha tenido siempre una íntima relación con la tecnología, sobre todo con la que la sociedad encontraba hegemónica en cada momento”* (Mena, 2004).

La aplicación de estas transformaciones tecnológicas a la educación, es la causa del surgimiento de complejos entornos denominados *e-learning*, los que se han convertido en una estructura necesaria en el campo de la educación. Tanto los esquemas tradicionales de enseñanza presencial como los de aprendizaje a distancia, han obtenido con el desarrollo de entornos *e-learning* un valioso recurso de apoyo. Convirtiéndose en la modalidad más innovadora dentro del campo educacional (Mena, 2004).

Los conceptos e ideas que ayudan a definir y giran en torno el término *e-learning* son muchos, si se busca la raíz de la palabra se puede traducir como *“aprendizaje electrónico”*, Rosenberg (2001) lo define como: *“el uso de tecnologías Internet para la entrega de un amplio rango de soluciones que mejoran el conocimiento y el rendimiento. Está basado en tres criterios fundamentales:*

# CAPÍTULO 1. Fundamentación teórica

- *El e-Learning trabaja en red, lo que lo hace capaz de ser instantáneamente actualizado, almacenado, recuperado, distribuido y permite compartir instrucción o información.*
- *Es entregado al usuario final a través del uso de ordenadores utilizando tecnología estándar de Internet.*
- *Se enfoca en la visión más amplia del aprendizaje que van más allá de los paradigmas tradicionales de capacitación.”*

Otros autores afirman que es el “conjunto de tecnologías, aplicaciones y servicios orientados a facilitar la enseñanza y el aprendizaje a través de medios electrónicos” (Fernandez-Majón, 2006). Finalmente en la presente investigación se tomará como definición de *e-learning* “el conjunto de tecnologías, aplicaciones y servicios orientados a facilitar la enseñanza y el aprendizaje a través de Internet/Intranet, que facilitan el acceso a la información y la comunicación con otros participantes” (Red TTnet España, 2006), debido a que es, de todos los conceptos analizados el que más se ajusta a la línea de la presente investigación.

En consecuencia con la definición antes dada y según la definición dada por López (2005), de entorno e-learning integral, la tendencia es a crear grandes repositorios de OA tal como se muestra en la Figura 1, los cuales forman parte del propio entorno. Esto ha traído como efecto la creación de complejos ROA dedicados al almacenamiento de recursos digitales, que dan soporte a los procesos de enseñanza y aprendizaje, evidenciado a través de las experiencias a nivel internacional como nacional entre las que se puede citar: MERLOT<sup>3</sup> (Multimedia Educational Resource for Learning Online Teaching), CAREO<sup>4</sup> (Campus Alberta Repository of Educational Objects), Agrega<sup>5</sup>, RREUVS<sup>6</sup> RUNAH, UH, CUJAE.

Los ROA se han convertido en los sistemas por excelencia dentro de los entornos e-learning para la gestión de los OA, “es por ello que la presencia de un repositorio de objetos de aprendizaje en un entorno educativo tiene gran importancia si se quiere lograr una buena gestión del OA, de lo contrario todas estas tareas costarían un alto costo y esfuerzo” (Salvador, et al., 2010).

Los argumentos expuestos evidencian como los ROA se han unido indisolublemente a los entornos e-learning por lo que se hace necesario profundizar y delimitar una definición para la presente investigación de ROA.

---

<sup>3</sup> Disponible en <http://www.merlot.org>

<sup>4</sup> Disponible en <http://www.careo.org>

<sup>5</sup> Disponible en <http://redes.agrega.indra.es>

<sup>6</sup> Disponible en <http://www.uvs.sld.cu/repositorio/recursos-uvs/>



# CAPÍTULO 1. Fundamentación teórica

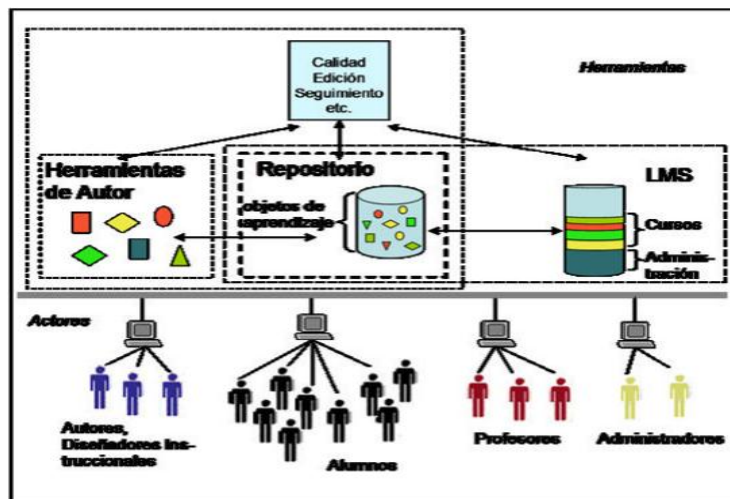


Figura 1. Entorno e-learning integral. (López, 2005)

## 1.2 Los Repositorios de Objetos de Aprendizaje

El origen de la palabra repositorio proviene del latín *repositorium* que significa armario, alacena en castellano. Según el diccionario de la Real Academia Española (RAE) el término repositorio significa: “*lugar donde se almacena algo*” (RAE, 2010). Los conceptos relacionados al término se vieron revolucionados con el advenimiento de la era de la información, lo que permitió aplicar el término repositorio al léxico específico de la informática para designar los “*depósitos de información digital*”. (Huaroto, 2010)

Los ROA son una distinción especial de los repositorios digitales quienes regularmente “*constituyen sitios centralizados donde se almacena recursos digitales de todo tipo*” (López, 2005). Enfocado a ellos existen una amplia gama de conceptos, algunos autores se refieren a estos como “*un catálogo electrónico/digital que facilita las búsquedas en internet de objetos digitales para el aprendizaje*” (Griff, et al., 2002), otros como Daniel (2004), aseguran que “*son bases de datos con búsquedas, que alojan recursos digitales y/o metadatos que pueden ser utilizados para el aprendizaje mediado*” (Graeme, 2004), el JORUM+ project<sup>7</sup> (2005), 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*

<sup>7</sup> JORUM+ project es un proyecto financiado y dirigido por Joint Information Systems Committee del Reino Unido. El proyecto se inició en el año 2002 y terminó en el 2005. Con dos líneas fundamentales: Investigar la necesidad de la utilización de repositorios y la creación de repositorios dirigidos facilitar el trabajo en la educación universitaria.

## CAPÍTULO 1. Fundamentación teórica

*las ubicaciones de aquellos objetos almacenados en otros sitios, tanto en línea como en ubicaciones locales” (JORUM TEAM PROJECT , 2005).*

Las anteriores definiciones, en sentido general no difieren mucho y llegan a un punto común en que los ROA son bases de datos o catálogos creados para ser utilizados en el proceso de enseñanza, estas definiciones aunque están bastante cerca de la realidad, pueden conllevar a cometer errores y generar confusiones, de esta forma el autor concluye que: los ROA son sistemas complejos, especializados en el almacenamiento de OA. Accesibles y operables por usuarios, con funciones encaminadas a la clasificación, localización, reutilización, recuperación y mantenimiento de los OA, que permiten compartir estos con otros sistemas de los entornos e-learning.

La anterior definición de ROA carece de sentido, si antes no se define correctamente qué es un OA, condición que permite realizar una completa diferenciación entre las bibliotecas digitales (Borgman, 1999) y repositorios de objetos de aprendizaje (Chowdhury, et al., 2002). Los criterios relacionados con los OA son muy variados, algunos autores los definen como *“elementos para la instrucción, aprendizaje o enseñanza basada en computadora. No son realmente una tecnología, más propiamente dicho son una filosofía”* (Wiley, 2000) que se fundamenta en la corriente de la ciencia de la computación conocida como orientación a objetos (Coad, et al., 1993). El Comité de Estandarización de Tecnología Educativa IEEE-LTSC (2001), dice que los OA son *“una entidad digital o no digital que puede ser utilizada, reutilizada y referenciada durante el aprendizaje apoyado con tecnología”* (IEEE LTSC, 2001). Otros se refieren a los OA como *“entidades digitales distribuibles a través de Internet, con posibilidades de acceso simultáneo, utilizables por los diseñadores para construir pequeñas piezas de componentes instruccionales, reutilizables en diferentes contextos. Estas piezas pueden ser autocontenidas e incluir en su estructura otros objetos o soportar objetivos instruccionales individuales”*. (Chiarani, y otros, 2006)

Zapata (2009), coincide en muchos puntos con los anteriores autores y además hace especial énfasis en la *“ordenación o secuenciación de los contenidos de enseñanza que asegure el enlace entre los objetivos educativos y las actividades de aprendizaje de los alumnos”* que debe poseer todo OA. (Ros, 2009)

Formalmente no existe una única definición para OA, en coincidencia con los criterios aportados por los anteriores autores, el autor de la presente investigación, utilizará como definición de OA: 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

# CAPÍTULO 1. Fundamentación teórica

asegure el correcto enlace entre los elementos de su estructura didáctica y que puede ser reutilizado en entornos *e-learning*.

## 1.3 Tipos de Repositorios de Objetos de Aprendizaje

Los ROA pueden clasificarse utilizando distintos criterios de unificación, de ellos los más usados hoy día tienden a utilizar como característica fundamental: ¿Qué es lo que almacenan?, ¿Cómo se organizan los catálogos de metadatos? Según Downes (2004) utilizando el primer criterio se pueden diferenciar 2 tipos principales de repositorios: los que están orientados a almacenar los objetos de aprendizaje y sus metadatos (Figura 2a) y los que solo almacenan los metadatos de los OA (Figura 2b) (Downes, 2004). En el primero los OA y sus descriptores se encuentran dentro de un mismo sistema, regularmente concentrados en un mismo servidor; en el segundo caso el ROA solo almacena los datos que describen el OA y una referencia que apunta a la ubicación física del mismo y a través de la cual se puede acceder a él, generalmente ubicado en otros repositorios o sistemas (Mason, et al., 2004).

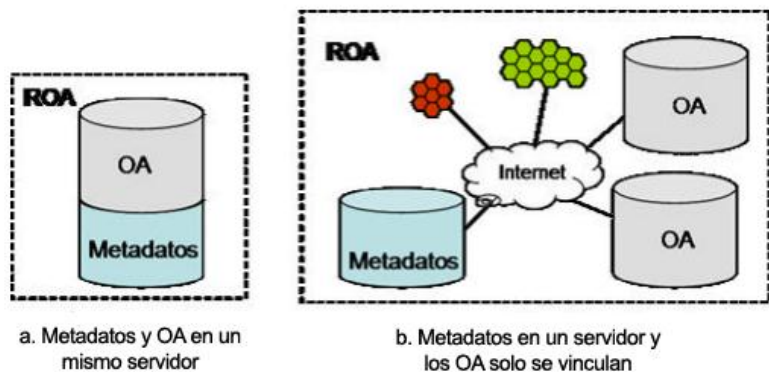


Figura 2. Tipos de ROA según almacenamiento. (Mason, et al., 2004)

Como otro tipo se puede encontrar una variante híbrida de los dos antes mencionados y que es conocido como mixto.

Utilizando el segundo criterio, se pueden definir dos modelos principales de ROA según Downes (2004): centralizados y distribuidos. Los primeros poseen un sistema de catalogación donde los metadatos de los OA están alojados en un mismo servidor aunque el objeto esté almacenado en otro. Los segundos aunque menos conocidos, implementan eficientes sistemas de localización y búsqueda, operando a través de

# CAPÍTULO 1. Fundamentación teórica

varios servidores donde cada uno contiene diferentes grupos de metadatos, los que mantienen un alto nivel de comunicación para intercambiarlos (Downes, 2004).

Independientemente de las características que permiten clasificar a los ROA, lo más importante es que contribuyan a mejorar los procesos de gestión de OA dentro de los entornos educativos y que tengan *“capacidades, tanto para ofrecer una interfaz web, para que los usuarios humanos puedan acceder a la colección, así como la capacidad de comunicarse directamente con las plataformas de aprendizaje y hacer posible la interoperabilidad<sup>8</sup> entre sistemas de diferente naturaleza”* (López, 2005).

## 1.4 La creación de Repositorios de Objetos de Aprendizaje

La práctica de crear ROA, aunque tiene algunos años de experiencia, sigue siendo algo novedosa, en este aspecto ha sido de vital importancia incorporar cierto valor al desarrollo de estos complejos sistemas. Teniendo como prioridad, la facilidad con que los usuarios interactúan con estos, las facilidades que brindan para el mantenimiento, producción, almacenamiento y localización de OA, incluyendo las posibilidades de comunicación con otras plataformas de igual corte aplicativo, de forma tal que este intercambio provea una colaboración entre ambos sistemas, permitiendo así que una reutilice los recursos y servicios publicados en la otra.

La creación de aplicaciones web como los ROA ha tenido seguimiento por parte de muchas organizaciones a nivel internacional (López, et al., 2006) las que han colaborado en creación de estándares convertidos hoy en guías para el desarrollo. Esto ha permitido el comienzo de importantes proyectos que hoy marcan y constituyen referencia obligatoria para los diseñadores de ROA. A continuación se mencionan varias de las características de algunas de las iniciativas que constituyen fuentes de referencia a nivel mundial en este campo.

## 1.5 Soluciones conocidas, nacional e internacionalmente

### MERLOT

Es un recurso libre y abierto, diseñado principalmente para profesionales y estudiantes enmarcados en la enseñanza universitaria, creado en 1997 por California State University-Center for Distributed Learning.

---

<sup>8</sup> Capacidad de comunicar, ejecutar programas, o transferir datos entre varias unidades funcionales de forma que el usuario no tenga la necesidad de conocer las características únicas de estas unidades. (ISO/IEC 2382) disponible en: [http://www.digitalpreservationeurope.eu/publications/briefs/es\\_interoperabilidad.pdf](http://www.digitalpreservationeurope.eu/publications/briefs/es_interoperabilidad.pdf) página 1.

# CAPÍTULO 1. Fundamentación teórica

Entre sus principales características se encuentran: que es un repositorio centralizado que contiene solo los metadatos de los objetos y referencia la ubicación de estos en sitios remotos. Su funcionamiento es del tipo *standalone*<sup>9</sup>, y actúa como un portal de OA. Brinda servicios de búsquedas, las que de forma interna se ejecutan a través de la coincidencia con palabras clave, las que se pueden combinar con criterios como las categorías, posee además un formulario para la ejecución de búsquedas avanzadas que permiten diseñar complejos criterios de búsquedas y así satisfacer las necesidades de los usuarios sobre un contenido que se necesite y esté creado. Permite que se ejecuten funciones de importar y exportar recursos digitales, en el caso particular de los OA, los mismos son exportados según el formato técnico con el que se han creado generalmente SCORM 1.2. Además permite la creación de un sistema de metadatos para los recursos, los que sirven para su posterior localización y catalogación.

La funcionalidad de importar solo la pueden ejecutar los usuarios registrados, tarea que se realiza de forma gratuita y sin adquirir responsabilidades directas. Automáticamente se adquiere el rol de autor, durante el proceso de registro puede seleccionar el tipo de miembro que desea ser. Los OA y recursos digitales importados y/o creados se revisan en pares y como métricas de calidad se utilizan:

- Calidad del contenido.
- Potencial de efectividad como herramienta para la enseñanza.
- Facilidad de Uso.

Estos tres indicadores permiten dar una validación para la publicación a los autores, los que son retroalimentados de forma inmediata y automática si su objeto o recurso fue aceptado o rechazado, en este último caso el autor puede realizar comentarios de los señalamientos realizados por los revisores, los cuales son publicados en la sesión del autor y enviados a través de un resumen al correo del mismo. Las evaluaciones de los revisores son otorgadas en una escala del 1 al 5, y el sistema tiene como política que los objetos con evaluaciones inferiores a 3 no puedan publicarse.

## CAREO

Proyecto apoyado por Alberta Learning Campus y CANARIE (Canadian Network for the Advancement of Research in Industry and Education). El objetivo principal de este repositorio es la creación de una

---

<sup>9</sup> Palabra correspondiente al léxico informático para denominar a sitios con funcionamiento independiente.

## CAPÍTULO 1. Fundamentación teórica

colección de materiales multidisciplinares que sean investigables enfocados a una amplia red de educadores.

Es un repositorio centralizado e independiente que da acceso a los OA remotos y locales a través de los metadatos de su colección. El programa está integrado en dos componentes la aplicación web Careo y el servidor de metadatos Aloha, que contempla las especificaciones IMS LOM como base para su esquema de metadatos CanCore<sup>10</sup>.

La aplicación permite realizar búsquedas, utilizando los criterios introducidos por los usuarios y comparándolos con la información que aparece en los metadatos, este sistema de búsqueda permite ofrecer a los usuarios respuestas rápidas. Cuando se crea un recurso en un servidor web los metadatos que lo describen son enviados directamente al servidor de metadatos a través de un componente creado para el intercambio de estos, lo que permite recopilar la colección de registros almacenados en otros repositorios o almacenes de metadatos.

A diferencia de MERLOT los metadatos de los OA están más completos y detallados, incluyendo que los OA se pueden descargar en forma paquete comprimido, aunque el paquete no es estándar ya que en el no vienen incluidos los metadatos. Es un repositorio que puede establecer comunicación con otras aplicaciones similares utilizando las especificaciones IMS DRI<sup>11</sup> como base para la interoperabilidad.

Posee un sistema de mensajería que le permite mantener informados a los usuarios de repositorio, incluye un sistema de sindicación RSS, permitiendo obtener un grupo importante de informaciones a los usuarios sin necesidad de que estos accedan al repositorio.

### AGREGA

Es un proyecto de software que corre a cargo de Red.es<sup>12</sup>, el Ministerio de Educación y Ciencia, el Ministerio de Industria, Turismo y Comercio, y los Consejos de las Comunidades Autónomas en España,

---

<sup>10</sup> **CanCore**, especificación de metadatos propuesta por la iniciativa CanCore, una organización creada por el Multimedia Learning Group of Industry Canada y soportada por la universidad de Athabasca. El conjunto de metadatos propuesto por la norma, al igual que otros estándares de metadatos, está compuesto a partir un subconjunto de elementos que aparecen en el estándar LOM de IMS.

<sup>11</sup> **IMS DRI**, trata de abordar la interoperabilidad de los repositorios abarcando cada una de las funciones que estos deben cumplir un durante su uso: búsqueda, representación, recolección, envío, almacenaje, solicitud, entrega y aviso. El propósito es poder acceder a cualquier almacén de recursos educativos para obtener dichos recursos sin necesidad de conocer cuál es la organización desarrolladora o la estructura de dicho almacén.

<sup>12</sup> **Red.es** es una Entidad Pública Empresarial adscrita al Ministerio de Industria, Turismo y Comercio a través de la Secretaría de Estado de Telecomunicaciones y para la Sociedad de la Información que tiene, con carácter general, legalmente encomendadas una serie de funciones con el objeto de contribuir al fomento y desarrollo de la sociedad de la información en España.

## **CAPÍTULO 1. Fundamentación teórica**

es un repositorio federado conformado por uno central y otros con carácter autonómico. El repositorio almacena tanto los OA como los metadatos de este, utilizando como esquema para la catalogación LOM-ES y SCORM 2004 para el empaquetado.

El repositorio implementa la búsqueda de contenidos entre repositorios usando el protocolo SQLI (Simple Query Interface). Mientras que los mecanismos de búsqueda de los nodos se basa en la indexación de un subconjunto de los metadatos que vienen descritos en el manifiesto de un OA, una vez indexados son procesados con una herramienta Lucene, que es la que se usa posteriormente para llevar a cabo las búsquedas.

La creación de los OA está limitada respecto a las capacidades máximas que permite el estándar SCORM 2004. En este sentido solo se pueden crear OA que contengan como máximo una organización y con secuenciación guiada. Las función de importar se ve limitada por la misma situación, si contiene más de una organización solo se tiene en cuenta la principal. Agrega incluye también un sistema de notificaciones sobre todo integrado por un sistema de noticias y Faqs (Frequently Asked Questions). La administración de este sistema de notificaciones a los usuarios corre a cargo del gestor de cada nodo, el cual da de alta o de baja a las noticias que le corresponden, del mismo modo son administradas las Faqs.

### **Repositorio de Recursos Educativos de la Universidad Virtual de la Salud**

EL Repositorio de Recursos Educativos de la Universidad Virtual de la Salud (UVS) es un proyecto emprendido por la red telemática de la salud en Cuba: Infomed, desarrollado con el fin de crear, almacenar, catalogar y consultar recursos educativos publicados en la UVS. El repositorio cuenta además con un grupo importante de funcionalidades que se traducen en su correcto y eficiente funcionamiento.

Con soporte en Scout Portal Toolkit<sup>13</sup>(SPT) este repositorio permite a sus usuarios realizar un grupo importante de acciones, al igual que los repositorios antes mencionados registrarse es gratis y no se adquieren obligaciones por esto. El repositorio es del tipo centralizado, su funcionamiento de forma independiente y no almacena los recursos educativos, solo los metadatos y una referencia a la web del

---

<sup>13</sup> **SPC** es una herramienta libre que permite compartir colecciones de recursos usando la World Wide Web. Creada por Internet Scout Project un centro de investigaciones dedicado al desarrollo de herramientas que mejoren los servicios de búsqueda, filtrado y presentación de información y metadato. La herramienta SPC ofrece una colección de campos de metadato con base en el estándar Dublin Core Metadata Element Set (ANSI/NISO Z39.85-2001) Disponible en: <http://scout.wisc.edu/Projects/SPT>

# **CAPÍTULO 1. Fundamentación teórica**

autor donde se encuentra físicamente. El Dublin Core Metadata Elements Set es el modelo de metadatos usado, el cual provee una serie de elementos que permiten la descripción de los recursos, lo que permite una rápida localización de los mismos.

La ejecución de la búsqueda de los recursos se puede realizar del tipo simple o avanzada. El tipo avanzado de búsqueda permite construir complejos criterios de búsqueda, incluyendo que una vez definido el criterio permite guardarlo, de manera que si se necesita realizar una búsqueda semejante, no es necesario volver a construir el criterio, solo lo seleccionamos de nuestra lista de búsquedas guardadas.

## **1.6 Herramientas colaborativas**

El desarrollo y auge de la web 2.0 o web social como comúnmente se le denomina, ha incluido importantes desafíos en la construcción de los ROA. La web 2.0 representa un gran salto en cuanto a las tecnologías web, ya que han hecho que internet sea más creativo y participativo socialmente.

Las nuevas aplicaciones web interactivas como foros, wikis y blogs junto a los sistemas de notificaciones (RSS) y a los sistemas de control de versiones, forman parte de la amplia gama de herramientas colaborativas presentes en la web. En este caso las aplicaciones web interactivas y los sistemas de notificaciones guiados por las listas de correo, se han convertido en herramientas muy populares dentro de la llamada web social.

Las herramientas que brinda la web 2.0 son particularmente aptas para preservar y organizar el conocimiento, con una gestión de aprendizaje cercano a través del uso compartido. Permiten que los usuarios dejen de comportarse como constructores pasivos, para convertirse en colaboradores en la gestión del conocimiento (Milligan, 2007).

Varios autores han recomendado integrar herramientas de la web 2.0 en los repositorios. Las estrategias de participación basadas en la web 2.0 pueden favorecer la implicación de los usuarios con estos. (Geser, 2007)

El repositorio multimedial Merlot ha sido uno de los primeros en poner en práctica el uso de herramientas de la web 2.0, la creación de comunidades ha sido una fortaleza que se ha explotado muy bien, esto ha convertido este repositorio en referencia obligatoria a nivel internacional, debido a la motivación lograda en sus usuarios y en el rápido crecimiento de su población de OA. A continuación se expondrán algunas de



# **CAPÍTULO 1. Fundamentación teórica**

las características, por las que se ha decidido integrar algunas de estas herramientas a la versión 3.0 de RHODA.

## **Weblogs**

Un weblog es una página web con apuntes fechados en orden cronológico inverso, de tal forma que la anotación más reciente es la que primero aparece, y es, ante todo, una forma libre de expresión, de creación y compartir conocimiento. Los blogs son un medio de comunicación colectivo que promueve la creación y consumo de información original y veraz, y que provoca, con mucha eficiencia, la reflexión personal y social sobre los temas de los individuos (Ovalles, 2007).

El fin de estos blogs es potenciar el análisis, la reflexión, y a la vez poder compartir experiencias, conocimientos, contenidos, va creando en el usuario una costumbre de aprender colectivamente, y mientras desarrolla sus entradas (post), se hace dueño de su propio proceso de aprendizaje.

## **Wikis**

Una wiki es un sitio web colaborativo llevado adelante por el perpetuo trabajo colectivo de muchos autores. En estructura y lógica es similar a un blog, pero en este caso cualquier persona puede editar sus contenidos, aunque hayan sido creados por otra. Permite ver todos los borradores o modificaciones del texto hasta tener la versión definitiva. El término wiki se refiere tanto al sitio web como al software usado para crear y mantener el sitio (Bermon, 2010). El primer wiki lo creó Ward Cunningham en 1995. El mejor ejemplo de wiki es la famosa enciclopedia GNU Wikipedia.

La integración a los repositorios, de herramientas como esta permite que se cree un ambiente de colaboración y de aprendizaje colectivo, donde los usuarios comparten conocimiento, lo que ayuda sin dudas a la mejor comprensión de aspectos respecto a los cuales se tiene poco conocimiento.

## **RSS**

Really Simple Syndication formato XML que se utiliza para syndicar o compartir contenidos en la web. Es un servicio donde los usuarios de internet pueden suscribirse a muchos contenidos de páginas web sin tener que recibir la sarta de información adicional y sin la necesidad de visitar cada uno de los lugares que está interesado y chequear por actualizaciones. De esta forma, los usuarios pueden crear un servicio personal de alertas de distintos canales de información y distinto tipo de fuentes. (Wordpress, 2010)

# CAPÍTULO 1. Fundamentación teórica

## Foro electrónico

Un foro electrónico es de manera general una herramienta, integrada normalmente a una página web, donde los usuarios crean conversaciones sobre diversos temas. Usualmente cada tema está relacionado a una pregunta, una noticia, anécdota u otro aspecto, generalmente iniciado por algún usuario o grupo de usuarios, el resto de los usuarios que interactúan en la web generalmente emiten criterios acerca del aspecto en cuestión, lo que significa que el foro es un lugar para emitir opiniones que dan respuesta a la pregunta o tema origen (Ovalles, 2007).

Todo foro debe poseer un moderador que es quien se encarga de coordinar los temas o hilos de discusión que se abren en él. Los foros constituyen herramientas comunicacionales, que ofrece un aporte educativo en dependencia de cómo se utilice. La inclusión de herramientas como los foros en plataformas como los ROA, permite indudablemente que los usuarios que sobre ella se desempeñan, puedan preguntar y satisfacer ciertas de sus necesidades sociales y de conocimiento. Esta herramienta es una forma más de almacenamiento del conocimiento colectivo.

## 1.7 Herramientas de Control de Versiones

Las herramientas de control de versiones son software basados en la combinación de procedimientos que permiten la gestión de versiones por la que transita un software o recurso digital. Este permite la administración de este recurso y poder determinar en qué momento y fase de desarrollo se encuentra. Las herramientas de control de versiones permiten a su vez poder facilitar el trabajo paralelo de un grupo o comunidad de usuarios.

Los controladores de versiones en su mayoría proporcionan un mecanismo de almacenamiento de los elementos que deba almacenar, la posibilidad de realizar cambios tanto parciales como totales de dichos elementos, brindan un registro o historial de las acciones que se realizaron sobre un recurso digital determinado y permiten volver a un estado anterior del mismo.

El uso de herramientas de control de versiones permite (Palomar, 2008):

- **Control exacto:** Saber cuál es la última versión del código, quién y cuándo la ha cargado.
- **Comparación de versiones:** Ver cuáles han sido los cambios realizados.

# CAPÍTULO 1. Fundamentación teórica

- **Regresar atrás:** Cuando el trabajo desarrollado no nos ha dado los resultados esperados, se puede retroceder a una versión anterior.
- **Crear distintas ramas del proyecto:** Si llegado a un punto se hace necesario hacer dos aplicaciones con distintas funcionalidades, pero con elementos en común, se pueden separar en dos ramas.
- **Concurrencia del trabajo:** Donde más de una persona puede trabajar con el mismo archivo.
- **Etiquetar distintas revisiones:** Para indicar que la revisión coincide con la versión del proyecto.

Una vez que el repositorio cuente con un sistema de control de versiones sería posible, obtener una versión anterior de un OA, llevar un correcto control de los cambios, acciones que se requieren para una completa gestión de los OA.

## 1.8 Estándares y especificaciones.

En la actualidad las personas usan los términos “estándares” y “especificaciones” indistintamente, cabe destacar que aunque no son términos equivalentes poseen aplicaciones semejantes (Fernandez-Majón, 2006). Se puede decir que un estándar es lo “que sirve como tipo, norma, modelo, patrón o referencia” (RAE, 2010). De forma general los estándares se pueden clasificar de dos formas, los de *jure* u oficiales y los de *facto* (Salvador, et al., 2010).

Los estándares de *jure* son aquellos que han sido aprobados por alguna organización acreditada que certifica su validez (IEEE/LTSC, ISO/IEC), estos estándares en algunos casos son de cumplimiento obligatorio. Los de *facto* son aquellos que se usan por voluntad propia y poseen un elevado nivel de aceptación, aunque este no haya sido previamente certificado por alguna organización, este tipo de estándar primero fue una especificación.

“Cuando un formato, tecnología o método propuesto no ha sido aprobado por alguna entidad certificadora, entonces este es una especificación” (Iglesias, 2003). Cuando el uso de una especificación es extendido entonces se puede considerar como un estándar de *facto*.

La estandarización es una acción necesaria, el uso de estándares permite habilidades como (Hodgins, 2001):

# CAPÍTULO 1. Fundamentación teórica

- Reutilización: flexibilidad de poder utilizar componentes en otras aplicaciones y contextos.
- Interoperabilidad: capacidad para poder integrarse en estructuras (plataformas) y sistemas diferentes.
- Accesibilidad: para localizar y acceder materiales instruccionales independientemente de su localización.
- Durabilidad: capacidad de soportar los cambios tecnológicos sin necesidad de reajuste, reconfiguración o recodificación que impliquen un alto costo.
- Adaptabilidad: que se facilite la adaptación o personalización del entorno de aprendizaje.
- Productividad: si los proveedores de tecnología e-learning desarrollan sus productos siguiendo estándares comúnmente aceptados, la efectividad de e-learning se incrementa significativamente y el tiempo y costos se reducen.

Son varias las organizaciones que se han dedicado a la creación de especificaciones y estándares relacionados a las tecnologías aplicadas a los entornos *e-learning*, por la importancia que guarda la aplicación de estos estándares a los ROA, teniendo en cuenta que son los más utilizados a nivel mundial a continuación se exponen los de mayor relevancia para la presente investigación.

## Estándares y especificaciones de contenido

Nombre: **IEEE Standard for Learning Object Metadata (LOM)**

Organización: IEEE/LTSC<sup>14</sup>

**LOM**, es un modelo de datos que permite describir adecuadamente recursos digitales, normalmente usado en XML. Es el estándar de metadatos más aceptado en la comunidad *e-learning* según Hilera y Hoya (2010), es el principal modelo de referencia para la creación de subconjuntos de metadatos como Dublin Core<sup>15</sup>, CanCore, LOM-ES<sup>16</sup>. Su principal objetivo es la creación de descripciones estructuras que

---

<sup>14</sup> **IEEE/LTSC**, Learning Technology Standardization Committee (Comité de estandarización de tecnologías aplicadas al aprendizaje), perteneciente al Institute of Electrical and Electronics Engineers (IEEE). Su principal objetivo es desarrollar estándares, prácticas recomendadas y guías para componentes de software, herramientas, tecnologías y métodos de diseño que faciliten el desarrollo, implantación, mantenimiento e interoperabilidad de entornos educativos (). El área relacionada con los metadatos de los recursos educativos es donde mayor impacto ha tenido esta organización con la creación del estándar LOM ().

<sup>15</sup> El estándar The Dublin Core Metadata Element Set de ISO recoge el conjunto de metadatos definidos por la iniciativa Dublin Core, un foro abierto dedicado al desarrollo de estándares de metadatos de propósito general enfocado principalmente a la localización y catalogación de recursos. Es uno de los conjuntos de metadatos más utilizados en la actualidad, y muchos estándares de eLearning lo implementan. Reduce los

# CAPÍTULO 1. Fundamentación teórica

permiten conocer el contenido del OA sin necesidad de abrirlo. Esto facilita la búsqueda y catalogación de OA de forma rápida y eficiente.

Su modelo de datos especifica alrededor de 80 campos que deberían ser descritos en un OA, agrupados en 9 categorías, estas categorías son (González, 2008):

- General (Para aspectos generales como el título, palabras claves, etc.).
- Ciclo de Vida (Para tabular el ciclo de vida por el que pasa el OA).
- Meta-Metadatos (Aspectos relacionados con los metadatos versión, etc.).
- Aspectos Técnicos (Para información técnica como el formato, requisitos de software, etc.).
- Uso Educativo (Agrupa características de uso educativo como tipo de recurso, densidad semántica, tipo de usuario final, etc.).
- Derechos de Copia (Para aspectos de derecho de copia).
- Relación (Se utiliza para describir la relación entre el recurso en cuestión con otros recursos).
- Anotación (Registra las diferentes anotaciones que se le hacen al recurso o al OA que describe).
- Clasificación (Se utiliza para dar clasificaciones a los recursos que se describen).

Nombre: **Modelo Referenciado de Objetos de Contenido Compartible (SCORM).**

Organización: IMS<sup>17</sup>, AICC<sup>18</sup>, IEEE, ADL<sup>19</sup>.

---

más de setenta elementos de metadatos utilizados por el estándar LOM de IEEE a tan solo quince. El conjunto de elementos descrito por la iniciativa de Dublin Core es el siguiente: título, creador, tema, descripción, casa editorial, colaborador, fecha, tipo, formato, identificador, fuente, idioma, relación, cobertura, derechos.

<sup>16</sup> LOM-ES describe un perfil de aplicación que adapta al sistema educativo español el estándar LOM, este perfil ha sido realizado en el ámbito de la entidad de normalización AENOR. El perfil extiende LOM con nuevos elementos y vocabulario, lo que ha hecho necesario la nueva implementación de esquemas XML para el uso y adaptación de las herramientas de autorías existentes.

<sup>17</sup> **IMS**, IMS Global Learning Consortium es un grupo independiente, sin ánimo de lucro que inició su labor en 1997. Actualmente es el principal promotor de la creación de especificaciones abiertas dirigidas a la enseñanza electrónica. Su objetivo es que, a partir de estas especificaciones, se consiga la interoperabilidad de aplicaciones y servicios.

<sup>18</sup> **AICC**, Aviation Industry Computer Based Training Committee (Comité de la Industria de la Aviación para el aprendizaje basado en Computadoras), fue creado en 1992 y fue el primer organismo creado para desarrollar un conjunto de normas que permitiese el intercambio de cursos CBT (Computer Based-Training) entre diferentes sistemas. Fue la primera organización en separar los entornos de ejecución (definidos por AICC como Computer Management Instruction, CMI), y los contenidos entregados a los alumnos (Computer Based Training, CBT).

<sup>19</sup> **ADL**, Advanced Distributed Learning es una iniciativa creada por el departamento de Defensa de EEUU y la oficina de Ciencia y Tecnología de la Casa Blanca lanzada en Noviembre de 1997. ADL ha sido una de las organizaciones más activas en el esfuerzo de la estandarización de las

# CAPÍTULO 1. Fundamentación teórica

**SCORM**, es la unión de un conjunto de normas dirigidas al empaquetamiento, distribución, reutilización y ejecución de contenidos educativos, enfocado a lograr una mayor interoperabilidad entre las diferentes herramientas de un entorno *e-learning*. (Fernandez-Majón, 2006)

Esta norma cuenta con varias versiones, siendo la 1.2 y 2004 las más difundidas y utilizadas. SCORM está dividido en 3 libros técnicos, cada tópico de los que a continuación se especifican constituye un libro de SCORM (ADL, 2004).

- Modelo de Agregación de Contenidos (CAM): Tiene como objetivo principal ofrecer un medio común que permita contener recursos educativos desde diversas fuentes, compartibles y reusables. Define un contenido educativo puede ser identificado, descrito y agregado dentro de un curso o parte de él y como puede ser compartido por diversos LMS o repositorios (Hilera González, et al., 2010).
- Entorno de Tiempo de Ejecución (RTE), trata sobre 3 temas fundamentales: la administración del entorno en tiempo de ejecución, una API que permitirá programar las relaciones entre el contenido y el LMS, y un protocolo específico para el intercambio de contenidos web. También describe como el libro SCORM SN afecta al libro SCORM RTE.
- Secuencia y Navegación (SN), está basado en la especificación Simple Sequencing de IMS, y desarrolla un modelo para presentar los contenidos al alumno en función de las necesidades o capacidades del mismo. Describe las reglas de secuencia para que el desarrollador de contenidos logre trazar la ruta de lanzamiento de estos.

SCORM usa IMS-CP como modelo de agregación de contenido, como esquema de metadatos LOM u otro y la secuenciación de los contenidos está basada en los trabajos de la AICC en esta área.

## Estándar de interoperabilidad

Nombre: **Simple Query Interface Specification for Learning Repositories (SQI)**

---

tecnologías de aprendizaje, en colaboración con otras iniciativas principalmente IEEE, IMS y AICC. Su principal resultado es un conjunto de especificaciones que, bajo la denominación Shareable Content Object Reference Model.

# CAPÍTULO 1. Fundamentación teórica

Organización: CEN<sup>20</sup>

**SQLI** ofrece una interfaz de programación de aplicaciones (API), para la consulta de repositorios de OA. Esta interfaz debe ser programada y publicada como un servicio por el repositorio que sirve de fuente. Entre las ventajas que contemple SQLI, es la posibilidad de realizar búsquedas tanto asíncronas como síncronas, convirtiéndose en el “*mediador existente entre la fuente y el origen*” según Nguen (2007).

SQLI define un conjunto de métodos que pueden ser agrupados por la funcionalidad a la que dan soporte (Nguen, 2007):

- Manejo de Consultas: métodos que permiten la configuración de los parámetros de consultas: `setQueryLanguage()`, `setResultsFormat()`, `setMaxQueryResults()`, `setMaxDuration()`.
- Consultas Síncronas: los resultados de las consultas son directamente retornados por el método `synchronousQuery()`. Métodos adicionales permiten la elección de los números de resultados retornados por una consulta (`setResultsSetSize`) y además permiten conocer el número total de resultados de una consulta (`getTotalResultsCount`).
- Consultas Asíncronas: el API para este tipo de interacción define los siguientes métodos: `queryResultsListener` para ejecutar una consulta asíncrona, `setSourceLocation` para establecer la ubicación del cliente para que al terminar de realizar la consulta pueda entregar los resultados, `queryResultsListener` este método forma parte de un servicio que implementa el cliente (que se convierte en servidor en este caso) para que le sean entregados los resultados.
- Manejo de Sesión: define las funcionalidades para crear una sesión (`createSession()`), crear una sesión anónima o invitado (`queryResultsListener()`) y para destruir las sesiones creadas (`queryResultsListener()`).

## 1.9 Herramientas y tecnologías a utilizar

Para la realización del análisis y diseño de la versión 3.0 de RHODA se hace necesaria la especificación de un conjunto de tecnologías y herramientas. El análisis y diseño que se propone como solución en la

---

<sup>20</sup> **CEN**, Comité Europeo de Normalización es la organización regional europea de estandarización. Fue fundado en 1961 y su principal objetivo es fomentar la economía europea en el negocio global, el bienestar de ciudadanos europeos y el medio ambiente, proporcionando una infraestructura eficiente a las partes interesadas para el desarrollo, el mantenimiento y la distribución de sistemas estándares coherentes y de especificaciones. El CEN está formado por veintinueve miembros europeos y posee más de 60.000 expertos encargados de analizar las diferentes normas que afectarán a cerca de 460 millones de personas.

# CAPÍTULO 1. Fundamentación teórica

presente investigación, constituye una actualización y perfeccionamiento de las dos anteriores, desde el punto de vista arquitectónico los cambios no son notables, por lo que se ha decidido continuar con las especificaciones realizadas para la arquitectura en la versión 2.0.

## **Lenguaje de Modelado: UML 2.0**

El Lenguaje Unificado de Modelado permite visualizar, especificar, construir y documentar artefactos sobre todo durante el ciclo de desarrollo de un software. Este lenguaje no constituye una guía para la realización del análisis y diseño de un software, tampoco es un proceso sin embargo está fuertemente relacionado con la programación orientada a objetos.

UML propone un grupo de diagramas que se dividen en 2 categorías, diagramas de estructura y diagramas de comportamiento. Los diagramas o modelos de UML describen la estructura y comportamiento de un sistema, pero no indican cómo construirlo. A continuación se realizan un grupo de especificaciones relacionadas a elementos de UML que poseen gran relevancia dentro del modelado de la propuesta de solución.

**Diagrama de Casos de Uso:** muestra las distintas operaciones que se esperan de un sistema o aplicación y como se relaciona con los usuarios u otras aplicaciones. El diagrama se describe en función de *actores* y *casos de uso* y las relaciones que entre estos se establecen. Los casos de uso constituyen una útil técnica de captura de requisitos, a través de los mismos se puede visualizar los resultados de las acciones que se esperan de la aplicación, son muy útiles además para mostrar a los clientes rasgos distintivos de la aplicación (Larman, 2003).

**Diagrama de Clases:** muestra el conjunto de clases y de objetos importantes que forman partes de un sistema junto con las relaciones existentes entre clases y objetos. Modela los conceptos relacionados con el dominio de la aplicación. Muestra de una manera estática la estructura de información del sistema y la visibilidad que tiene cada una de las clases, dada por sus relaciones con los demás en el sistema (Larman, 2003).

**Diagrama de Colaboración:** muestra la forma en que los objetos colaboran entre sí, destacando el orden en que estos interactúan. Es una extensión de un diagrama de objetos, además de las relaciones entre los objetos muestra los mensajes que se envían entre sí (Larman, 2003).



# CAPÍTULO 1. Fundamentación teórica

**Diagrama de Despliegue:** muestra la configuración de los nodos de procesamiento en tiempo de ejecución, con sus respectivos procesos de software. En él se visualiza la distribución de componentes, así como la relación existente entre los elementos de procesamiento (Larman, 2003).

## **Metodología de desarrollo: Proceso Unificado de Desarrollo**

Rational Unified Process (Proceso Unificado de Desarrollo), es un conjunto de actividades que permiten transformar los requerimientos de un usuario en un sistema de software. RUP más que un proceso constituye un marco de trabajo genérico que puede especializarse para una gran variedad de: sistemas de software, áreas de aplicación, tipos de organizaciones niveles de aptitud y tamaños de software (Jacobson, et al., 2000).

El proceso unificado está basado en componentes, lo que significa que el software que se construye está conformado por componentes interconectadas a través del uso de interfaces. Este utiliza UML como lenguaje de modelado, de manera general con UML se crean todos los esquemas de un sistema de software, lo que se traduce en que UML es una parte esencial de RUP.

RUP posee tres aspectos que definitivamente lo hacen único e identificable en cualquier espacio: dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental (Jacobson, et al., 2000).

El proceso unificado de Rational está dividido en 4 fases: Inicio, Elaboración Construcción y Despliegue para las que se han definido un total de 9 actividades o flujos de trabajo, de ellas 6 denominadas de ingeniería y 3 de apoyo. Estos flujos de trabajo se desarrollan de forma paralela existiendo la presencia de un flujo de trabajo más que otro durante el avance de proceso de desarrollo de software distribuido en las diferentes iteraciones (Figura 3).

# CAPÍTULO 1. Fundamentación teórica

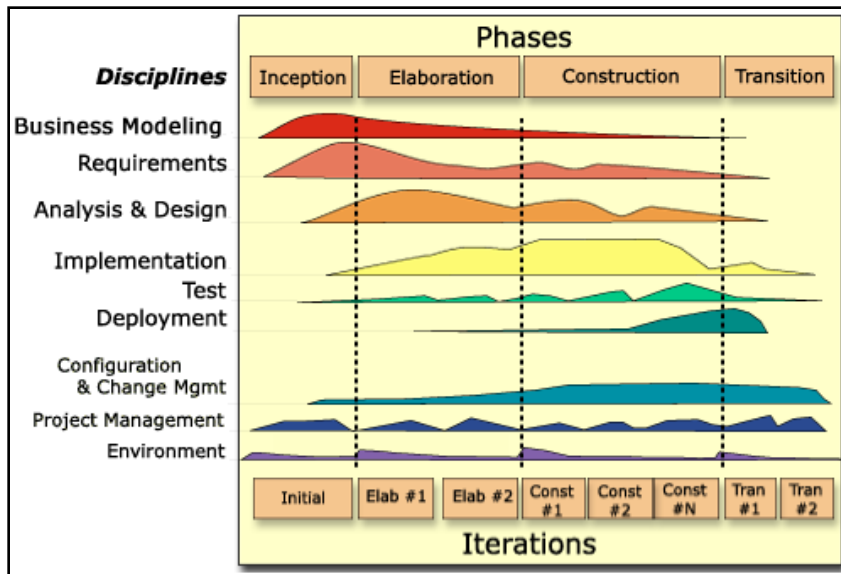


Figura 3. Ciclo de Vida de RUP (Jacobson, et al., 2000)

## Herramienta CASE: Visual Paradigm for UML 6.4 Enterprise Edition

Visual Paradigm (VP) permite el modelado y construcción de artefactos propuestos en el ciclo de vida de RUP usando UML como lenguaje de modelado, con soporte para este hasta su versión 2.1. Con VP es posible: una rápida construcción de aplicaciones de calidad, mejores y a un menor costo. Permite crear todos los tipos de diagramas de clases, generar diagramas a partir de código, generar código desde diagramas, generar documentación a partir de los artefactos creados, la creación de todos los diagramas propuestos por UML, modelar artefactos correspondientes a la captura de requerimientos, crear el modelo de datos integrándose a gestores como PostgreSQL, construir artefactos relacionados con la modelación del negocio y crear prototipos de interfaz de usuario.

También incluye la posibilidad de exportar los artefactos como archivos XML u otros formatos (jpg, png, pdf, EMS, SVG). VP posee un diseño de interfaz gráfica, enfocado a facilitar la interacción y el trabajo del usuario. La herramienta también proporciona abundantes tutoriales, demostraciones interactivas y proyectos de UML. Su capacidad para integrarse con distintos entornos de desarrollo (IDE) y de funcionar sobre plataformas GNU/Linux (Debian, Ubuntu) son aspectos que permiten engrosar su ya amplio grupo de funcionalidades.

# CAPÍTULO 1. Fundamentación teórica

## CMMI (Capability Maturity Model Integration)

El Modelo de Madurez de la Capacidad del Software (CMMI) del SEI<sup>21</sup> es un conjunto de modelos que permiten generar un diagnóstico preciso de la madurez de los procesos relacionados con las tecnologías de la información de una organización y describe las tareas que se tienen que llevar a cabo para mejorar dichos procesos. CMMI se desarrolló para facilitar y simplificar la adopción de varios modelos de forma simultánea y su contenido integra y da relevo a la evolución de sus predecesores<sup>22</sup> (SEI , 2010). En él se definen un conjunto de áreas claves del proceso (PA) que detallan las funciones de ingeniería de software que deben llevarse a cabo para el desarrollo de una buena práctica, estas áreas se encuentran agrupadas en 5 niveles inclusivos, que sirven de referencia para saber el estado de madurez del proceso del software en la organización.

CMMI es un modelo descriptivo, en el sentido que detalla los atributos esenciales que se esperan caractericen una organización dentro de un nivel de madurez en particular. Es un modelo normativo ya que las prácticas detalladas caracterizan el tipo de norma de comportamiento que se espera de una organización que realiza proyectos a gran escala, no es prescriptivo pues no dice a la organización como mejorar. Posee una estrecha relación con RUP, el hecho de utilizarlo como metodología de desarrollo permite que ciertas PA de CMMI sean alcanzadas automáticamente.

## PHP

PHP (acrónimo de Hipertext Preprocesor) es un lenguaje script interpretado que corre del lado del servidor en la Arquitectura Cliente - Servidor. Es uno de los lenguajes preferidos por los desarrolladores de páginas web dinámicas. Por ser libre y disponer de acceso pleno al código fuente, además de ser gratuita su distribución, ha posibilitado el surgimiento y consolidación de una gran comunidad de programadores que le dan soporte y contribuyen a la mejora de su código. Posee gran eficiencia cuando se combina con el servidor web Apache, soportando millones de visitas diarias sin complicaciones (PHP, 2010).

---

<sup>21</sup> SEI es un centro de investigación y desarrollo perteneciente a la Universidad de Carnegie Mellon, fundado y financiado por el Departamento de Defensa de los Estados Unidos, a través de la Oficina de la Subsecretaría de Defensa para la Adquisición, Tecnología y Logística. Su meta es proporcionar a las organizaciones las pautas de actuación necesarias para obtener mejoras observables en su proceso del software, de manera que desarrollen productos sin defectos respetando requerimientos, fechas y costes.

<sup>22</sup> CMM – SW (CMM para el software), SE – CMM (CMM para la ingeniería de sistemas), IPD – CMM(CMM para el desarrollo integral de productos)

# CAPÍTULO 1. Fundamentación teórica

PHP dispone de una conexión propia a todos los SGBD. Además de MySQL, puede conectarse directamente a las bases de datos de PostgreSQL, Oracle, etc. Es multiplataforma e incorpora una gran cantidad de funciones integradas para realizar útiles tareas relacionadas con la web. La sintaxis de PHP se basa en otros lenguajes de programación, principalmente en C y Perl. PHP está disponible para una gran cantidad de sistemas operativos diferentes. Su código funciona sin necesidad de aplicar ninguna modificación a los diferentes sistemas en que se ejecute.

## **Framework de desarrollo: Symfony 1.4**

La construcción de aplicaciones web como los ROA cada día se torna un poco más compleja, debido a la gran cantidad de funcionalidades que en estas se manejan. El uso de un framework simplifica la construcción de una aplicación mediante la automatización de patrones utilizados para resolver tareas comunes, proporciona estructura al código fuente y facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas.

En la UCI se ha hecho común el uso de frameworks de desarrollo para acometer tareas esenciales como la producción de software. En los proyectos productivos relacionados a aplicaciones web se hace imprescindible por las características antes mencionadas el uso de este tipo de sistemas (frameworks), habiéndose expandido Symfony por las ventajas que presenta.

Symfony es un completo framework diseñado para optimizar el desarrollo de aplicaciones web. Posee atributos que han contribuido a su aceptación, este separa la lógica del negocio, la lógica de servidor de la presentación de la aplicación web, proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja (SENSIOLABS, 2010), cuenta con diferentes entornos de ejecución, siendo los más utilizados e importantes el entorno de Desarrollo y el de Producción, el primero posee un potente sistema de depuración y el segundo se utiliza en la ejecución de aplicaciones ya en producción. Symfony fue diseñado con el objetivo de que se ajustara a los siguientes requisitos y que hoy forman parte de sus características principales (Potencier, et al., 2005):

- Fácil de instalar y configurar en la mayoría de plataformas.
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.

# CAPÍTULO 1. Fundamentación teórica

- Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales, adaptable a las políticas y arquitecturas propias de cada empresa, suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.
- Symfony está desarrollado completamente con y para PHP 5 y su interacción con Ajax es muy fácil de implementar mediante los helpers que permiten encapsular los efectos de JavaScript.

Las anteriores versiones de RHODA fueron desarrolladas utilizando este framework en su versión 1.2, la presente versión será desarrollada utilizando la versión 1.4 que incorpora mejoras potenciales de las características antes mencionadas.

## **Base de datos nativas XML**

Las bases de datos son una parte fundamental de todas las organizaciones, en el mundo de la programación de aplicaciones web se ha hecho casi imprescindible y en el área de los repositorios de OA constituye el corazón de la aplicación. En un ROA una de las principales funcionalidades, radica en la búsqueda de OA y recursos digitales a través de los metadatos que lo describen, lo que hace necesario la indexación de estos metadatos que vienen dados en formato XML. La alternativa más útil en este caso la constituye las bases de datos nativas en XML (Azcárate, 2008).

No existe una definición estándar de una base de datos nativa en XML, pero la organización XML:DB Initiative for XML Databases describe una base de datos de este tipo como un: modelo lógico para documentos XML que almacena y recupera documentos de acuerdo a dicho modelo.

Entre las ventajas con que se cuenta al utilizar este tipo de base de datos figuran: no necesitan mapeo adicional, conservan la integridad de los documentos y permiten almacenar documentos heterogéneos en la misma colección (Gaona, 2008). Además permiten que se realice un uso completo de XML como tecnología, ya que es posible la utilización de XPath y XQuery como lenguajes de consulta, estos devuelven los resultados en formato XML, y XSL para la transformación de los resultados.

# CAPÍTULO 1. Fundamentación teórica

Entre las bases de datos XML más utilizadas (DBCOM, eXist, dbXML, Tamino XML Server) se puede mencionar por la importancia que posee para la presente investigación: eXist, esta ha sido la utilizada en las anteriores versiones del repositorio de la UCI, y los resultados han sido satisfactorios. Por las características que esta entraña y que a continuación se describen, ha sido seleccionada para el desarrollo de la versión 3.0 de RHODA.

## **eXist**

eXist es una base de datos nativa en XML, muy usada en actuales desarrollos de aplicaciones web. Entre las características más distinguibles de eXist se encuentran (Azcárate, 2008):

- Base de datos diseñada para manejar data enteramente en XML.
- Totalmente compatible con el estándar Xquery.
- Posee varios tipos de índices, incluso para dato espacial.
- Implementación de transacciones que cumplen ACID probada.
- Hecho y pensado para trabajar con Java, puede trabajar como un servlet embebido en una aplicación Java.
- Soporta conexión a otros lenguajes.
- Incluye su administrador web propio por defecto.
- Máximo número de elementos  $2^{31}$  (2mil millones), cada elemento sin límite teórico de tamaño pero dependerá del sistema de archivos en el que esté instalado.

## **PostgreSQL**

PostgreSQL es un SGBD multiplataforma muy conocido y usado en entornos de software libre. Es distribuido bajo licencia BSD, la que permite su libre uso, modificación y redistribución con la única restricción de mantener el copyright del software original a sus autores. Es un sistema de bases de datos objeto-relacional con una probada arquitectura que ha ganado gran reputación por su confiabilidad, estabilidad y mantenimiento de la integridad de los datos (PostgreSQL Global Development Group, 2010).

Incluye extensiones de orientación a objetos, permitiendo definir un nuevo tipo de tabla a partir de una previamente definida. Posee interfaces de programación nativas para diversos lenguajes y plataformas como Java, .Net, C/C++, Perl, Python, así como ODBC, y es capaz de ejecutar procedimientos

# CAPÍTULO 1. Fundamentación teórica

almacenados en muchos de estos lenguajes, y en su propio lenguaje PL/pgSQL (PostgreSQL Global Development Group, 2010).

Presenta otras características como el control de concurrencia multi-versión (MVCC), los puntos de recuperación dentro de las transacciones, replicación asincrónica, conjunto de caracteres internacionales. Las características que brinda PostgreSQL permiten que sea de gran acogida por la comunidad de desarrolladores. El análisis detallado de sus atributos respecto a otros SGBD fue lo que influyó para que fuese seleccionado para formar parte de las herramientas y tecnologías a utilizar para el desarrollo de la versión 3.0 de RHODA.

## Arquitectura

La arquitectura de software es la organización fundamental de un sistema y está basada en los componentes, las relaciones entre ellos, el ambiente y los principios que orientan su diseño y evolución. Esta establece algunos conceptos de mucho valor, para que analistas, diseñadores, programadores puedan trabajar sobre una misma línea, permitiendo alcanzar los objetivos previstos y cubriendo prácticamente todas las necesidades.

La clave del trabajo arquitectónico tiene que ver con una correcta elección del estilo arquitectónico. Los estilos arquitectónicos son arquitecturas de software comunes, marcos de referencias arquitectónicas, formas comunes o clases de sistemas (Schmidt, et al., 2000). Los estilos de arquitectura se definen como las 4C (Schmidt, et al., 2000):

- Componentes.
- Conectores
- Configuraciones
- Restricciones (Constraints).

Cuando se define un estilo, es necesario tener en cuenta el tipo de aplicación, ya que pueden imponer restricciones que acotan la interacción de los componentes. Algunos de los principales estilos que se usan en la actualidad están divididos por clases de estilos, las que engloban una serie de estilos arquitectónicos:

- Estilos de flujo de datos.

# CAPÍTULO 1. Fundamentación teórica

- Estilos centrados en datos.
- Estilos de llamada y retorno.
- Estilos de código móvil.
- Estilos heterogéneos.
- Estilos Peer to Peer.

Formando parte de los estilos de llamada y retorno se puede encontrar el denominado Modelo – Vista – Controlador (MVC) el cuál ha constituido el estilo adoptado en las anteriores versiones de RHODA y que a continuación se describe.

## **Modelo – Vista – Controlador**

Reconocido como estilo por Taylor y Medvidovic (Schmidt, et al., 2000), en ocasiones es tomado más bien como un patrón de diseño o como una práctica recurrente, mientras Buschman y otros autores lo consideran un patrón correspondiente al estilo de los sistemas interactivos (CyberFicient Technologies Inc., 2001). Este estilo es muy usado en las aplicaciones web aunque no es exclusivo de estas. El estilo define el *modelo* (modelo de datos), la *vista* (visualización de la información) y la *lógica de control* (lógica del negocio) en 3 componentes distintos, los cuales interactúan entre sí para dar una salida final al usuario, donde cada componente conoce a los demás solo lo necesario. Por lo que una solución basada en este estilo es muy flexible. Permite soporte a vistas múltiples dado que la se halla separada del modelo y no hay una dependencia directa del modelo respecto a la vista, por lo que es posible que se puedan mostrar distintas vistas de los mismos datos simultáneamente, lo que evidencia la fácil adaptación a los cambios.



# CAPÍTULO 2. Propuesta de solución. Análisis y diseño

## Capítulo 2. Análisis y Diseño

### Introducción

En el presente capítulo se expone la propuesta de análisis y diseño de la versión 3.0 de RHODA, donde se incluye el modelo de dominio, los requerimientos, identificación de casos de usos del sistema, diagramas y descripción de los casos de uso arquitectónicamente significativos, la definición de roles, diagramas de: clases del análisis, colaboración, clases del diseño, así como los patrones de diseño que se utilizan.

### 2.1 Modelo de dominio

Un modelo de dominio constituye una especificación de los conceptos más importantes de la vida real que se utilizarán en el sistema, *“una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés”* (Fowler, et al., 1996). RUP define la creación de un modelo de dominio como uno de los artefactos a generar, durante el flujo de trabajo: Modelado del negocio.

El modelo de dominio se construye utilizando la notación UML y se representa a través de un conjunto de diagramas de clases, en los que no hay definidas operaciones (Larman, 2003) y puede mostrar:

- Objetos de dominio o clases conceptuales.
- Asociaciones entre las clases conceptuales.
- Atributos de las clases conceptuales.

Se puede deducir que el modelo de dominio es un diccionario visual de las abstracciones relevantes, vocabulario e información del dominio. La identificación y asignación de nombres a estos conceptos permite describir estructuralmente el negocio que se modela y crear un glosario de términos que facilita la comprensión del modelo y logra un entendimiento entre clientes y equipo de desarrolladores.

El modelo de dominio que a continuación se presenta constituye una representación gráfica de los conceptos más importantes que forman parte de la versión 3.0 de RHODA.

#### 2.1.1 Definición de conceptos fundamentales

### Repositorio

## **CAPÍTULO 2. Propuesta de solución. Análisis y diseño**

Espacio donde se almacenan y gestionan los OA. Es donde interactúan los usuarios con los diferentes servicios.

### **Usuario**

Representa los usuarios del sistema que están divididos en 5 roles diferentes: Administrador, Revisor, Autor, Usuario, Invitado.

### **Rol**

Permite identificar de manera única el conjunto de privilegios que pueda tener un usuario en el sistema. Todo usuario del repositorio debe tener al menos un rol.

### **Categoría**

Representa las clasificaciones según las cuales se puede catalogar un OA según sus contenidos. Cada categoría posee una o más subcategorías.

### **OA**

Acrónimo utilizado para definir un OA.

### **Foro**

Representa el área de intercambio de opiniones de los usuarios, cada una de estas áreas está organizada por hilos de discusión, los que a su vez están formados por temas y estos últimos por comentarios.

### **Comentario**

Opiniones de los usuarios del repositorio relacionadas a un OA. Estos son revisados y posteriormente publicados en el sistema.

### **Mensaje**

Servicio de notificaciones usadas en el repositorio, estas notificaciones son enviadas a los usuarios usando la mensajería interna de la plataforma.

### **RSS**

Canal seguro para la transmisión de datos. Servicio que utiliza el repositorio y que permite a los usuarios poder sindicarse a distintos tipos de notificaciones, las que son recibidas en su buzón de correo externo.

### **Blog**

## **CAPÍTULO 2. Propuesta de solución. Análisis y diseño**

Servicio de creación de bitácoras online que brinda el repositorio. Es usado por los usuarios de la plataforma para la creación de páginas personales con un marcado interés educativo y con la finalidad de socialización del conocimiento.

### **Versión**

Son estados por los que ha transitado un OA.

### **Notifica**

Acción que realiza el repositorio, por lo general se trata de información importante para un usuario, o información que el usuario ha decidido que desea recibir en su buzón de correo externo.

### **Mensajería Externa**

Servicio que posee el repositorio mediante el cual, es posible enviar a los usuarios información relevante que deben conocer o que éstos solicitaron durante el registro.

### **Mensajería interna**

Servicio que posee el repositorio, al cual son enviadas algunas de las notificaciones que genera el repositorio. Cada usuario posee un buzón de mensajería interna, y el repositorio tiene definidas qué notificaciones envía a este buzón.

### **Búsqueda**

Servicio que brinda el repositorio, este es utilizado por los usuarios con el fin de localizar un OA o información detallada que se encuentra en su base de datos. Permite la realización de varias acciones de gran utilidad para los usuarios.

### **Wiki**

Herramienta para el trabajo colaborativo, permite la creación de páginas web, persigue el objetivo de socialización del conocimiento, posibilitando que los usuarios puedan editar estas páginas publicando aspectos que pueden ser de interés para los miembros de repositorio.

### **Log**

Fichero mediante en el cual se almacenan todas las acciones y sucesos generados en la plataforma, sirve de soporte para realizar auditorías, detectar acciones indebidas e intrusiones no autorizadas.

### **Herramienta colaborativa**

## CAPÍTULO 2. Propuesta de solución. Análisis y diseño

Conjunto de herramientas denominadas: Herramientas de la Web 2.0, que permiten la interacción de los usuarios y la colaboración de unos con otros a fin de lograr un objetivo específico, entre estas herramientas se pueden nombrar las Wikis, los Foros y los Blogs.

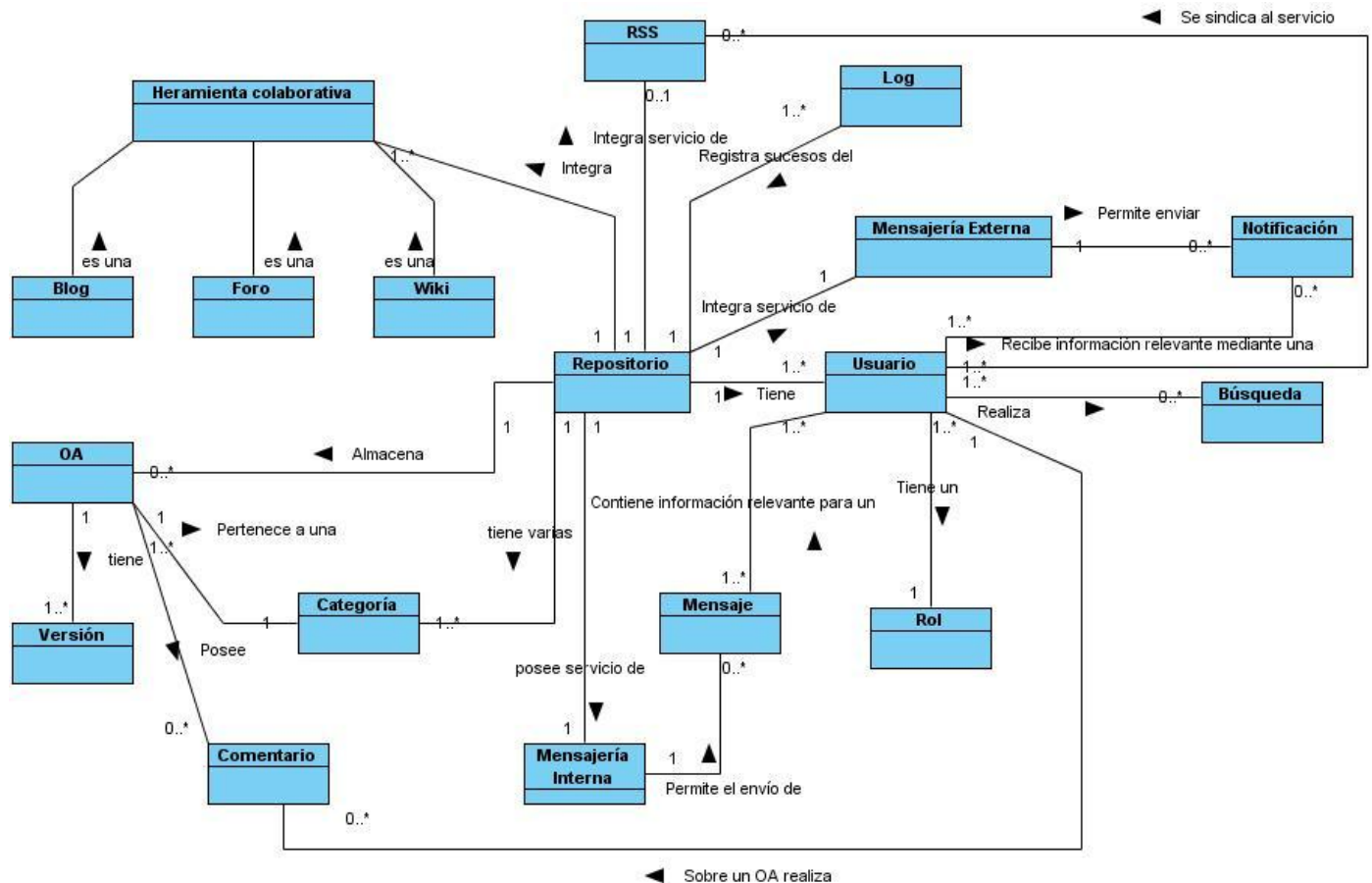


Figura 4. Modelo de dominio de RHODA 3.0

### 2.2 Requerimientos

Los requerimientos de software constituyen capacidades y condiciones con las cuales debe cumplir el sistema. Estos permiten “encontrar, comunicar y recordar lo que se necesita realmente, de manera que tenga un significado claro para el cliente y los miembros del equipo de desarrollo” (Larman, 2004). Los requerimientos o requisitos como normalmente se les conoce, tienen varias clasificaciones, según el modelo FURPS (Grady, 1992) se distinguen 5 tipos de requisitos: Funcionales, Usabilidad, Fiabilidad, Rendimiento y Soporte.

## CAPÍTULO 2. Propuesta de solución. Análisis y diseño

En la actualidad una práctica muy difundida es clasificar los requisitos en 2 categorías: funcionales y no funcionales. Los funcionales describen lo que el sistema debe hacer, describen de forma detallada las funciones de este, sus entradas, salidas y excepciones. (Sommerville, 2005). Los requisitos no funcionales engloban al resto de las clasificaciones dadas en el modelo FURPS, constituyen cualidades o propiedades que el sistema debe tener, por lo general especifican o restringen las cualidades de un sistema o producto.

De acuerdo a lo antes mencionado para el modelado de la versión 3.0 de RHODA se proponen un total de 113 requisitos funcionales, los mismos son enunciados en lenguaje natural en forma ascendente, según los roles de los usuarios en la plataforma (Figura 5), los no funcionales se encuentran descritos en el [Anexo I](#).

### Requisitos funcionales para RHODA 3.0

#### Invitado

1. Ver perfil de los usuarios.
2. Registrar usuario.
3. Registrar usuario de forma manual.
4. Registrar usuario a través de directorio activo.
5. Registrar usuario a través de correo electrónico.
6. Autenticar usuario.
7. Listar usuarios en línea.
8. Listar categorías existentes en el sistema.
9. Listar todos los usuarios.
10. Listar las publicaciones recientes.
11. Listar OA más solicitados.
12. Listar OA más visualizados.
13. Listar más descargados.
14. Descargar OA.
15. Recuperar contraseña y nombre de usuario a través de correo electrónico.
16. Realizar búsqueda básica.
17. Realizar búsqueda por metadatos.

## **CAPÍTULO 2. Propuesta de solución. Análisis y diseño**

18. Realizar búsqueda avanzada.
19. Configurar parámetros a integrarse en la búsqueda avanzada.
20. Guardar parámetros de búsqueda realizada en historial de búsquedas.
21. Editar búsquedas del historial.
22. Borrar historial de búsquedas.
23. Refinar resultados de una búsqueda.
24. Visualizar ficha técnica por elementos arrojados en una búsqueda.
25. Seleccionar formato del resultado de una búsqueda.

### ***Usuario***

26. Ver mi perfil.
27. Editar mi perfil de usuario.
28. Re - autenticar usuario.
29. Agregar OA a mis favoritos.
30. Excluir OA de mis favoritos.
31. Sindicarse a una subcategoría.
32. Elegir el tiempo de envío de las notificaciones del sistema.
33. Sindicarse a canales RSS.

### ***Autor***

34. Listar mis publicaciones.
35. Listar OA favoritos.
36. Importar OA.
37. Listar OA en edición.
38. Modificar las propiedades no técnicas de un OA.
39. Congelar envíos de miembro de grupo de trabajo colaborativo para la creación de OA.

### ***Revisor***

40. Listar OA en revisión.
41. Revisar OA.
42. Publicar OA revisado.

## CAPÍTULO 2. Propuesta de solución. Análisis y diseño

43. Enviar a edición OA revisado.
44. Revisar comentarios realizados a OA publicados.
45. Publicar comentarios revisados.
46. Regresar a edición un comentario revisado.
47. Eliminar comentarios.

### **Administrador**

48. Seleccionar tipo de registro.
49. Consultar historial de un revisor.
50. Limitar capacidad de almacenamiento de los OA en el servidor.
51. Gestionar tamaño de un OA.
52. Configurar conexión con un directorio activo.
53. Configurar servicio de envío de correo electrónico SMTP.
54. Configurar servicio de envío correo electrónico POP.
55. Configurar página a mostrar cuando se realizan acciones de mantenimiento en la plataforma.
56. Listar comprobaciones de estado del servidor.
57. Editar configuración de la conexión a la base de datos PostgreSQL.
58. Editar configuración de la conexión a la base de datos nativa xml eXist.
59. Listar OA para realizar acciones de Administración.
60. Ordenar OA a administrar por atributos.
61. Filtrar OA listados de acuerdo a sus atributos.
62. Administrar propiedades de OA.
63. Eliminar OA.
64. Configurar almacenamiento de los Logs.
65. Configurar la rotación de los Logs.
66. Filtrar los Logs.
67. Exportar Logs.
68. Configurar aspectos de seguridad del sistema relacionados a la contraseña de usuario.
69. Editar el perfil de un usuario.
70. Administrar solicitudes realizadas por los usuarios.

## CAPÍTULO 2. Propuesta de solución. Análisis y diseño

- 71. Gestionar categorías y subcategorías.
- 72. Gestionar roles de las categorías.
- 73. Administrar cuentas de usuario.
- 74. Crear cuenta a un usuario determinado.

### ***A ser generados por la plataforma***

- 75. Informar a través del uso de la mensajería externa los OA más recientes.
- 76. Informar a través del uso de la mensajería externa los OA más solicitados.
- 77. Recomendar la utilización de otros OA teniendo en cuenta el perfil del usuario.
- 78. Informar a través de la mensajería interna y externa sobre comentarios a OA de su autoría.
- 79. Informar a través de la mensajería externa al autor cuando sea rechazado un OA.
- 80. Informar a través de la mensajería externa al autor cuando sea aceptado un OA de su autoría.
- 81. Informar a través de la mensajería externa al usuario antes de que su contraseña caduque.
- 82. Informar al usuario a través de la mensajería externa, que forma parte de un grupo de trabajo para la creación colaborativa de un OA.
- 83. Informar a través de la mensajería interna a los miembros de un grupo de trabajo colaborativo cuando se añade un nuevo miembro a este.
- 84. Informar a través de la mensajería interna, cuando se elimina algún miembro del grupo de trabajo colaborativo.
- 85. Informar a través de la mensajería interna y externa, al usuario, que ha sido eliminado del grupo de trabajo colaborativo.
- 86. Informar a través de la mensajería interna cuando sea actualizada la versión de un OA en edición, que se crea de forma colaborativa.
- 87. Informar a través de la mensajería interna, a los miembros de un grupo de trabajo colaborativo cuando sus envíos sean congelados por el coordinador del grupo.
- 88. Informar a través de la mensajería interna a los miembros de un grupo de trabajo colaborativo, cuando la creación del OA haya concluido.
- 89. Enviar notificación RSS sobre OA seleccionados como favoritos.
- 90. Enviar notificación RSS sobre objetos de aprendizaje en edición.
- 91. Enviar notificación, a la mensajería externa de los usuarios, sobre publicación de OA de su autoría.



## CAPÍTULO 2. Propuesta de solución. Análisis y diseño

92. Enviar notificación, a la mensajería externa de los usuarios, sobre regreso de OA de su autoría a edición.
93. Enviar notificación, a la mensajería interna de los usuarios, sobre comentario de su autoría publicado.
94. Enviar notificación, a la mensajería interna de los usuarios, sobre comentario de su autoría regresado a edición.
95. Notificar, a la mensajería interna de los usuarios, eliminación de comentario realizado a un OA.
96. Enviar notificación RSS a los usuarios sobre OA más visualizados.
97. Enviar notificación RSS a los usuarios sobre OA más descargados.
98. Enviar notificación RSS a los usuarios sobre OA más solicitados.
99. Notificar a la mensajería externa del administrador del sistema sobre solicitud existente en su buzón de mensajería en la plataforma.
100. Notificar al usuario a través de la mensajería interna, sobre solicitud de su autoría aceptada.
101. Notificar al usuario a través de la mensajería interna sobre solicitud de su autoría rechazada.
102. Notificar, a usuarios que poseen OA en una categoría cuando esta sea modificada, a través de la mensajería externa.
103. Notificar a usuarios que poseen OA en una categoría cuando esta sea eliminada, a través de la mensajería externa.
104. Notificar a usuarios que poseen OA en una subcategoría cuando esta sea modificada, a través de la mensajería externa.
105. Notificar a usuarios que poseen OA en una subcategoría cuando esta sea eliminada, a través de la mensajería externa.
106. Informar a través de la mensajería interna, al usuario, cuando sea removido del rol que estaba ocupando en una categoría.
107. Informar a través de la mensajería interna, al usuario cuando su cuenta ha sido bloqueada o desbloqueada.
108. Informar a través de la mensajería externa, al usuario cuando el administrador del sistema ha creado una cuenta para él en el sistema.
109. Permitir conservar búsquedas de un usuario una vez cerrada sesión.
110. Sugerir resultados similares al de las búsquedas realizadas.

## CAPÍTULO 2. Propuesta de solución. Análisis y diseño

- 111. Visualizar estadísticas del resultado de una búsqueda.
- 112. Resaltar vocablos coincidentes en el resultado de una búsqueda.
- 113. Indexar OA y sus componentes una vez que estos se publican en el sistema.

### 2.3 Identificación de roles, actores y casos de uso del sistema

En productos de software como RHODA, es muy importante tener clara la definición de cuáles son los roles que se van a incluir, esta definición permite crear un esquema de seguridad basado en roles, con el cual, se pueden definir: privilegios, nivel de acceso a la información y funciones de la plataforma. Los roles en RHODA se muestra la siguiente figura:

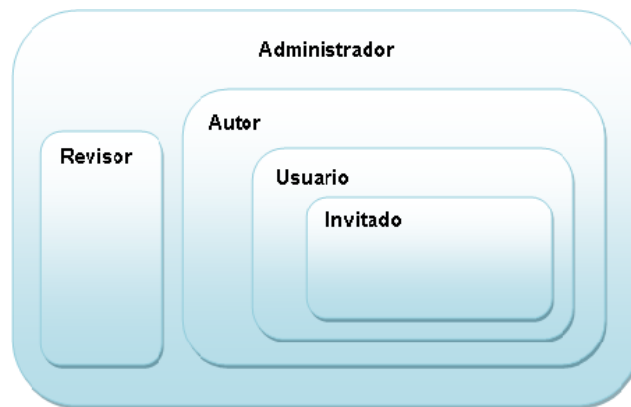


Figura 5. Jerarquía de roles en RHODA 3.0

En este caso se tienen 5 roles: Invitado, Usuario, Autor, Revisor y Administrador. Tal y como se indica en la figura el rol que cuenta con los menores privilegios es el Invitado, sus acciones en la plataforma son mínimas. El Autor es uno de los roles de mayor categoría, puede realizar todas las acciones que engloba poseer el rol de Invitado y Usuario. El Revisor es un rol que parecido al Autor posee todos los privilegios que implica ser Usuario, pero no puede realizar ninguna de las acciones que son propias de los autores, a menos que tenga ambos roles y viceversa. El administrador, posee todos los privilegios posibles en el sistema, es el rol de mayor rango. Una vez definidos los roles se procede a especificar los actores del sistema, los cuales tendrán correspondencia con los roles establecidos.

Un actor es algo con comportamiento, los actores no son solamente los roles que juegan las personas, sino también organizaciones, software y máquinas. En el caso de la versión 3.0 de RHODA, cada rol representa un actor del sistema, incluyendo actores que constituyen sistemas de software que serán integrados a la plataforma y que tienen funcionamiento propio. Por tanto, se está en presencia de 10

## CAPÍTULO 2. Propuesta de solución. Análisis y diseño

actores: Invitado, Usuario, Autor, Revisor, Administrador y los actores de apoyo: Servidor, Foro, Wiki, Blog, Reloj. Estos se relacionan con las funcionalidades de la plataforma, según lo especificado para cada uno de los requerimientos, a través de los casos de uso (CU).

Los CU constituyen una colección de escenarios con éxito y fallo relacionados, estos describen a los actores utilizando un sistema para satisfacer un objetivo, sitúan las características y funciones en un contexto orientado al objetivo (Cockburn, 2000). Normalmente los CU son requisitos, por lo general funcionales, pero no todos los requisitos son casos de uso, debido a que un mismo CU puede englobar varios requisitos (Sommerville, 2005).

A continuación se relacionan los casos de uso más significativos; los cuales serán incluidos en la línea base de la arquitectura; forman parte de las funcionalidades que se pretenden someter a un proceso de perfeccionamiento y de aquellas que se integran nuevas a la plataforma.

- CU 1. Incluir OA en los Favoritos.
- CU 2. Gestionar intereses.
- CU 3. Gestionar categorías.
- CU 4. Gestionar solicitudes.
- CU 5. Revisar OA.
- CU 6. Editar perfil.
- CU 7. Seleccionar tipo de autenticación.
- CU 8. Configurar almacenamiento de los Logs.
- CU 9. Configurar rotación de los Logs.
- CU 10. Generar comprobaciones del servidor.
- CU 11. Configurar conexión con directorio activo.
- CU 12. Configurar correo electrónico SMTP.
- CU 13. Importar un OA.
- CU 14. Listar OA en revisión.

## CAPÍTULO 2. Propuesta de solución. Análisis y diseño

CU 15. Configurar búsqueda.

CU 16. Guardar búsqueda en historial.

CU 17. Realizar búsqueda avanzada.

CU 18. Realizar búsqueda por metadatos.

### Patrones de casos de uso

Durante la estructuración del modelo de CU, se tuvo en cuenta la aplicación de patrones de CU, buenas prácticas recomendadas por varios autores (Övergaard, et al., 2004). Un patrón por lo general es un diseño probado, junto con una descripción del contexto en que se va a utilizar y qué consecuencias tendrá su aplicación en el modelo (Övergaard, et al., 2004).

Su aplicación permite una descripción profunda de la interacción actor-sistema, permitiendo además poder crear una estructura fácilmente entendible por los clientes.

Una vez obtenido el modelo de CU inicial, se decidió aplicar sobre el mismo algunos patrones, estos permitieron definir nuevos CU en función de crear un modelo más sencillo y estructurado. Entre los patrones aplicados se pueden mencionar (Övergaard, et al., 2004):

- **Reglas de negocio: Definición estática**

Este patrón no tiene influencia sobre la estructura del modelo de CU, sino sobre la descripción del CU. Se basa en la especificación de reglas establecidas por la organización y que deben de ser cumplidas a fin de lograr un correcto y solicitado funcionamiento del sistema.

- **Secuencia común: Reusabilidad**

Este patrón está formado por al menos 3 CU, el primero de estos se denomina: Sub - secuencia común y modela una secuencia común de eventos que anteriormente ocurrían en el resto de los CU que forman el patrón. En este patrón la sub-secuencia común debe estar contenida en una sola pieza y tener relación de inclusión con los restantes CU (Fig 6).

## CAPÍTULO 2. Propuesta de solución. Análisis y diseño

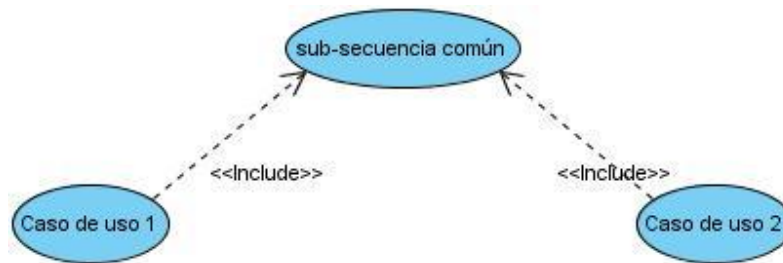


Figura 6. Secuencia común: Reusabilidad

- **Secuencia común: Adición**

Patrón formado igualmente que el anterior por al menos 3 CU, el que modela la sub-secuencia común contribuye a expandir al resto de los CU, que no necesitan la sub-secuencia para modelar su flujo de eventos completo, sino que la utilizan como una expansión no obligatoria (Figura 7).

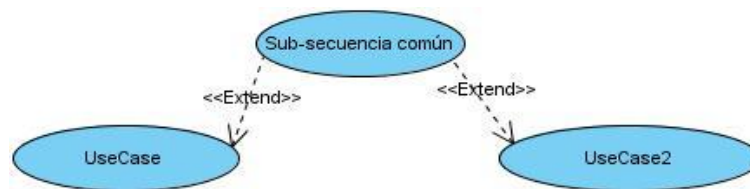


Figura 7. Secuencia común: Adición

- **CRUD Completo (Crear, Leer, Actualizar, Eliminar)**

También llamado CRUD información o Gestionar Información, modela todas las operaciones que pueden ser desarrolladas sobre una pieza de información: crear, leer, actualizar y eliminar (Figura 8).



Figura 8. CRUD Completo

- **CRUD Parcial**

Es utilizado cuando se decide separar una de las acciones debido a que esta es más importante, larga o mucho más compleja que el resto de las acciones (Figura 9).

## CAPÍTULO 2. Propuesta de solución. Análisis y diseño

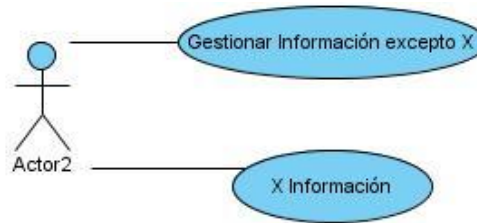


Figura 9. CRUD Parcial

- **Múltiples actores: Roles comunes**

Este patrón es aplicable cuando los actores juegan roles comunes durante el desempeño de determinadas tareas, en este caso, los actores heredan el comportamiento del rol común cuando interactúa con un determinado CU (Figura 10).

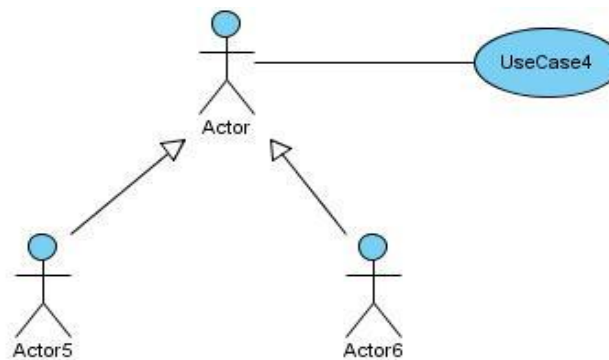


Figura 10. Múltiples actores: Roles comunes

- **Múltiples actores: Roles diferentes**

Este patrón está estructurado por un CU y al menos 2 actores. Es usado cuando ambos actores interactúan con un CU de forma distinta, es decir, estos actores juegan roles diferentes dentro del sistema (Figura 11).

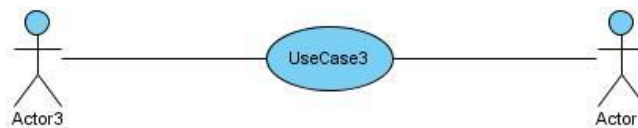


Figura 11. Múltiples actores: Roles diferentes

## CAPÍTULO 2. Propuesta de solución. Análisis y diseño

### 2.4 Diagramas de casos de uso

Serán mostrados los diagramas de casos de uso, según los módulos donde estos realizan sus funciones, en este caso los diagramas de CU abarcan no solo los arquitectónicamente significativos, sino todos los que contribuyen a las funcionalidades que se pretenden incorporar y perfeccionar en la versión 3.0 de RHODA.

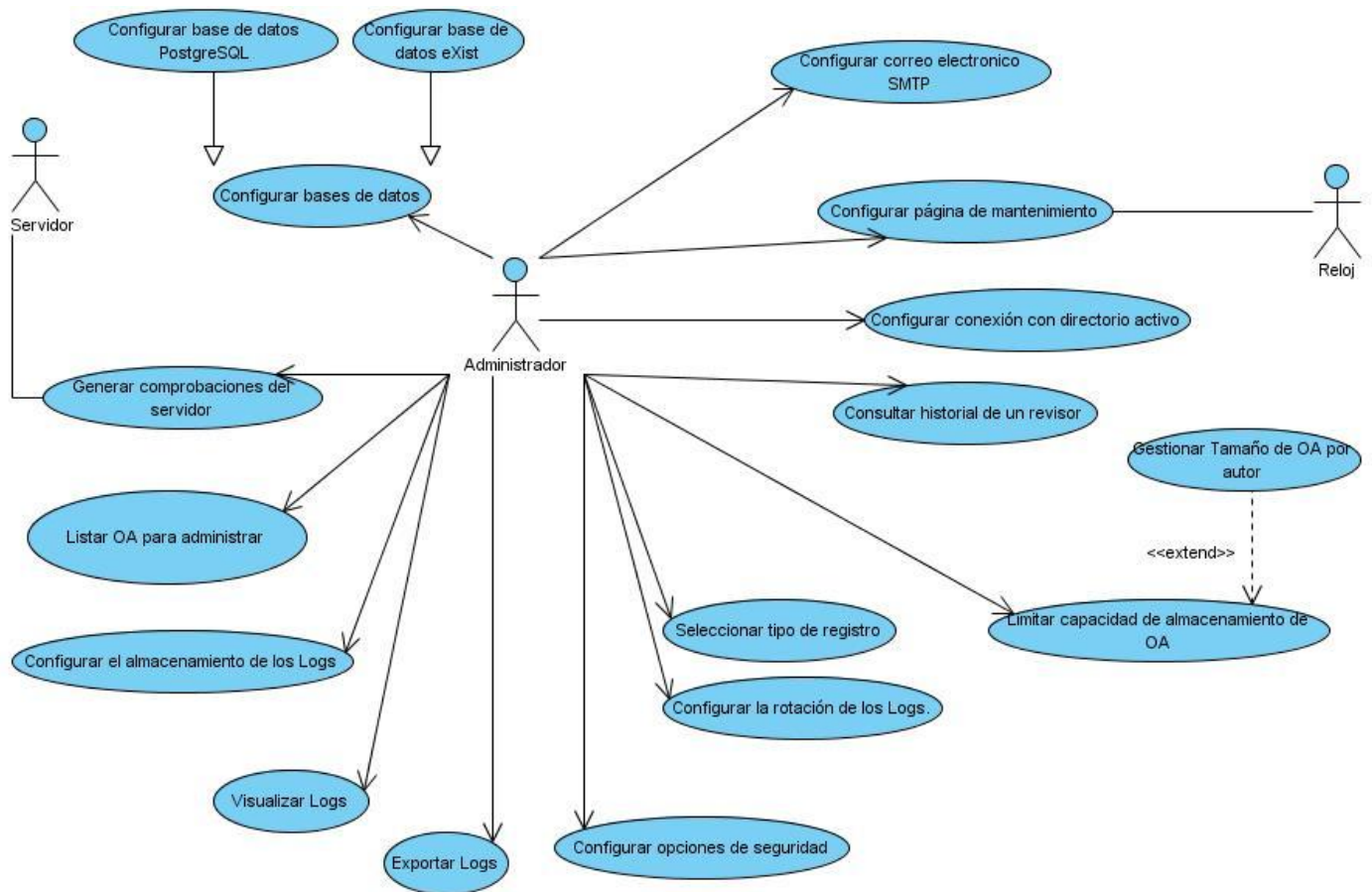


Figura 12. Diagrama de casos de uso: Módulo Administración y Seguridad

## CAPÍTULO 2. Propuesta de solución. Análisis y diseño

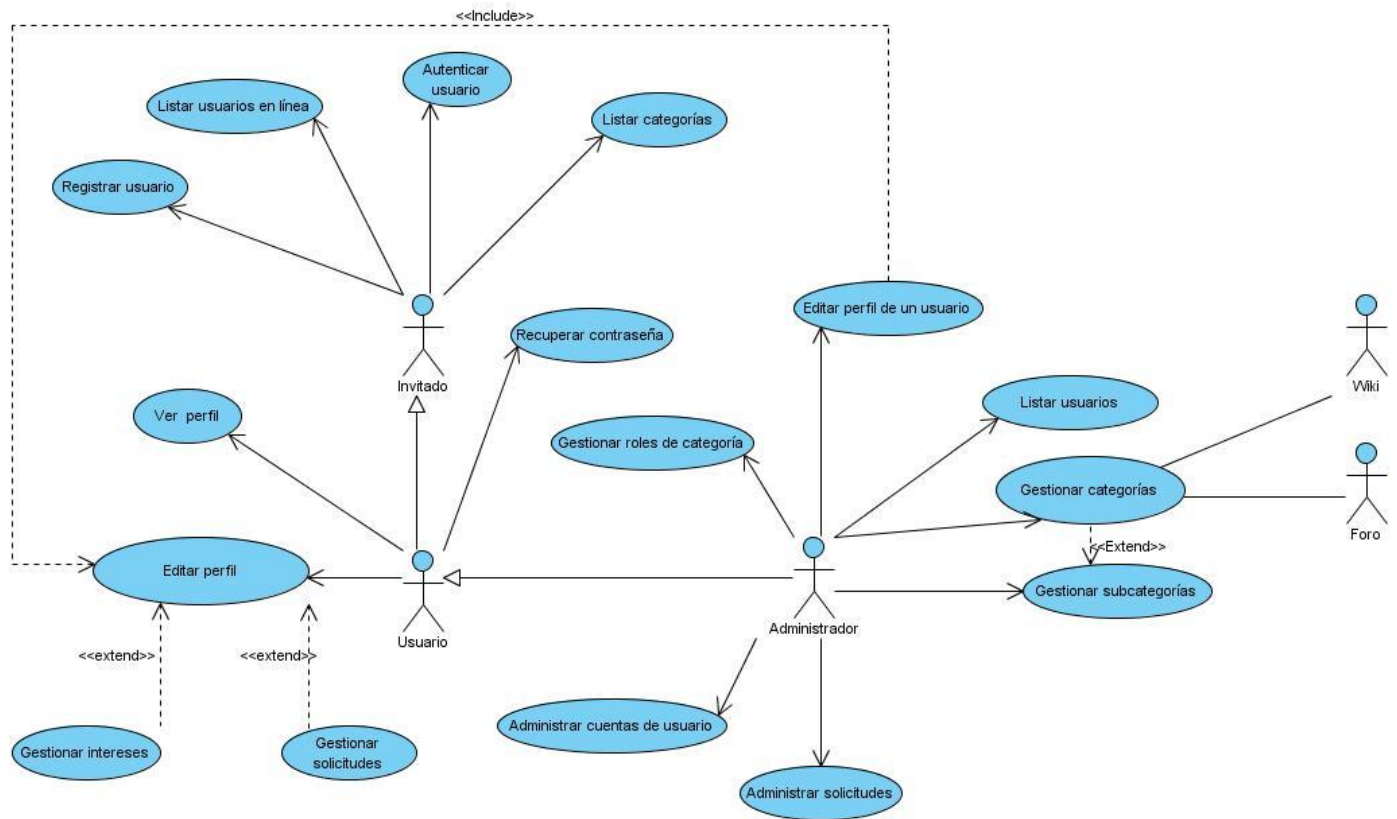


Figura 13. Diagrama de casos de uso: Módulo Gestión de Usuario y Categoría.



## CAPÍTULO 2. Propuesta de solución. Análisis y diseño

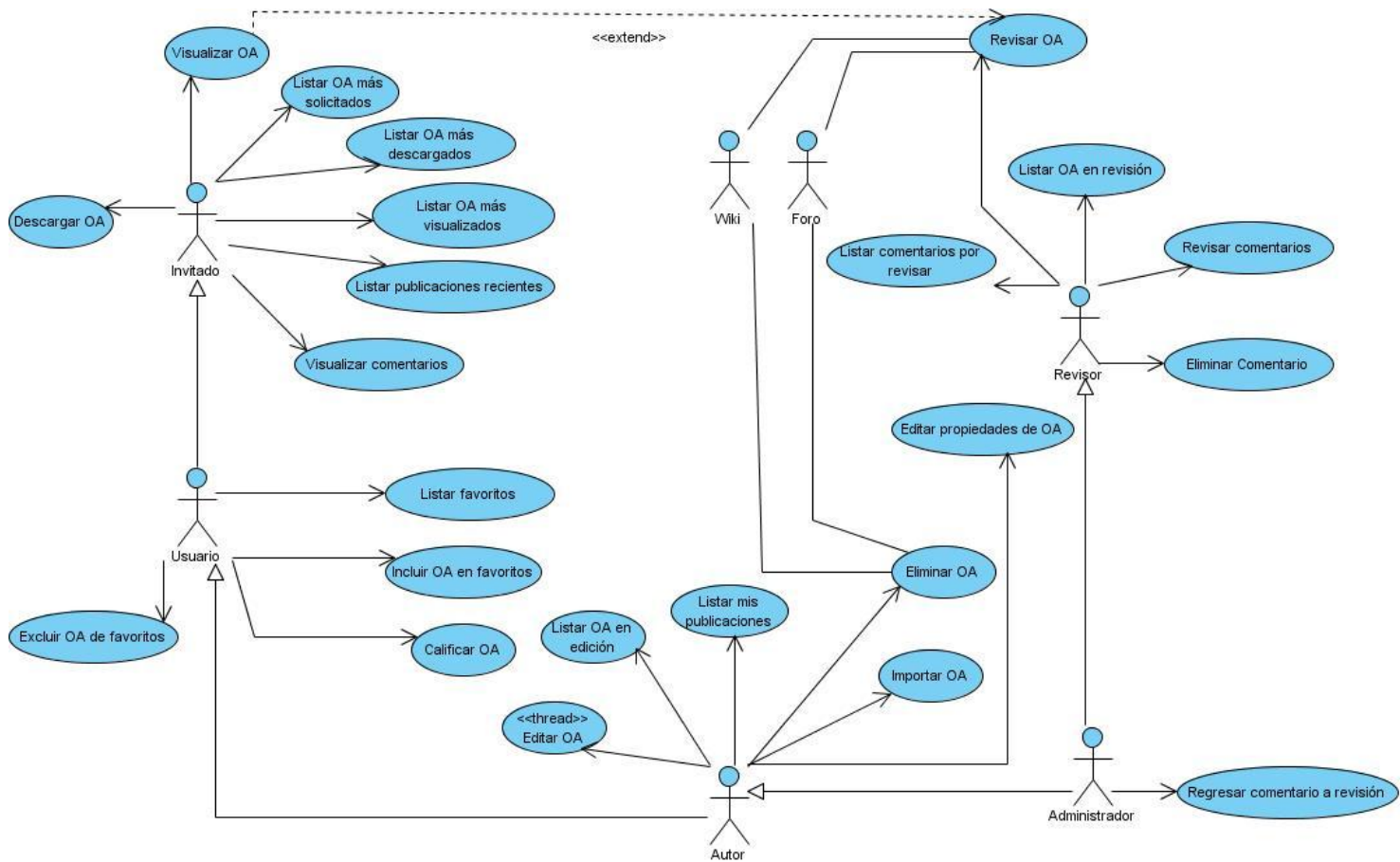


Figura 14. Diagrama de casos de uso: Módulo para la gestión de Objetos de Aprendizaje.

## CAPÍTULO 2. Propuesta de solución. Análisis y diseño

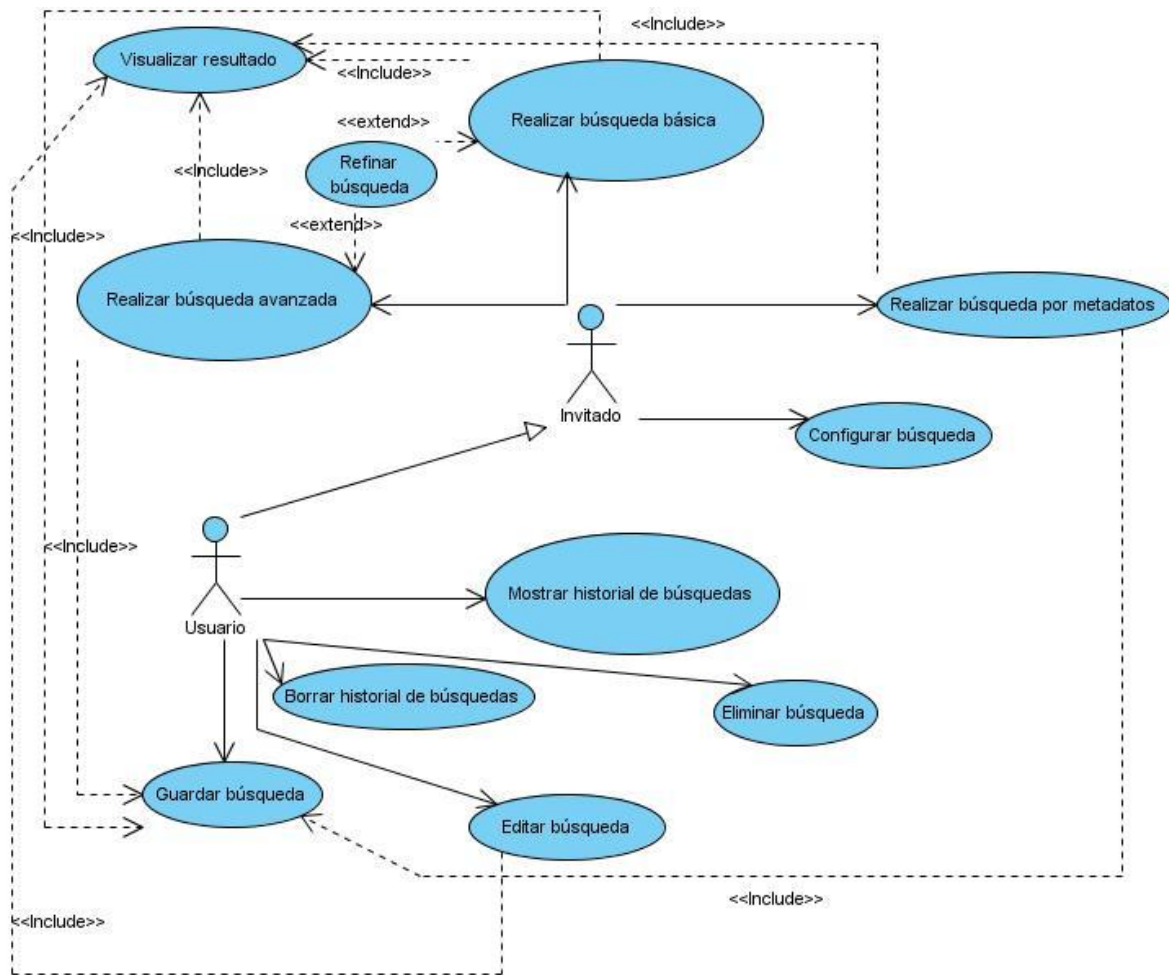


Figura 15. Diagrama de CU: Módulo Búsquedas

### 2.5 Descripción de casos de uso

Las descripciones de CU permiten comprenderlos mejor, constituyen una descripción absoluta de los requisitos que engloban (Sommerville, 2005). Permiten realizar un análisis y a partir de este poder realizar los diagramas de clases del análisis, diagramas de colaboración y estructurar los diagramas de clases del diseño. En esta sección será expuesta una muestra de las descripciones de los CU arquitectónicamente significativos, el resto de las descripciones se podrán localizar en el [Anexo II](#).

Tabla 1. Descripción del caso de uso: Gestionar categorías

<b>Objetivo</b>	Agregar, Mostrar, Modificar o Eliminar una categoría.
-----------------	---

## CAPÍTULO 2. Propuesta de solución. Análisis y diseño

<b>Actores</b>	Administrador (Inicia)	
<b>Resumen</b>	El caso de uso inicia una vez que el actor selecciona la opción: Administrar categorías. Termina una vez ha sido agregada, mostrada, modificada o eliminada una categoría.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	El actor debe haberse autenticado en el sistema.	
<b>Postcondiciones</b>	Se ha agregado, modificado, eliminado o mostrado una categoría.	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt;Nombre del flujo básico&gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona en la pantalla principal la opción: Administrar categorías.	<p>1.1 Muestra una nueva interfaz donde aparecen listadas las categorías del sistema y las subcategorías correspondientes.</p> <p>1.2 Permite que se pueda agregar una nueva categoría al sistema. <i>(Ver sección 1: Agregar categoría)</i></p> <p>1.3 Permite por cada categoría realizar las acciones:</p> <ul style="list-style-type: none"> <li>- Agregar subcategoría. <i>(Ver sección 2: CU Gestionar subcategorías)</i></li> <li>- Editar categoría <i>(Ver sección 1: Editar categoría)</i>.</li> <li>- Eliminar. <i>(Ver sección 3: Eliminar categoría)</i>.</li> <li>- Gestionar roles. <i>(Ver CU Gestionar roles de una categoría)</i></li> </ul>
2.	Decide realizar alguna de las operaciones que provee el sistema, seleccionando la opción correspondiente.	<p>2.1 Permite que sea ejecutada la acción seleccionada por el actor.</p> <p>2.2 Termina el caso de uso.</p>
<b>Flujos alternos</b>		
<b>1.3 &lt;Modificación de la categoría anónima&gt;</b>		
	<b>Actor</b>	<b>Sistema</b>

## CAPÍTULO 2. Propuesta de solución. Análisis y diseño

1.		1.1 Solamente permite para esta categoría (Anónima) que viene definida por defecto realizar la acción: Gestionar roles ( <i>Ver CU Gestionar roles de una categoría</i> )
<b>Sección 1: “Agregar categoría”</b>		
<b>Flujo básico &lt;Gestionar categorías&gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción: Agregar categoría de la interfaz Administrar categorías.	<p>1.1 Muestra una ventana donde solicita entrar el nombre de la categoría a agregar:</p> <ul style="list-style-type: none"> <li>- Nombre categoría (<i>Formato: Alfanumérico, Estado: Obligatorio</i>)</li> </ul> <p>1.2 Muestra las opciones de aceptación y cancelación de la acción: Agregar y Cancelar respectivamente.</p>
2.	Inserta en el campo Agregar categoría, el nombre que necesita añadir como categoría y selecciona la opción: Agregar, para enviar el dato al sistema.	<p>2.1 Valida el dato entrado por el usuario.</p> <p>2.2 Inserta en la lista de categorías, el dato entrado por el actor, como nombre a la categoría que se agrega.</p> <p>2.3 Actualiza la información mostrada al actor.</p> <p>2.4 Inicia un nuevo hilo de discusión en el Foro, otorgándole como nombre el mismo de la categoría agregada.</p> <p>2.5 Crea nueva entrada en la Wiki.</p> <p>2.6 Retorna al paso 1.1 del flujo básico de eventos.</p>
<b>Flujos alternos</b>		
<b>4 &lt;Cancela acción&gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción: Cancelar	<p>1.1 Termina la ejecución de la acción: Agregar categoría.</p> <p>1.2 Retorna al paso 1.1 del flujo</p>

## CAPÍTULO 2. Propuesta de solución. Análisis y diseño

		básico de eventos.
<b>4 &lt;Nombre existente&gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
2.		2.1 Muestra mensaje de error: “ <i>El nombre introducido para la nueva categoría ya existe en el sistema</i> ” 2.2 Retorna al paso 3.1 del flujo básico de eventos.
<b>Sección 2: “Editar categoría”</b>		
<b>Flujo básico &lt;Gestionar categorías&gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción: Renombrar, correspondiente a una categoría.	1.1 Muestra una ventana donde es posible modificar el nombre de la categoría seleccionada. - Nombre categoría (Formato: Alfanumérico, Estado: Obligatorio) 1.2 Muestra las opciones de aceptación y cancelación de la acción: Agregar y Cancelar respectivamente.
2.	Modifica el nombre mostrado en el campo de texto y selecciona la opción: Aceptar para terminar la acción.	2.1 Valida el dato entrado por el actor. 2.2 Modifica el nombre anterior de la categoría por el entrado. 2.3 Actualiza la información mostrada al actor y envía notificación a los usuarios cuyos objetos de aprendizaje se encontraban en esa categoría: <ul style="list-style-type: none"> <li>• Dirigido a: (Autores de los OA de la categoría)</li> <li>• Asunto: Categoría renombrada</li> <li>• Mensaje: “Estimado usuario, por propósitos administrativos la categoría (Nombre de la categoría antes de renombrar;</li> </ul>

## CAPÍTULO 2. Propuesta de solución. Análisis y diseño

		<p>a la cual pertenecen los OA: (Nombre de los OA del que es autor); ha sido renombrada, ahora se llama (Nombre de la categoría después de renombrar). Pedimos disculpas por los inconvenientes que esta decisión pueda ocasionar.”</p> <p>2.4 Actualiza en los OA de la categoría renombrada el valor del campo: Categoría en el archivo de metadatos.</p> <p>2.5 Actualiza en el foro el hilo de discusión relacionado a la categoría modificada, realizando en él, la misma modificación.</p> <p>2.6 Actualiza entrada creada en la Wiki.</p> <p>2.7 Retorna al paso 1.1 del flujo básico de eventos.</p>
<b>Flujos alternos</b>		
<b>6.1 &lt;Cancela acción&gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción: Cancelar	<p>1.1 Termina la ejecución de la acción: Agregar categoría.</p> <p>1.2 Retorna al paso 1.1 del flujo básico de eventos.</p>
<b>6.1 &lt;Nombre existente&gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
2.		<p>2.1 Muestra mensaje de error: “El nombre introducido para la nueva categoría ya existe en el sistema”</p> <p>2.2 Retorna al paso 3.1 del flujo básico de eventos.</p>
<b>Sección 3: “Eliminar categoría”</b>		
<b>Flujo básico &lt;Gestionar categorías&gt;</b>		
	<b>Actor</b>	<b>Sistema</b>

## CAPÍTULO 2. Propuesta de solución. Análisis y diseño

1.	<p>Selecciona la opción: Eliminar, correspondiente a una categoría.</p>	<p>1.1 Muestra ventana emergente para seleccionar posibles acciones a realizar con los OA de la categoría:</p> <ul style="list-style-type: none"> <li>• Desea convertir las subcategorías en categorías y mover los objetos a la subcategoría anónima.</li> <li>• Desea mover los OA de la categoría a la categoría anónima y eliminar completamente.</li> <li>• Desea eliminar completamente la categoría.</li> </ul> <p>1.2 Permite realizar las acciones: Eliminar y Cancelar de conformación y cancelación respectivamente.</p>
2.	<p>Selecciona de las opciones la necesaria y presiona el botón: Eliminar.</p>	<p>2.1 Elimina la categoría según selección realizada.</p> <p>2.2 Actualiza la información de la plataforma.</p> <p>2.3 Elimina en el foro el hilo de discusión relacionado a la categoría que se elimina.</p> <p>2.4 Elimina entrada creada con el mismo nombre en la Wiki.</p> <p>2.5 Envía notificación a los autores de los OA que se encontraban en la categoría. Envía la notificación en dependencia de la acción que se haya realizado:</p> <p><b>Caso 1:</b></p> <ul style="list-style-type: none"> <li>• Dirigido a: <i>(Autores de los OA de la categoría a eliminar)</i></li> <li>• Asunto: <i>Eliminación de categoría.</i></li> <li>• Mensaje: <i>“Estimado usuario,</i></li> </ul>

## CAPÍTULO 2. Propuesta de solución. Análisis y diseño

		<p><i>por propósitos administrativos la categoría: (categoría eliminada) fue eliminada de la plataforma. Por tales motivos ahora las subcategoría son categorías y sus objetos de aprendizaje se encuentran en la subcategoría anónima de cada una de las categorías: (Nombres de las subcategorías convertidas en categorías). Pedimos disculpas por los inconvenientes ocasionados”</i></p> <p><b>Caso 2:</b></p> <ul style="list-style-type: none"><li>• Dirigido a: <i>(Autores de los OA de la categoría a eliminar)</i></li><li>• Asunto: <i>Eliminación de categoría.</i></li><li>• Mensaje: <i>“Estimado usuario, por propósitos administrativos la categoría: (categoría eliminada) fue eliminada de la plataforma. Por tales motivos ahora las subcategoría son categorías y sus objetos de aprendizaje han sido temporalmente movidos a la categoría anónima. Pedimos disculpas por los inconvenientes ocasionados”</i></li></ul> <p><b>Caso 3:</b></p> <ul style="list-style-type: none"><li>• Dirigido a: <i>(Autores de los OA de la categoría eliminada)</i></li><li>• Asunto: <i>Eliminación de categoría.</i></li><li>• Mensaje: <i>“Estimado usuario,</i></li></ul>
--	--	---



## CAPÍTULO 2. Propuesta de solución. Análisis y diseño

		<p><i>por propósitos administrativos la categoría: (categoría eliminada) fue eliminada de la plataforma. Los OA que pertenecían a esta categoría han sido eliminados completamente del sistema. Pedimos disculpas por los inconvenientes ocasionados”</i></p> <p>2.6 Retorna al paso 1.1 del flujo básico de eventos.</p>
<b>Flujos alternos</b>		
<b>6.1 &lt;Cancela acción&gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción: Cancelar	<p>1.1 Termina la ejecución de la acción: Agregar categoría.</p> <p>1.2 Retorna al paso 1.1 del flujo básico de eventos.</p>
<b>Relaciones</b>	<b>CU Incluidos</b>	
	<b>CU Extendidos</b>	<p>Gestionar roles de categorías: Paso 1.2 del Flujo Básico</p> <p>Gestionar roles de categoría en el CU Gestionar categorías.</p> <p>Gestionar subcategorías: Paso 1.3 del Flujo Básico</p> <p>Gestionar subcategorías en el CU Gestionar categorías</p>
<b>Requisitos funcionales</b>	no	RNF 3, RNF 4, RNF 8, RNF 9, RNF 11, RNF 12, RNF 13
<b>Asuntos pendientes</b>		

## CAPÍTULO 2. Propuesta de solución. Análisis y diseño

### 2.6 Modelo de análisis

Una vez descritos los CU es necesario crear una primera representación técnica del sistema (Presman), en la cual se definen las clases que son relevantes al problema a resolver. Con la creación de un modelo de análisis se persiguen 3 objetivos fundamentales (Presman, 2005):

- Describir lo que el cliente requiere.
- Establecer una base para la creación de un diseño de software.
- Definir un conjunto de requisitos que se pueda validar una vez se construye el software.

En el modelo de análisis se produce un refinamiento de los requisitos, sin tener en cuenta el lenguaje de programación a utilizar en la construcción, la plataforma en la que se ejecutará la aplicación, los componentes prefabricados o reusables de otras aplicaciones ya que su objetivo es comprender los requisitos de software y no precisar, como se implementará la solución (Presman, 2005).

El modelo de análisis es utilizado fundamentalmente por los desarrolladores para comprender, cómo debería darse forma al sistema, es decir cómo debería ser diseñado e implementado, proporcionando una estructura centrada en el mantenimiento de aspectos tales como la flexibilidad ante los cambios y la reutilización. Esta estructura creada es la que hace posible tener una correcta entrada en las actividades de diseño e implementación.

#### 2.6.1 Diagramas de clases del análisis

Las clases del análisis son el artefacto fundamental del Modelo de análisis (Presman, 2005), la representación gráfica de cómo estas se relacionan para dar cumplimiento a un CU determinado, es de vital importancia para que el equipo de desarrollo pueda comprender con un mayor nivel de profundidad los requisitos de software. Una clase del análisis constituye una abstracción de una o varias clases o subsistemas del diseño de un sistema (Jacobson, et al., 2000).

Las clases del análisis por lo general se dividen en 3 categorías (Presman, 2005) (Figura 10):

- **Clases de interfaz:** se utilizan para modelar las interacciones entre el sistema y sus actores. Esta interacción normalmente implica recibir, presentar información y peticiones del y hacia los usuarios y sistemas externos. Modelan las partes de un sistema que dependen de sus actores, lo que

## CAPÍTULO 2. Propuesta de solución. Análisis y diseño

implica que clarifican y reúnen los requisitos en los límites del sistema. Las clases de interfaz representan abstracciones de ventanas, formularios, paneles, interfaces de comunicación y otros. Este tipo de clases deben mantenerse a un nivel bastante alto y conceptual.

- **Clases control:** representan coordinación, secuencia, transacciones, y control de otros objetos, se usan con frecuencia para encapsular el control de un CU en concreto. Representan la lógica del negocio que no puede asociarse a una información concreta, de larga duración almacenada por el sistema. No encapsulan aspectos relacionados con las interacciones con los actores.
- **Clases entidad:** se utilizan para modelar información que posee larga vida y que es a menudo persistente, modelan la información y el comportamiento asociado de algún fenómeno o concepto. Representan objetos presentes en el negocio y en el dominio del problema, suelen mostrar una estructura de datos lógica y contribuyen a comprender de que información depende el sistema.

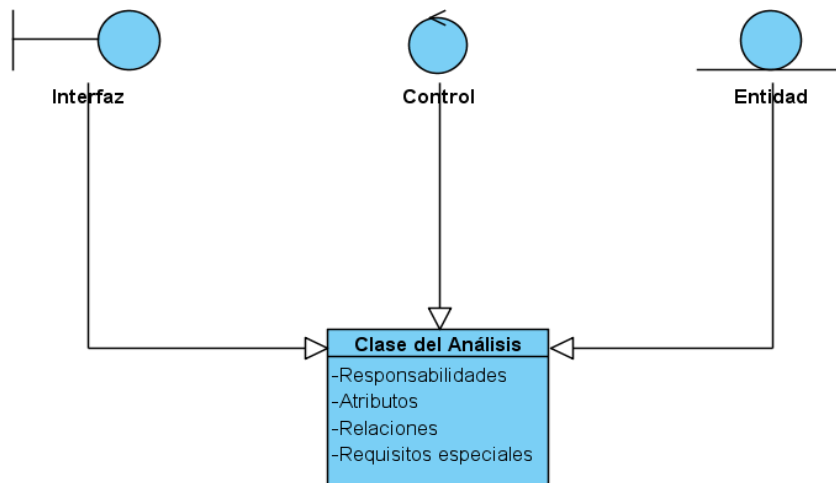


Figura 16. Atributos esenciales y estereotipos de una clase del análisis.

Cuando estas clases se relacionan, para dar a entender la realización de un CU determinado, se puede decir que se está en presencia de un Diagrama de clases del análisis (DCA). Los DCA a parte de representar la estructura de un sistema, además permiten al equipo de desarrollo comprender las relaciones que tienen lugar entre clases (Övergaard, et al., 2004).

En la presente investigación, las CA y los DCA constituyen una base sólida para entender el sistema que se modela, permitiendo además poder comprender las funcionalidades de las anteriores versiones de RHODA. A continuación se expone una muestra de los DCA creados durante el modelado de la versión

## CAPÍTULO 2. Propuesta de solución. Análisis y diseño

3.0 de RHODA, el resto de los diagramas que se corresponden con los CU arquitectónicamente significativos se pueden encontrar en el [Anexo III](#).

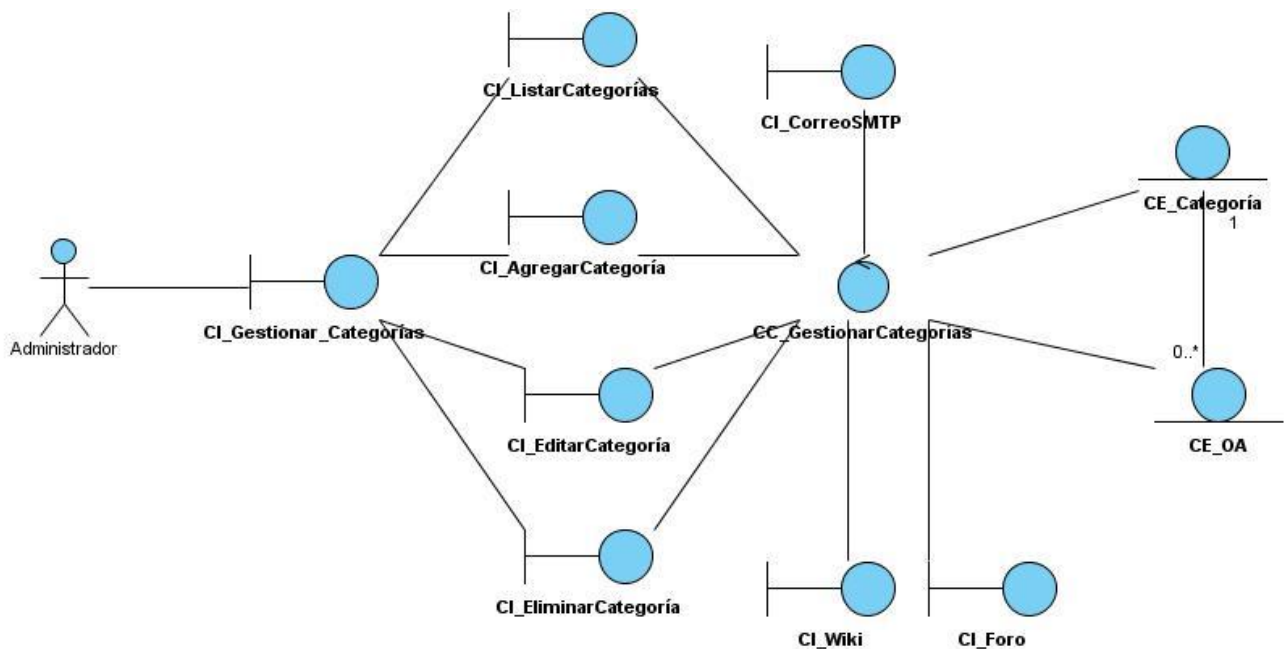


Figura 17. Diagrama de clases del análisis: CU Gestionar categorías.

### 2.6.2 Diagramas de colaboración

Al mismo tiempo que se construyen los DCA, es posible ir generando los diagramas de colaboración (DC). Un diagrama de colaboración es un diagrama de interacción que modela los objetos y enlaces implicados en la implementación de una interacción, por lo general solo se representan los objetos de CA implicados en la colaboración. Este enfatiza la organización estructural de los objetos que envían y reciben mensajes, muestra las interacciones organizadas alrededor de instancias y de los enlaces entre ellas y son muy útiles cuando se desea mostrar el diseño detallado de procedimientos (Presman, 2005).

Las características antes enunciadas, fueron las que hicieron posible que en la presente investigación se le diera un peso fundamental a este tipo de diagrama de interacción. A continuación se expone una muestra de los diagramas de colaboración creados para los casos de uso arquitectónicamente significativos, el resto de los diagramas se pueden localizar en el [Anexo IV](#).

## CAPÍTULO 2. Propuesta de solución. Análisis y diseño

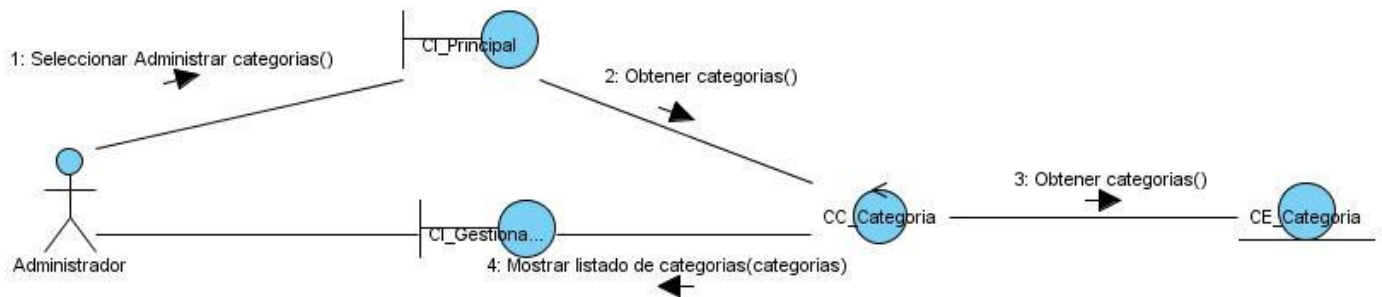


Figura 18. DC Caso de uso: Gestionar Categorías, sección Mostrar

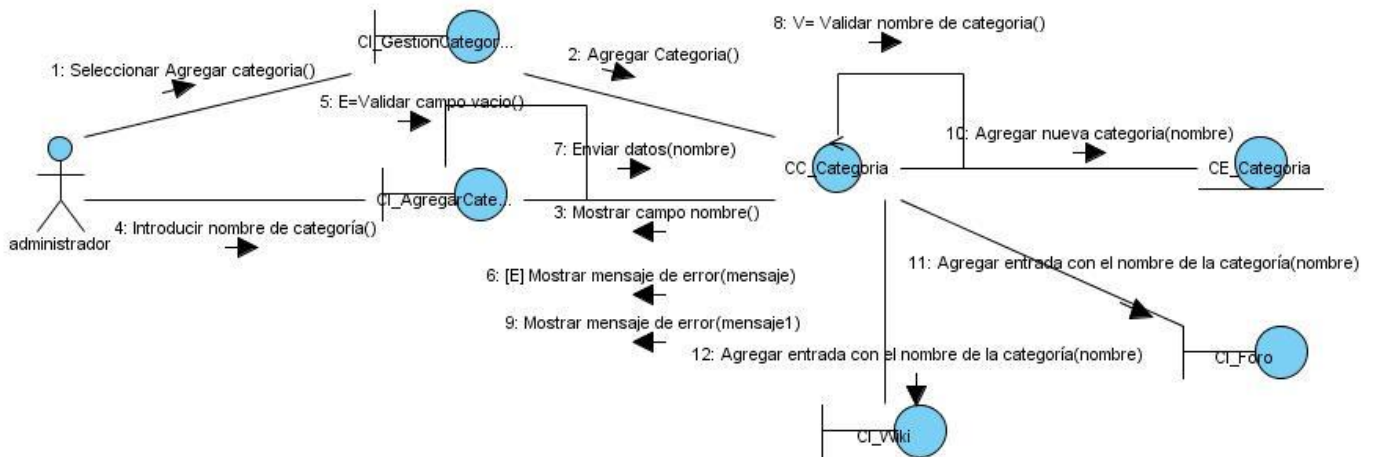


Figura 19. DC Caso de uso: Gestionar Categorías, sección Agregar

## CAPÍTULO 2. Propuesta de solución. Análisis y diseño

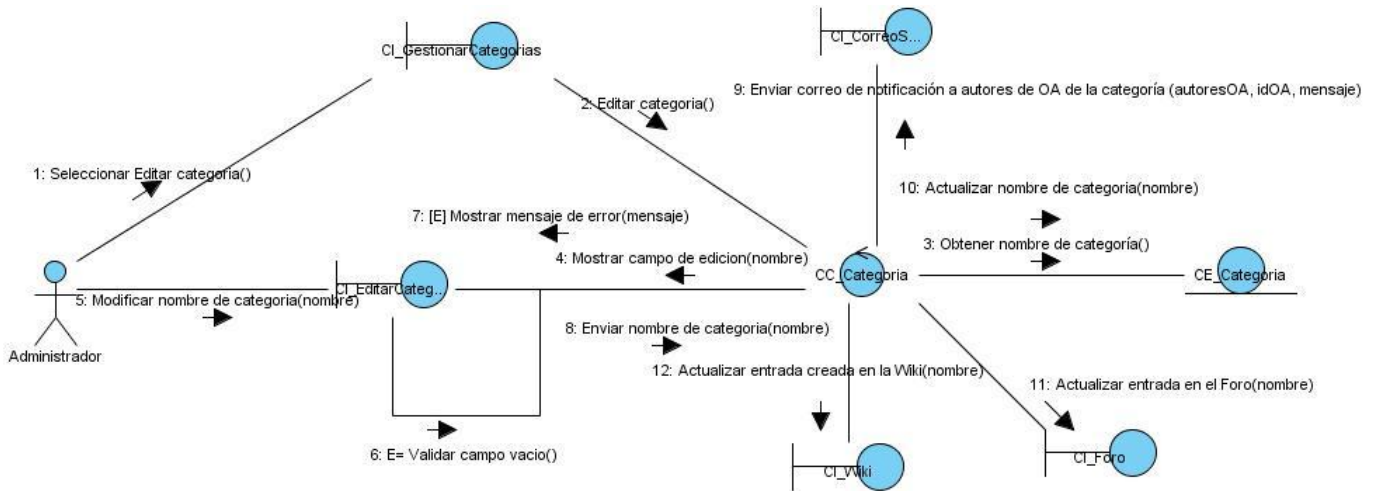


Figura 20. DC Caso de uso: Gestionar Categorías, sección Editar

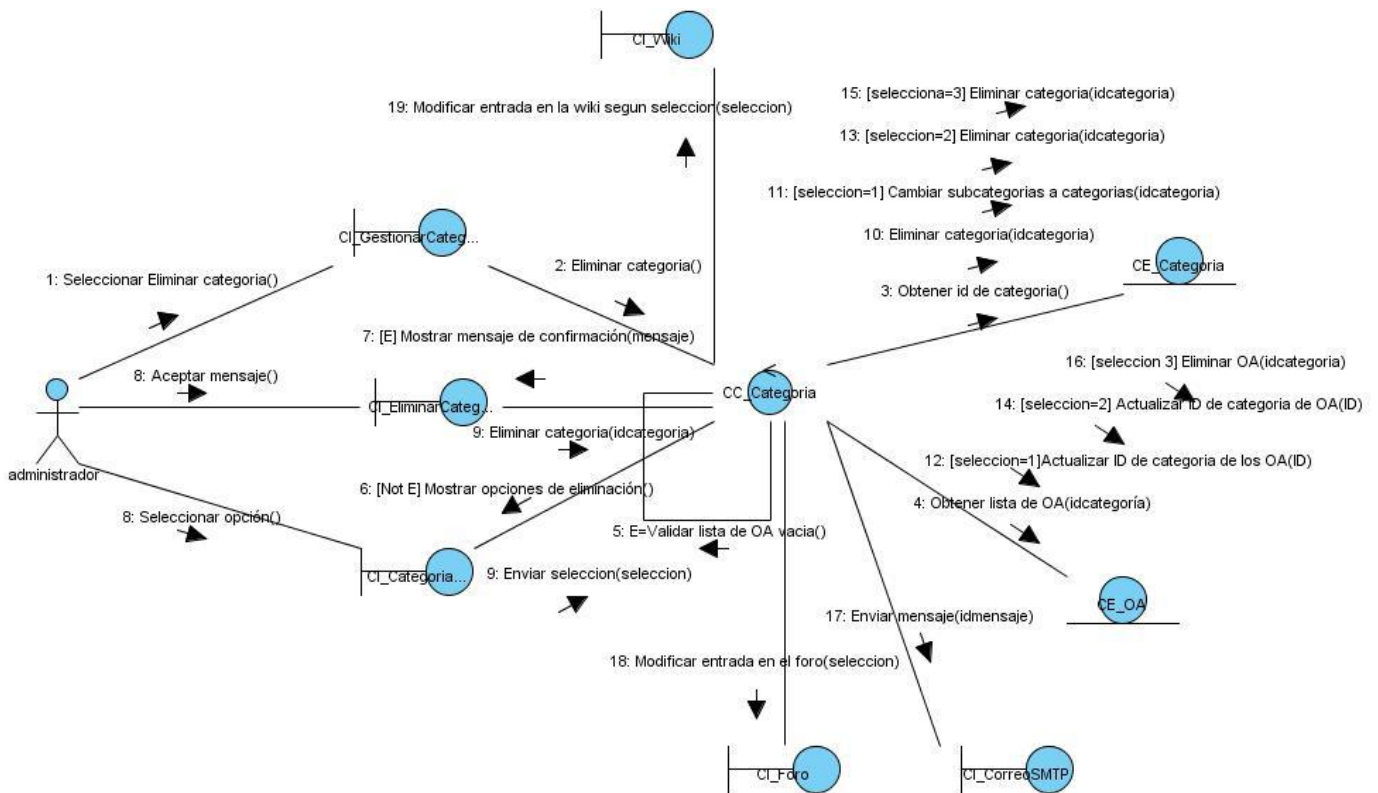


Figura 21. DC Caso de uso: Gestionar Categorías, sección Eliminar

## CAPÍTULO 2. Propuesta de solución. Análisis y diseño

### 2.7 Modelo de diseño

Es un modelo físico que constituye el plano de la implementación. Este describe la realización física de los CU, centrándose fundamentalmente como los requisitos, junto con otras reglas relacionadas con el ambiente de implementación, tienen impacto en el sistema a desarrollar, esta razón hace que el modelo de diseño sirva como una abstracción de la implementación y por consiguiente la entrada principal a las actividades de esta.

“Una clase del diseño es una abstracción sin costuras de una clase o construcción similar” (Jacobson, et al., 2000). Las clases del diseño están caracterizadas por (Jacobson, et al., 2000):

- El lenguaje utilizado para especificar una clase del diseño es el mismo que el lenguaje de programación. Consecuentemente las operaciones, parámetros, atributos, tipos y demás son especificados utilizando la sintaxis del lenguaje de programación elegido.
- Pueden realizar y proporcionar interfaces si tiene sentido hacerlo en el lenguaje de programación que se utiliza.

#### Patrones de Diseño utilizados

Los patrones de diseño son soluciones a problemas específicos y comunes del diseño orientado a objetos, constituyen salidas basadas en la experiencia que se ha demostrado que funcionan. Estos no son fáciles de entender, pero una vez entendidos se obtienen diseños más flexibles, modulares y reutilizables (Larman, 2004). Un patrón de diseño no debe entenderse como el resultado al que se quiere llegar, sino como el inicio. Por tal motivo en la presente investigación una vez comenzado el flujo de trabajo de diseño se pensó en la utilización de estos patrones, teniendo como antesala, el entendimiento de los requerimientos y el modelo de análisis.

Entre los patrones de diseño utilizados se puede hacer alusión a:

- Patrones de asignación de responsabilidades (Grasp): Alta cohesión, Bajo acoplamiento, Experto, Controlador (Schmidt, et al., 2000).
- Patrones estructurales: Facade y Decorator (Larman, 2004).
- Patrones de comportamiento: Cadena de responsabilidades, Memento, Observer (Larman, 2004).

## CAPÍTULO 2. Propuesta de solución. Análisis y diseño

### 2.7.1 Diagramas de clases del diseño (DCD).

Los DCD están constituidos por un conjunto de clases del diseño, que se relacionan e interactúan para lograr la realización física de un requisito. Por lo general un DCD se corresponde con la realización de un CU, constituyendo un mapa de la implementación del mismo, mostrando de esta forma las clases participantes, subsistemas y sus relaciones.

En la presente investigación se hizo necesario considerar 2 aspectos, el primero está relacionado al tipo de aplicación que se construye (aplicación o plataforma web) y el segundo el *framework* de desarrollo que se utiliza.

Respecto al primero de los aspectos se utilizó los estereotipos destinados al modelado de clases del diseño. En el segundo, se utilizó el framework de desarrollo Symfony, lo que trae como consecuencia que para diseñar obligatoriamente se tiene en cuenta el patrón arquitectónico que implementa (MVC). Este patrón separa la lógica de presentación, lógica de negocio y el modelo de datos.

A continuación se expone una muestra de cómo son los DCD de la versión de RHODA que se modela, en ellos ha sido necesario ocultar las funciones y atributos de las clases, el resto de los diagramas correspondientes con los CU arquitectónicamente significativos se encuentran en el [Anexo V](#).



## CAPÍTULO 2. Propuesta de solución. Análisis y diseño

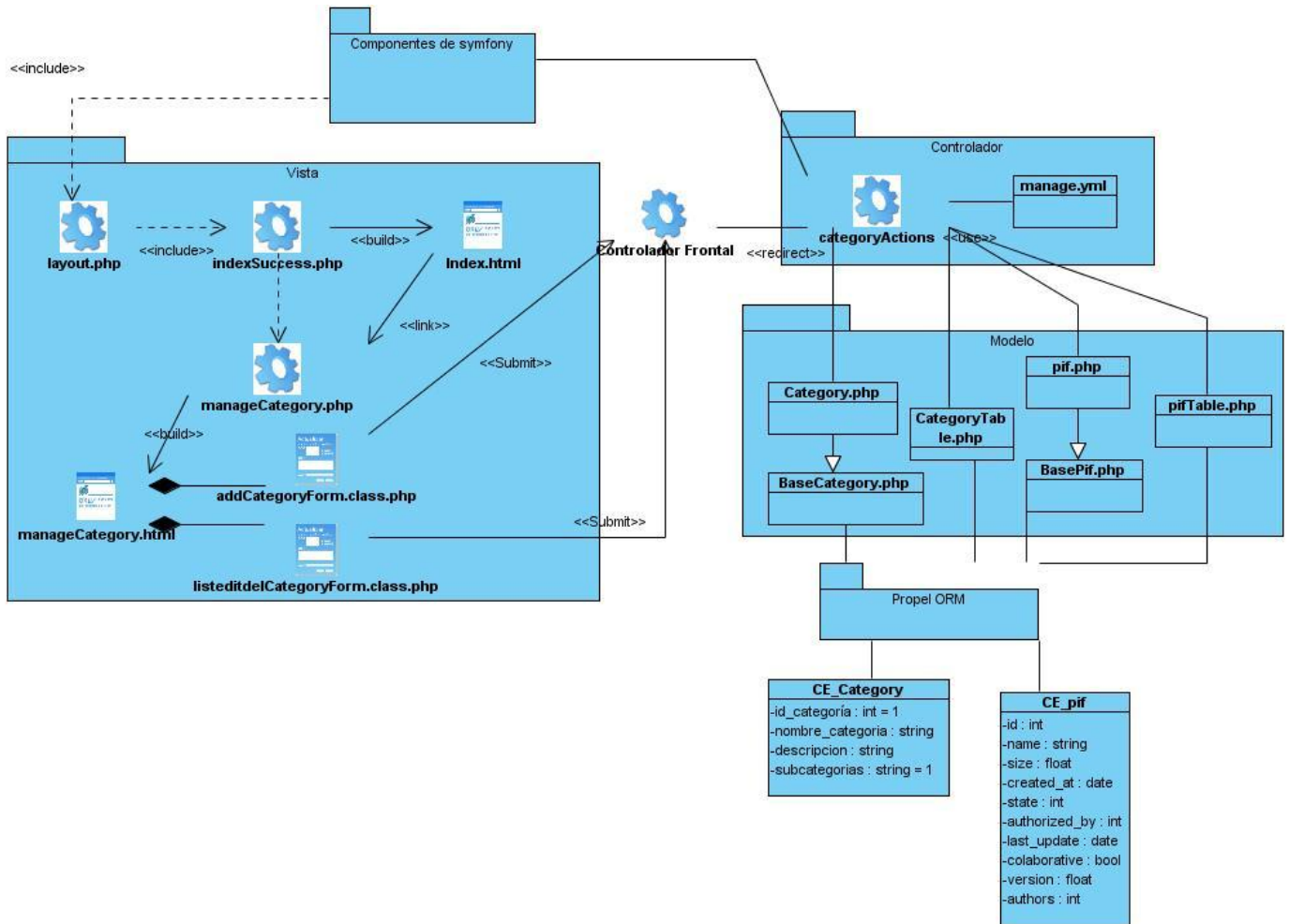


Figura 22. DCD CU Gestionar categorías.

# CONCLUSIONES

## Conclusiones

Con la realización de la presente investigación se han dado cumplimiento a los objetivos propuestos, obteniéndose los siguientes resultados:

- Los ROA estudiados no cuentan con procesos de búsqueda en contenidos para la recuperación de la información, pero este es un mecanismo necesario en este tipo de sistema para mejorar eficacia en los resultados.
- Se definieron, a través del estudio del estado del arte, las funcionalidades a incorporar y perfeccionar en la versión 3.0 de RHODA.
- Se obtuvo una nueva estructuración jerárquica de los roles del repositorio, lo que contribuye al proceso de gestión de OA en el sistema.
- Se delimitó el contexto de utilización de las herramientas de la web 2.0: Wiki, Foro y Blog, de forma tal que estas apoyen exclusivamente la gestión de OA.

## RECOMENDACIONES

### Recomendaciones

Realizar estudios periódicos de los avances tecnológicos encaminados al desarrollo y gestión de los procesos que ocurren en los repositorios de objetos de aprendizaje, con el objetivo de poder realizar propuestas de actualización del producto.

Realizar estudio de factibilidad respecto a la utilización de CMS en el desarrollo de los ROA, a fin proveer un perfeccionamiento en la gestión de información.

Continuar rigiendo el proceso de desarrollo del producto, a través del uso de las pautas definidas por CMMI, para garantizar un software con la calidad requerida.

Sugerir al equipo de desarrollo, prestar especial atención a garantizar los aspectos de seguridad, definidos durante la concepción, análisis y diseño del producto.

# REFERENCIAS BIBLIOGRÁFICAS

## Referencias Bibliográficas

**ADL. 2002.** *Emerging and Enabling Technologies for the design of Learning Object. Advanced Distributed Learning Initiative.* 2002. [Consultado el: 16 de diciembre de 2010]. Disponible en: <http://xml.coverpages.org/ADLRepositoryTIR.pdf>. .

**Azcárate, Ernesto Quiñones. 2008.** *Sistemas Administradores de Bases de Datos Libres para el Entorno Empresarial.* Lima, Perú : s.n., 2008. [Citado el: 13 de enero de 2011]. Disponible en: <http://www.eqsoft.net>.

**Bermon, Angarita Leonardo. 2010.** *Librería de Activos para la gestión del conocimiento sobre procesos: Software Pal-Wiki. Tesis doctoral.* Madrid : s.n., 2010. Tesis doctoral.

**Borgman, Christine L. 1999.** *What are digital libraries, who is building them, and why?. Digital Libraries: Interdisciplinary Concepts, Challenges, and Opportunities. Proceedings of the Third International Conference on Conceptions of Library and Information Science.* Dubrovnik, Croatia : Zagreb: Lokve, 1999. [Citado el: 18 de diciembre de 2010]..

**Chiarani, Marcela Cristina, Pianucci, Irma Guadalupe y Leguizamón, Guillermo. 2006.** *Repositorio de Objetos de Aprendizaje: propuesta de diseño.* San Luis, Argentina : s.n., 2006.

**Chowdhury and Chowdhury, Sudatta. 2002.** *Introducción to Digital Libraries.* London : Neal-Schuman Publishers, 2002. 978-1-85604-465-3.

**Coad, Peter and Jill, Nicola. 1993.** *Object Oriented Programming.* New York : Yourdon Press, 1993. 0-13-032b1b-X.

**Cockburn, Alastair. 2000.** *Writing effective use cases.* s.l. : Addison - Wesley , 2000. 978-0201702255.

**CyberFicient Technologies Inc. 2001.** *Modell-View-Controller, Architecture Overview.* USA : s.n., 2001. [Citado el: 17 de enero de 2011].

**Downes, Stephen. 2004.** *The Learning Marketplace. Meaning, Metadata and Content Syndication in the Learning Object Economy.* Moncton, New Brunswick : s.n., 2004.

**Fernandez-Majón, Baltasar. 2006.** *Especificaciones y estándares en e-learning.* Madrid, España : s.n., 2006. [Citado el: 18 de diciembre de 2010]. Disponible en: [http://reddigital.cnice.mec.es/6/Articulos/Articulo\\_resumen.php](http://reddigital.cnice.mec.es/6/Articulos/Articulo_resumen.php).

**Fowler, Martin and Scott, Kendal. 1996.** *UML Gota a Gota.* s.l. : Prentice Hall, 1996. 9684443641.

**Gaona, Amparo López. 2008.** *XML y Bases de Datos XML.* 2008. [Consultado el: 11 de enero de 2011].

## REFERENCIAS BIBLIOGRÁFICAS

**Geser, Guntram. 2007.** Prácticas y recursos de educación abierta: la hoja de ruta OLCOS 2012. 2007. Vol. 4, 1. [Consultado el:9 de enero de 2011]. Disponible en: <http://www.uoc.edu/rusc/4/1/dt/esp/geser.pdf>. 1698-580x.

**González, Roxana Cañizares. 2008.** Framework para la gestión de contenidos educativos. *Trabajo de Diploma en opción al título Ingeniero en Ciencias Informáticas*. La Habana : s.n., 2008.

**Grady, Robert Boch. 1992.** *Practical Software Metrics for Project Management and Process Improvement*. s.l. : Prentice Hall, 1992. 0138218447.

**Graeme, Daniel. 2004.** Learning Objects Repositories. *WWW tools for Education*. 2004.

**Griff, Richards, et al. 2002.** *Repository, The Evolution of Learning Object Repository Technologies: Portals for On-line Objects for Learning*. Canadá : s.n., 2002. Vol. 17.

**Hilera González, José Ramón and Hoya Marín, Rubén. 2010.** ESTÁNDARES DE E-LEARNING: Guía de consulta. Alcalá, España : s.n., 2010. 978-84-693-0263-7 .

**Hodgins, Wayne. 2001.** IEEE LTSC Learning Technology Standards Committee P1484. *ADLNET*. USA : s.n., 2001.

**Huaroto, Libio. 2010.** Repositorio Digital: Estrategias para su implementación y uso. Perú : s.n., 2010. <http://www.altamira-peru.org/jornadas/ponencias/repositorio-digital.pdf>.

**IEEE LTSC. 2001.** Learning Technologies Standard Comminttee. [En línea] 2001. [Citado el: 14 de 12 de 2010.] <http://ltsc.ieee.org/wg12/index.html>.

**Iglesias, Borja Manero. 2003.** Estudio de la propuesta IMS de estandarización de enseñanza asistida por. 2003. [Consultado el:11 de enero de 2011].Disponible en <http://www.fdi.ucm.es/profesor/borja/Informe%20tecnico.pdf>.

**IMS GLC. 2004.** IMS Global Consortium. [En línea] 2004. [Citado el: 15 de 12 de 2010.] <http://www.imslobal.org/digitalrepositories/index.html>.

**Jacobson, Ivar, Booch, Grady and Rumbaugh, Jam. 2000.** *El Proceso Unificado de Desarrollo de Software*. Madrid, España : Pearson Education, S.A, 2000. [Consultado el: 11 de enero de 2011]. 84-7829-036-2.

**JORUM TEAM PROJECT . 2005.** *Report on Open Source Learning Object Repository Systems*. Reino Unido : s.n., 2005.

**Larman, Craig. 2004.** *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*. 3ra Edición. s.l. : Prentice Hall, 2004. 978-0131489066.

## REFERENCIAS BIBLIOGRÁFICAS

—. 2003. *UML y Patrones*. 2003. [Consultado el: 20 de diciembre de 2010].

**Leyva Leyva, David and Tamayo Ávila, Daymy. 2008.** REPOSITORIO SEMÁNTICO DE OBJETOS DE APRENDIZAJE PARA LA REUTILIZACIÓN DE CONTENIDOS EDUCATIVOS. *VI Congreso Internacional Pedagogía*. La Habana : s.n., 2008. [Citado el:11 de diciembre de 2010]. Disponible en:[http://e-spacio.uned.es/fez/eserv.php?pid=bibliuned:19955&dsID=Repositorio\\_sem\\_\\_ntico\\_\\_de\\_objetos\\_de\\_aprendizaje.pdf](http://e-spacio.uned.es/fez/eserv.php?pid=bibliuned:19955&dsID=Repositorio_sem__ntico__de_objetos_de_aprendizaje.pdf).

**López, Guzmán Clara and García, Peñalvo Francisco J. 2006.** Repositorios de Objetos de Aprendizaje: Bibliotecas para compartir y reutilizar recursos en los entornos e-learning. Distrito Federal, Mexico : s.n., 2006. Vol. 09, 002. 0187-750X.

**López, Guzmán Clara. 2005.** Los Repositorios de Objetos de Aprendizaje como soporta un entorno e-learning. *Tesis Doctoral*. Salamanca, España : s.n., 2005. [Citado el: 11 de noviembre de 2010]. Disponible en:[http://gredos.usal.es/jspui/bitstream/10366/56649/1/DIA\\_Repositoriosobjetos.pdf](http://gredos.usal.es/jspui/bitstream/10366/56649/1/DIA_Repositoriosobjetos.pdf).

**Mason, Robin and Rehak, Daniel R. 2004.** Keeping the Learning in Learning Objects. *Reusing Online Resources: A Sustainable Approach to E-learning* . Sterling, London : VA Taylor & Francis, 2004.

**Mena, Marta. 2004.** *La educación a distancia en América Latina. Modelos Tecnologías y realidades*. Buenos Aires : s.n., 2004.

**Milligan, Colin. 2007.** Community Dimensions of Learning Object Repositories. *JISC DEVELOPMENT PROGRAMMES*. Glasgow, Reino Unido : s.n., 2007. [http://www.academy.gcal.ac.uk/cdlor/documents/CDLOR\\_Final\\_Recommendations\\_v1p0\\_000.pdf](http://www.academy.gcal.ac.uk/cdlor/documents/CDLOR_Final_Recommendations_v1p0_000.pdf).

**Nguen, Van Nhu. 2007.** *Binding the Simple Query Interface*. 2007.

**Ovalles, Pavó Freddy Oswaldo. 2007.** Herramientas colaborativas: Foros, Wikis, Blog. s.l. : SENA NDS, 2007.

**Övergaard, Omar and Palmkvist, Karin. 2004.** *Uses Cases Patterns and Blueprints*. s.l. : Addison Wesley , 2004. 0-13-145134-0.

**Palomar, David. 2008.** Sistema de Control de Versiones Distribuido. [Online] mayo 17, 2008. [Cited: Diciembre 19, 2010.] <http://adamaconsulting.blogspot.com/2008/05/sistemas-de-control-de-versiones.html>.

**PHP. 2010.** PHP. [Online] 2010. [Cited: diciembre 17, 2010.] [www.php.net](http://www.php.net).

**PostgreSQL Global Development Group. 2010.** *PostgreSQL 9.0 Reference Manual*. s.l. : Network Theory Ltd, 2010. 978-1-906966-04-1.

## REFERENCIAS BIBLIOGRÁFICAS

- Potencier, Fabien and Zaninotto, Francois. 2005.** *Symfony: La guía definitiva.* 2005. [Consultado el: 24 de diciembre de 2010].
- Presman, Roger. 2005.** *Ingeniería de Software: Un Enfoque Práctico.* Sexta Edición. s.l. : McGraw-Hill, 2005. 9701054733.
- Pujol, Dolores. 2005.** *Contenidos Digitales y Formación on-Line. eLearning Workshops.* [Online] 2005. [Cited: 12 15, 2010.] <http://www.elearningworkshops.com/modules.php?name=News&file=article&sid=456>.
- RAE. 2010.** Real Academia Española. [En línea] 2010. [Citado el: 19 de 12 de 2010.] [www.rae.es](http://www.rae.es).
- Red TTnet España. 2006.** *La Formación Sin Distancia.* Madrid, España : s.n., 2006.
- Ros, Miguel Zapata. 2009.** SECUENCIACIÓN DE CONTENIDOS. ESPECIFICACIONES PARA LA SECUENCIACION INSTRUCCIONAL DE OBJETOS DE APRENDIZAJE. *Tesis Doctoral.* Alcalá de Henares, España : s.n., Diciembre 2009.
- Rosenberg, Mark Jeffrey. 2001.** *E-Learning Strategies for Delivering Knowledge in the Digital Age.* United States : Mac - Hill Companies, 2001. 0-07-136268-1.
- Salvador, Broche Orlando Felipe and Soler, Martín Javier. 2010.** *Implementación de un Repositorio de Objetos de Aprendizaje para la Universidad de las Ciencias Informáticas.* Ciudad de La Habana : s.n., 2010.
- Schmidt, Douglas, et al. 2000.** *Pattern-Oriented Software Architecture, Patterns for Concurrent and Networked Objects.* London : John Wiley & Sons, Inc., 2000. [Citado el: 16 de enero de 2011]. 0-471-60695-2.
- SEI . 2010.** CMMI. *Capability Maturity Model Integration.* [Online] SEI, 2010. [Cited: enero 2011, 20.] <http://www.sei.cmu.edu/cmmi/>.
- SENSIOLABS. 2010.** *Symfony. More with Symfony 1.4.* Enero 25, 2010. [Consultado en: 17 de enero de 2011].
- Sommerville, Ian. 2005.** *Ingeniería del Software.* s.l. : Addison Weasley, 2005. 84-7829-074-5.
- Wiley, David A. 2000.** Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. *The Instructional Use of Learning Objects.* Utah : University of Utah, 2000, Vol. 2830, pp. 1-35. Consultado 10 de enero 2011 en <http://reusability.org/read/chapters/wiley.doc>.
- Wordpress. 2010.** RSS Explicado. *RSS Explicado.* [Online] Wordpress, 2010. [Cited: 12 10, 2010.] [www.rss.nom.es](http://www.rss.nom.es).

## Glosario de Términos

### Glosario de Términos

**BSD:** Distribución de Software de Berkeley, es una licencia otorgada principalmente para los sistemas creados por Berkeley Software Distribution, es una licencia de software libre permisiva. Tiene menos restricciones en comparación con otras como la GPL ya que permite el uso del código en aplicaciones no libres.

**e-learning:** grupo o conjunto de tecnologías, aplicaciones y servicios que apoyan los procesos de enseñanza – aprendizaje. Su aplicación está ligada a los modelos formativos, sobre todo a los que son semipresenciales o no presenciales. El término está fuertemente asociado al uso de las TIC y su aplicación a la educación.

**Entorno e-learning:** conjunto de herramientas y tecnologías que interactúan en un espacio virtual determinado, con el fin de mejorar los procesos de enseñanza – aprendizaje. El más simple de los entornos e-learning debe estar formado al menos por: un LMS, una HA y un ROA.

**HA:** software que permite a un autor generar un programa instruccional sin elementos de programación explícitos, simplemente especificando el contenido instruccional y la lógica del aprendizaje. Estas combinan los componentes de guiones en una lección estructurada con interacciones definidas para mostrar a los estudiantes a través de una computadora. Además proporcionan un marco de lección actual con una estrategia de aprendizaje implícita o explícita.

**Interés:** término utilizado en el repositorio para nombrar áreas o categorías de preferencia del usuario, una vez creado el interés el usuario puede solicitar recibir las notificaciones respectivas al mismo.

**LMS:** por lo general son plataformas web utilizadas en la educación como medio de apoyo a los procesos de enseñanza – aprendizaje, sus funciones están encaminadas a la gestión y registro de alumnos y contenidos formativos, control de acceso y seguimiento del progreso de los alumnos, administración y programación de cursos y gestión de informes.

**SGBD:** sistema gestor de base de datos, permite la gestión y mantenimiento de las bases de datos.

**Solicitud:** peticiones que realizan los actores del sistema al administrador con el fin de obtener algún beneficio.