

2011



*Universidad de las  
Ciencias Informáticas  
Facultad 4*

*Secuencia y navegación en los Objetos de Aprendizaje  
de la herramienta de autor CRODA.*

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

*Autores:*

*Gretell Lili Companioni Castillo*

*Daril Florat Reyes*

*Tutores:*

*Ing. Jorge Iturria Pozo*

*Ing. Osvaldo E Stable Vilches*

*Ciudad de la Habana, Junio 2011*

*“Año 53 de la Revolución”*

## *Pensamiento*

*' '... Batalla de ideas no significa solo principios, teorías, conocimientos, cultura, argumentos, réplica y contrarréplica, destruir mentiras y sembrar verdades. Significa hechos y realizaciones concretas ' '...*

*Fidel Castro.*

## *Declaración de autoría*

Por este medio declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) para que hagan el uso que estimen pertinente con él. Para que así conste firmamos la presente a los \_\_ días del mes de \_\_ de 2011.

### Autores

---

Gretell Lili Companioni Castillo

---

Daril Florat Reyes

### Tutores

---

Ing. Osvaldo Ernesto Stable Vilches

---

Ing. Jorge Iturria Pozo

## *Agradecimientos*

### *De Lili*

Todo el esfuerzo por lograr la meta de ser un profesional culmina en un día como este, donde nos queda la satisfacción y el orgullo de haber compartido con personas maravillosas que siempre quedarán en nuestros corazones. A esas personas quisiera agradecerles.

A mi grupo de primer año, “el mejor de todos”, donde tengo grandes amistades que han sabido compartir buenos y malos momentos. A mis amigas Dianamary, Annia, Evelin, Milena, Liset, Lisandra y La Suli. A mis amigos Eduar, Pedro Sáez, Pedro Gonzáles, Jose, Rubén, Rene, Santiago y Yasmany. A mi amiga Susana que comparte conmigo en los estudios desde hace más de 10 años, por su cariño y amistad incondicional. A mis profesores por enseñarme el camino correcto por el que transcurrir, en especial a mis tutores Héctor, Dunia, Jorge y Osvaldo, por mantener siempre la disposición de ayudarnos en lo que fuera necesario.

Agradezco a todas mis tías, en especial a tía Tabe por mantenerse en contacto siempre al tanto de mis estudios y a mis tíos Carmita y Luis por acogerme en mis años de universidad y darme la posibilidad de sentirme como en mi casa.

A mi otra familia de Nuevitas, que la quiero mucho. A Mery, Angelito, Tere, Machete, Susel, Zory, Angel, Marbelis y otros. Gracias por ser como son y dejarme formar parte de su familia. Quisiera detenerme en una persona maravillosa a la que quiero agradecer de una forma especial, en mi primer año conocí a mi novio y compañero de tesis Daril. En él encontré un amigo y compañero que hizo que mis días de universidad se convirtieran en los mejores de mi vida. Hay personas que pasan por la vida dejando huellas imborrables en la vida de otras, tú eres una de esas personas, gracias por haber compartido junto a mí estos 5 años. Te quiero mucho.

Por último quiero agradecerles a mi papá Onildo y a mi hermano Willman, por brindarme el apoyo que necesitaba para seguir mis estudios y finalizar mi carrera. Y en especial a mi mamá Cacha por darme sus consejos, preocuparse tanto por mí y por ser la persona por la cual me esfuerzo. Espero que la vida me dé la oportunidad de retribuirles todo lo que han hecho por mí.

¡A todos Gracias!

## *De Daril*

Cuando era pequeño escuché la frase: “No es difícil alcanzar la cima de la montaña, la dificultad radica en subir la escarpada”. Realmente subir la escarpada, para alcanzar la cima de mis estudios, hubiera sido imposible si determinadas personas no se hubieran involucrado en este empeño. A todas ellas les expreso mis más sinceros agradecimientos. Quisiera comenzar agradeciendo a Dios por guiar mis pasos en los estudios y ayudarme a que este sueño se hiciera realidad para el orgullo de mis padres y familia.

Desde lo más profundo de mí ser quiero agradecerle a mi novia y compañera de tesis Lilí. Agradecerle por tanta dedicación y comprensión, por soportarme con mis defectos y virtudes. Por estar siempre a mi lado, incluso en los momentos más difíciles de mi vida. Por ser mi confidente, mi amiga y por corresponderle en todo momento a este amor sin igual.

Agradecerle a mi familia por su preocupación en cada capítulo de mi vida. Especialmente a mis tíos y primos. Agradecer también a Edel, la Coja, Ángela y familias. Merecido agradecimiento a la familia de Lilí por la confianza y por el apoyo en estos años de carrera. Siempre formarán parte de mis pensamientos. Agradecer a mis amigos-hermanos Norlen, José Alberto, Ezequiel y José Ramón. Agradecer a los amigos y compañeros que encontré en mi paso por la universidad, por tantos momentos compartidos agradecer a Eduar, el Ruben, Jose, Roger, el Rene, los Pedros (González y Sáez), Carlos, Santiago, Yasmani y a las muchachitas Evelyn, Annia, Dianamarys, Liset, Susana, Milena, Suly y Lisandra La Hoz.

Agradecer fundamentalmente a todos los profesores en mi vida de estudiante y a los tutores Héctor, Jorge, Dunia y Osvaldo.

Agradecer a mis abuelos Margarita y Evelio. A mi abuelita Teresita por el cariño en estos 23 años. Agradecimiento especial para mi abuelo Ángel Enrique (Machete) por ser en gran parte de mi vida un guía y sobre todo un amigo.

Mi mayor agradecimiento a, quienes me enseñaron aquella frase cuando era niño, mis padres Ángel y Mérida por darme la oportunidad de nacer y crecer junto a ellos. Por guiar mis estudios. Por los sacrificios que han hecho por mí en tantos años. Por cada momento de felicidad y tristeza compartido, por la unidad y confianza que hemos mantenido y sobre todas las cosas quisiera agradecerles por el inmenso amor que me han ofrecido, siendo la principal fuente de inspiración en este empeño.

¡A todos eternamente agradecido!

## *Dedicatoria*

### *De Lili*

Dedico los logros obtenidos a lo largo de mi vida como estudiante a mi familia que me ha apoyado en estos largos años de educación, a mi papá, mi hermano y en especial a mi mamá por ser mi guía y enseñarme que en la vida todo requiere de esfuerzo.

A todos los profesores que de una forma u otra ayudaron en mi superación profesional.

### *De Daril*

Con gran nostalgia, pero a la vez satisfacción quiero dedicarle este trabajo a la memoria de mi abuelo Machete, pues la vida no permitió que celebráramos este momento juntos, sin embargo estoy convencido que donde quiera que se encuentre siente un inmenso regocijo por el resultado obtenido.

Dedicar también este trabajo de diploma a mis padres, como reconocimiento a su consagración y esmero en mis años de estudiante. Para ustedes este poquito de gratitud que hoy no constituye un sueño, es una realidad que, con gran emoción, les hago entrega.

## *Resumen*

El uso de las Tecnologías de la Información y las Comunicaciones (TIC) en el proceso docente educativo es cada vez mayor. Para ahorrar tiempo y esfuerzo de los profesores, fueron creadas las herramientas de autor, las cuales representan un punto clave en el desarrollo de la comunidad del e-learning, pues facilitan la creación de recursos educativos para el proceso de aprendizaje a distancia.

Con el transcurso de los años estas herramientas se han ido modernizando y esto ha sido posible gracias al uso de estándares, que posibilitan la reutilización de Objetos de Aprendizaje (OA) sin importar el origen de los mismos.

La herramienta de autor CRODA es desarrollada en la Universidad de las Ciencias Informáticas (UCI) y permite la creación y edición de OA, así como el trabajo colaborativo de los profesores. CRODA sigue las indicaciones de los estándares LOM (para la descripción de metadatos) y SCORM en su versión 1.2 (para el empaquetamiento de contenidos).

En el presente trabajo se desarrolla un módulo para CRODA, que implementa la última versión del estándar SCORM (2004), siguiendo como principal objetivo que la herramienta de autor posibilite la asignación de secuencia y navegación en los Objetos de Aprendizaje, enriqueciendo la herramienta en cuanto a funcionalidades que faciliten el trabajo de los profesores y la superación de los educandos.

### **Palabras claves:**

E-learning, Estándares, herramientas de autor, Objetos de Aprendizaje (OA), Modelo de Referencia de Objetos de Contenidos Compartibles (SCORM), Secuencia y Navegación.

## *Índice*

Introducción .....	1
Capítulo 1: Fundamentación Teórica .....	5
1.1 Introducción al e-learning .....	5
1.2 Objetos de Aprendizaje .....	5
1.3 Especificaciones para la secuencia en los OA.....	6
1.4 Estándares que facilitan asignar secuencia y navegación en los OA.....	11
1.4.1 IMS Simple Sequencing .....	12
1.4.2 Modelo de Referencia de Objetos de Contenidos Compatibles (SCORM) ...	12
1.5 Herramientas que posibilitan asignar secuencia y navegación en los OA.....	15
1.5.1 Herramientas de autor .....	15
1.5.2 Repositorio de Objetos de Aprendizaje: RHODA .....	16
1.6 Estudio de las metodologías de desarrollo del software .....	17
1.6.1 Extreme Programming (XP) .....	17
1.6.2 Rational Unified Process (RUP).....	18
1.7 Lenguaje Unificado de Modelado (UML).....	20
1.8 Lenguajes de programación.....	20
1.8.1 Lenguajes del lado del cliente .....	20
1.8.2 Lenguajes del lado del servidor.....	21
1.9 Estudio de las tecnologías.....	22
1.9.1 Servidores Web .....	22
1.9.2 Gestores de base de datos .....	23
1.9.3 Base de datos de XML nativo .....	24
1.9.4 Frameworks .....	24
1.10 Estudio de las herramientas .....	26
1.10.1 Herramientas CASE .....	26
1.10.2 Entorno de desarrollo integrado (IDE) .....	27
1.10.3 Gestores de administración de Postgres .....	27

1.10.4	Clientes para control de versiones .....	28
	Conclusiones .....	29
	Capítulo 2: Características del sistema.....	30
	Introducción .....	30
2.1	Objeto de Automatización .....	30
2.2	Descripción del sistema propuesto .....	30
2.2.1	Modelo de Dominio .....	31
2.3	Requerimientos.....	31
2.3.1	Requerimientos funcionales .....	31
2.3.2	Requerimientos no funcionales.....	33
2.4	Modelo de Caso de Uso del sistema .....	35
2.4.1	Definición de actores del sistema.....	35
2.4.2	Definición de casos de uso.....	35
2.4.3	Diagrama de Casos de Uso del Sistema (DCUS).....	38
2.5	Descripción textual de los casos de uso del sistema .....	39
	Conclusiones .....	44
	Capítulo 3: Análisis y Diseño del módulo.....	45
	Introducción .....	45
3.1	Modelo de Análisis.....	45
3.1.1	Diagrama de Clases del Análisis.....	45
3.1.2	Diagramas de Interacción.....	48
3.2	Modelo de Diseño.....	52
3.2.1	Diagrama de Clases del Diseño.....	52
3.3	Diseño de la base de datos.....	55
3.4	Descripción de las tablas de la base de datos.....	55
	Conclusiones .....	58
	Capítulo 4: Implementación del módulo .....	59
	Introducción .....	59
4.1	Implementación .....	59

4.2	Diagrama de Despliegue .....	59
4.3	Diagrama de Componentes.....	60
	Conclusiones .....	63
	Conclusiones generales .....	64
	Bibliografía .....	66

## **Introducción**

Con el surgimiento de las nuevas Tecnologías de la Información y las Comunicaciones (TIC) aparece la necesidad de recibir las clases sin la presencia física del profesor. Para solucionar estos problemas y brindar una nueva vía de formación surge el e-learning, también conocido como educación a distancia haciendo uso de los medios tecnológicos.

Con el avance del e-learning resulta necesario compartir y reutilizar recursos, surgiendo así la tecnología de Objetos de Aprendizaje (OA). En el presente trabajo se toma como definición para esta tecnología, que un OA es: *“cualquier recurso con una intención formativa, compuesto de uno o varios elementos digitales, descritos con metadatos, que pueda ser utilizado y reutilizado dentro de un entorno e-learning”*. (Guzmán, 2005)

De esta forma se fueron transformando las clases magistrales a clases virtuales, sustituyendo las clases presenciales por plataformas educativas o Sistemas de Gestión de Aprendizaje (LMS), donde el estudiante encontrará toda la información necesaria para su formación. Entre los ejemplos de estas plataformas se encuentran: Moodle, Webct, Claroline.

Estas plataformas constituyen espacios en los que se permite la producción y publicación de cursos en la red. En las mismas *“el docente, teniendo en cuenta los objetivos educativos, autónomamente y con la posibilidad de editar en cualquier momento, pone a disposición de sus estudiantes mediante alguno de los recursos ofrecidos por la plataforma: el programa del curso, contenidos o unidades temáticas (expuestos en textos, hipertextos, presentaciones, animaciones, videos...), actividades, bibliografía y evaluación. De manera análoga, con otro nivel de autonomía, el estudiante puede acceder a los contenidos y al desarrollo de las actividades propuestas.”* (MaríaAgudelo, 2006)

Luego de surgir la tecnología de los Objetos de Aprendizaje fue necesaria la creación de herramientas de autor. Estas herramientas constituyen aplicaciones informáticas en las cuales se automatiza todo el proceso de creación y edición de los Objetos de Aprendizaje, que pueden ser almacenados en Repositorios de Objetos de Aprendizaje (ROA) para su posterior uso.

Para lograr la interoperabilidad y compatibilidad entre los OA desarrollados en distintas herramientas, se hace necesario el uso de diversos estándares, los cuales rigen el proceso de creación y edición de los OA, permitiendo que diferentes entidades puedan intercambiar información y utilizar conjuntamente los componentes sin necesidad de introducir modificaciones.

CRODA es una herramienta de autor desarrollada en la Universidad de las Ciencias Informáticas (UCI) y sigue las indicaciones de los estándares LOM y SCORM, utilizados para la descripción de metadatos y el empaquetamiento de los contenidos, respectivamente.

CRODA en su primera versión implementa SCORM 1.2 sin tener en cuenta la secuenciación y navegación de los contenidos, pues no permite la comunicación entre los diferentes Objetos de Contenido Compartible (SCO) que forman parte del OA. Esta característica limita el proceso de formación del OA, impidiéndole al profesor establecer diferentes rutas de aprendizaje dependiendo del desempeño del alumno dentro del Objeto de Aprendizaje. Además no se puede establecer un orden de los contenidos de enseñanza que asegure el enlace entre los objetivos educativos y las actividades de aprendizaje de los estudiantes, impidiendo la consecución de las intenciones formativas propias del programa de formación.

Estas limitaciones imposibilitan un acercamiento progresivo desde la situación inicial de aprendizaje de los alumnos hasta los objetivos propuestos para el programa formativo, el cual es distinto para cada situación de aprendizaje, colectivo de destinatarios de la formación, e incluso para cada individuo.

Después de analizar la situación presentada se plantea el siguiente **problema de investigación** ¿Cómo facilitar la asignación de secuencia y navegación en los Objetos de Aprendizaje de la herramienta de autor CRODA? Este problema se enmarca en el **objeto de estudio** la secuencia y navegación en los Objetos de Aprendizaje. Por lo que se determina para el desarrollo del problema como **objetivo general** posibilitar la asignación de secuencia y navegación en los Objetos de Aprendizaje de la herramienta de autor CRODA. Se delimita el **campo de acción** a las tecnologías que facilitan la asignación de secuencia y navegación en los Objetos de Aprendizaje.

Se plantea como **idea a defender** que con la implementación del estándar SCORM 2004 en la herramienta de autor CRODA, se posibilitará la asignación de secuencia y navegación en los OA.

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

- Analizar las tecnologías que permiten la asignación de secuencia y navegación en los OA.
- Realizar el análisis y diseño del módulo SCORM 2004.
- Implementar el módulo SCORM 2004 en la herramienta de autor CRODA con el uso de las tecnologías escogidas

Para dar cumplimiento a los objetivos planteados se trazaron las siguientes **tareas**:

- Selección y revisión bibliográfica.
- Procesamiento y evaluación de la información obtenida.
- Estudio de las tecnologías que posibilitan la asignación de secuencia y navegación en los OA.
- Elaboración de los artefactos requeridos en el Modelo de Casos de Uso del Sistema.
- Obtención de los artefactos requeridos en el Modelo de Análisis.
- Desarrollo de los artefactos requeridos en el Modelo de Diseño.
- Elaboración de los artefactos requeridos en el Modelo de Implementación.

**Métodos de investigación utilizados:**

**Métodos teóricos:**

- **Analítico-Sintético:** para identificar los conceptos y definiciones más importantes relacionados con la creación de contenidos y uso del estándar SCORM 2004 en herramientas para la creación de contenidos.
- **Histórico-Lógico:** para llevar a cabo un estudio de las diferentes herramientas de autor que utilizan SCORM 2004 en la actualidad, así como sus características, ventajas y evolución.

- **Modelación:** para confeccionar diagramas que permitan modelar aspectos que faciliten la implementación.

**Métodos empíricos:**

- **Observación:** para observar el comportamiento de las herramientas existentes y recoger información relevante a medida que se desarrolla el proyecto.

**Estructura capitular:**

**Capítulo 1: Fundamentación teórica.**

Contiene los principales conceptos asociados al dominio del problema, así como el sustento teórico de la investigación.

**Capítulo 2: Características del sistema.**

Describe las principales características del módulo a desarrollar, así como sus requerimientos funcionales y no funcionales.

**Capítulo 3: Análisis y diseño del módulo.**

Se realizan los artefactos requeridos en el flujo de trabajo Análisis y Diseño de la metodología de desarrollo de software RUP.

**Capítulo 4: Implementación del módulo.**

Se realizan los artefactos requeridos en el Modelo de Implementación y se obtiene el resultado final del sistema.

## *Capítulo 1: Fundamentación Teórica*

### **Capítulo 1: Fundamentación Teórica**

En el presente capítulo se precisan elementos teóricos que sustentan la investigación y desarrollo de un módulo que permita la asignación de secuencia y navegación en los Objetos de Aprendizaje de la herramienta de autor CRODA. Se realiza un análisis exhaustivo del estado del arte del tema a investigar, teniendo en cuenta las herramientas existentes hasta el momento que facilitan la asignación de secuencia y navegación en los OA. Se hace un estudio de las herramientas, lenguajes y tecnologías a utilizar.

#### **1.1 Introducción al e-learning**

En la actualidad existen diferentes necesidades de información. La educación tradicional presenta algunas dificultades en cuanto a tiempos, sincronización de agendas, asistencia y viajes. Para dar solución a estos problemas surge el e-learning y con ello una nueva forma de educación, permitiendo recibir clases sin la presencia física del profesor.

El término e-learning es comprendido por la mayoría de autores como: el suministro de programas educacionales y sistemas de aprendizaje a través de medios electrónicos. El e-learning se basa en el uso de una computadora u otro dispositivo electrónico para proveer a las personas de material educativo.

*“Esta modalidad de aprendizaje puede involucrar una mayor variedad de equipo que la educación en línea. El término de e-learning o educación electrónica abarca un amplio paquete de aplicaciones y procesos, como el aprendizaje basado en Web, capacitación basada en computadoras, salones de clases virtuales y colaboración digital (trabajo en grupo).”* (Mendoza, 2003)

#### **1.2 Objetos de Aprendizaje**

A partir de las modificaciones del aprendizaje a distancia surge el término de Objetos de Aprendizaje. Dentro de las principales características de los OA podemos destacar: contenido educativo, reutilizables, modificables, interactivos, formatos y caracterización estandarizados y granularidad variable. La característica modificable, responde a la necesidad de crear OA en su estructura más simple para lograr su posterior uso adaptándolo a las necesidades de su autor.

Para garantizar la reusabilidad de un OA, este debe estar descrito por un grupo de metadatos donde se recoge información referente a: nombre, autor, fecha de creación, descripción, versión, localización, entre otros.

*“La posibilidad de que los OA sean reutilizados en diferentes contextos y plataformas refleja que deben tener la capacidad de ser interoperables, es decir, importados y exportados sin presentar problemas de compatibilidad, lo cual implica la necesidad de usar estándares para la creación de los OA.”* (Guzmán, 2005)

Con la finalidad de que los contenidos de un Objeto de Aprendizaje sean asumidos con la calidad requerida por los estudiantes, es de vital importancia realizar la organización y secuenciación de dichos contenidos, para lo cual se tienen en cuenta un conjunto de especificaciones.

### **1.3 Especificaciones para la secuencia en los OA**

La actividad docente es un fenómeno complejo de intenciones, acciones e intervenciones, que se determinan y condicionan mutuamente, encaminadas a conseguir fines educativos concretos.

En esta conceptualización los recursos y las actividades se sitúan en un esquema complejo de contenidos y condiciones de aprendizaje, donde el centro lo constituye el ciudadano en situación de formación.

Entre los elementos que determinan estas condiciones de aprendizaje se encuentran la organización y la secuencia de los recursos educativos que se pongan a disposición de los alumnos.

Como se puede apreciar estos elementos *constituyen el punto neurálgico del itinerario que conduce a diseñar los procesos de Aprendizaje.* (Ros, 2009)

Para plantear la organización y secuenciación de contenidos existen cuatro presupuestos básicos que se necesitan tener presentes y que permiten que el trabajo en equipo sea eficaz. A continuación se mencionan estos presupuestos:

- *Los objetivos generales establecidos para cada nivel de planificación, acordados previamente, tendrán que incidir en la acción formativa para cada nivel inferior de planificación a través de los objetivos generales establecidos para ese nivel y para los contenidos de ese nivel.*

- *En el alumno, en un caso óptimo, se van a desarrollar las capacidades y se van a incorporar los contenidos de aprendizaje establecidos para el nivel módulo o materia correspondiente, pero también se van a desarrollar otras capacidades o conocimientos no previstos, o en la correspondiente etapa de desarrollo personal se deben desarrollar otros conocimientos que corresponden a esa etapa. Por tanto los objetivos generales de la unidad o nivel que estamos trabajando, vinculados a contenidos particulares de los tres tipos indicados, han de orientarse a la formación integral de todos los alumnos.*
- *La aceptación de los principios del aprendizaje significativo comporta concebir los procesos de enseñanza-aprendizaje desde la óptica de la construcción del conocimiento por parte del alumno.*
- *La concepción de la formación vigente y universalmente aceptada exige, como imperativo ético, que se diseñen procesos de enseñanza-aprendizaje adecuadamente diferenciados, con el fin de poder atender a la diversidad de capacidades e intereses de los alumnos. (Ros, 2009)*

Existen varias técnicas y teorías clásicas, así como modelos que tratan el problema y la justificación teórica de la secuenciación de los contenidos de aprendizaje. A continuación se exponen los planteamientos más significativos de estas técnicas, teorías y modelos.

- **Técnica de Análisis de Contenidos**

En esta técnica se plantea que el proceso para secuenciar contenidos de aprendizaje consta de tres pasos:

1. Descubrir y destacar los ejes vertebradores de los contenidos que deben enseñarse a los alumnos.
2. Descubrir y destacar los contenidos fundamentales y organizarlos en un esquema jerárquico y relacional.
3. Proceder a la secuenciación según los principios de la organización psicológica del conocimiento.

También se considera que la secuenciación debe hacerse teniendo en cuenta 3 criterios generales:

-Primer criterio: La elaboración de secuencias de aprendizaje por parte de los profesores supone considerar la estructura del contenido de enseñanza que hay que proponer a los alumnos y, a la vez, la manera como los alumnos construyen su propio conocimiento.

- Segundo criterio: Los contenidos seleccionados como fundamentales deben ser los que tienen mayor capacidad de inclusión, es decir, los que pueden integrar otros contenidos que los alumnos también tendrán que aprender; y cuantos más contenidos puedan integrar, mejor.

- Tercer criterio: En primer lugar hay que presentar los conceptos más generales e inclusivos, dejando para después los aspectos más concretos y los más irrelevantes.

Estos criterios guiarán la organización de los bloques de contenido para el proceso de aprendizaje de los alumnos.

- **Técnica de Análisis de Tareas**

La técnica de análisis de tareas permite secuenciar los contenidos en términos de resultados esperados del aprendizaje de los alumnos. Según esta técnica, el proceso a seguir para secuenciar los contenidos de enseñanza consta de tres pasos:

1. Determinar la tarea que el alumno debe realizar (habilidad que tiene que aprender).
2. Determinar los posibles componentes de la tarea o habilidad (sub-tareas o sub-habilidades).
3. Secuenciar las sub-tareas o sub-habilidades, de la más sencilla a la más compleja.

Es una técnica útil para secuenciar procedimientos enfocados al desarrollo de determinadas habilidades o destrezas, pero no está suficientemente desarrollada para garantizar una correcta y adecuada secuenciación de los contenidos de enseñanza y el correspondiente proceso de aprendizaje de los alumnos.

- **Teoría de la Elaboración**

La Teoría de la Elaboración integra elementos de las técnicas antes mencionadas, mejorando sus deficiencias. Es muy recomendada como pauta para la secuenciación del aprendizaje y se fundamenta en el siguiente principio:

*Los contenidos de enseñanza tienen que ordenarse de manera que los elementos más simples y generales ocupen el primer lugar, incorporando después, de manera progresiva, los elementos más complejos y detallados. (Ros, 2009)*

Es una teoría que persigue prescribir criterios para seleccionar, secuenciar y organizar los contenidos educativos, logrando una óptima adquisición, retención y transferencia de estos contenidos.

- **Modelo de Trabajo de Síntesis**

Este modelo proporciona una orientación explícita para el alcance y la secuencia en que pueden integrarse los Objetos de Aprendizaje. Según este modelo los objetos deben secuenciarse en un orden que simula el mundo real, logrando que el rendimiento aumente en proporción a la realidad.

Además se plantea que como puede crearse más de un Objeto de Aprendizaje, partiendo de un único modelo de trabajo y en función de los mismos objetivos de aprendizaje, se establece entonces una relación de equivalencia entre los Objetos de Aprendizaje a la hora de secuenciarlos, lo cual permite sustituir unos por otros en la secuencia.

- **Teoría del Dominio**

Es una teoría que proporciona pautas sobre alcance y secuencia, donde se ve claramente que la experiencia es un factor que regula la secuenciación. Se plantea que los Objetos de Aprendizaje deben secuenciarse teniendo en cuenta su dificultad, partiendo de las experiencias personales.

- **Modelo de Diseño Instruccional de cuatro componentes de Van Merriënboer**

El modelo fue creado para brindar apoyo explícito para establecer el alcance y la secuencia de Objetos de Aprendizaje. Referenciando este modelo se deduce que los Objetos de Aprendizaje deben ser secuenciados en función de su nivel y tipo, promoviendo la transferencia cuando sea factible.

En el modelo se plantea que las agrupaciones de habilidades de nivel macro deben ser secuenciadas como anexos de una tarea y progresivamente en función de la carga cognitiva. También se plantea que los objetos de meso-nivel deben ser secuenciados en función de una tarea común a todas las habilidades que se enseñan al mismo tiempo. Para el caso del nivel micro, se plantea que los problemas específicos pueden ser

secuenciados en conjunto, de fácil a complejo o, cuando sea factible, en una secuencia aleatoria con el fin de promover la transferencia.

- **Teoría del Diseño y la Organización de Objetos de Aprendizaje (LODAS)**

Es una teoría general para la organización y secuenciación de Objetos de Aprendizaje, resultado de revisar, sintetizar y combinar la Teoría de la Elaboración, el Modelo de Trabajo de Síntesis, la Teoría del Dominio y el Modelo de Diseño Instruccional de cuatro componentes de Van Merriënboer. Además en esta se establecen las directrices para:

*a) El análisis del contenido, diferenciándolo en elementos de aprendizaje.*

*b) La síntesis y organización conjunta en un esquema de relaciones operativas y conceptuales, en ambos casos de los contenidos de aprendizaje de un área de contenido indiferenciada inicialmente (por ejemplo, Matemáticas o Psicología).*

*c) La aplicación de las especificaciones que se derivan de la organización para el alcance y la secuencia de Objetos de Aprendizaje. (Ros, 2009)*

- **Modelo Simplificado y Adaptativo de Diseño de Objetos de Aprendizaje y Secuenciación (SAM)**

El modelo SAM es fruto de la combinación y síntesis de métodos y teorías existentes, fundamentado básicamente en la Teoría de la Elaboración. En él se han hecho algunos aportes siguiendo los criterios de simplificación y adaptabilidad. Además en este se define el tipo de apoyo para la secuencia, necesario para soportar el uso instruccional exclusivo y explícito de Objetos de Aprendizaje en el entorno de la práctica docente.

*SAM ha sido diseñado para cumplir con cinco objetivos principales:*

- 1. Fomentar y catalizar el diálogo en torno a la utilización de Objetos de Aprendizaje en un contexto de diseño instruccional.*
- 2. Proporcionar apoyo explícito para el diseño de Objetos de Aprendizaje.*
- 3. Proporcionar un apoyo explícito a la secuencia de Objetos de Aprendizaje.*
- 4. Proporcionar apoyo en el uso de objetos en el aprendizaje de modo reutilizable.*
- 5. Proporcionar una base de compatibilidad con experiencias expertas en dominios concretos e investigaciones en modelos de aprendizaje. (Ros, 2009)*

En resumen se puede afirmar que las especificaciones derivadas de teorías, técnicas y modelos para la secuenciación, desempeñan un importante papel en la automatización de procesos de diseño instruccional. Por lo tanto, el objetivo principal de cualquier proyecto de diseño instruccional es proporcionar el alcance y la secuencia de orientación para la organización y despliegue de Objetos de Aprendizaje, lo cual permite establecer los enlaces entre las intenciones y los objetivos educativos en el aprendizaje.

#### **1.4 Estándares que facilitan asignar secuencia y navegación en los OA**

El nuevo enfoque de enseñanza descrito, basado en la existencia, uso compartido y reutilización de Objetos de Aprendizaje aporta beneficios innegables. Sin embargo, la sola utilización de Objetos de Aprendizaje no es suficiente para transformar la enseñanza tradicional en una nueva forma de enseñanza que proporcione todos esos beneficios. Es aquí donde entran en escena los estándares de e-learning.

La existencia de estándares que definan particularidades como la forma de empaquetar los Objetos de Aprendizaje o la secuenciación y navegación de los contenidos, resulta esencial para el desarrollo con éxito de los sistemas de e-learning. Entre los beneficios derivados de la estandarización se encuentran:

- *Accesibilidad del contenido, que estará disponible en cualquier momento y desde cualquier lugar.*
- *Interoperabilidad, entendida como la capacidad de que componentes desarrollados por distintas entidades puedan intercambiar información y ser utilizados conjuntamente.*
- *Reusabilidad de los contenidos como forma de economizar esfuerzos a la hora de crear nuevos contenidos educativos.*
- *Extensibilidad, o capacidad de ampliación, gracias a la construcción modularizada de contenidos.*
- *Facilidad de localización de los contenidos almacenados en repositorios que utilizan metadatos como forma de catalogación.*
- *Coste razonable, pues la estandarización reduce los costes de desarrollo.*
- *Facilidad de gestión de los contenidos, pues el diseño en pequeñas unidades modulares facilita los cambios y actualizaciones.*

- *Perdurabilidad, pues el desarrollo de contenidos estándar evita la obsolescencia de los mismos ante cambios en las plataformas. (Etxeberría, 2007)*

Se puede afirmar, que la estandarización fomenta la comunicación y el intercambio, lo cual posibilita que las organizaciones que generan contenidos obtengan rendimientos adicionales sobre sus inversiones. También con la estandarización se potencia el desarrollo de herramientas para la creación y gestión de contenidos estandarizados, propiciando la eficacia del aprendizaje. (Etxeberría, 2007)

Entre los estándares que se refieren a la secuencia y la navegación en los OA se destacan IMS Simple Sequencing y SCORM, los cuales se encuentran estrechamente vinculados pues SCORM en su versión 1.3 ha adaptado su modelo de agregación (CAM) incorporando las especificaciones de IMS Simple Sequencing para la secuenciación de contenidos.

#### **1.4.1 IMS Simple Sequencing**

IMS Simple Sequencing permite la definición de recorridos pedagógicos que el usuario puede seguir para atravesar los recursos. Define un método para representar el comportamiento previsto de una experiencia de aprendizaje, posibilitando que cualquier Sistema de Tecnología de Aprendizaje (LTS) pueda secuenciar las actividades de aprendizaje de una manera consistente.

Especifica los comportamientos requeridos y la funcionalidad que los sistemas de ajuste deben implementar. Incorpora reglas que describen la ramificación o flujo de instrucción a través de contenido de acuerdo a los resultados de las interacciones de un alumno con el contenido. (autores, 2011)

IMS Simple Sequencing, aunque rica y compleja, recibe la denominación *simple* debido a que está orientada a soportar únicamente los patrones de secuenciamiento más habituales en procesos de aprendizaje.

#### **1.4.2 Modelo de Referencia de Objetos de Contenidos Compartibles (SCORM)**

SCORM es creado por Advanced Distributed Learning (ADL). Constituye una recopilación de especificaciones y estándares adaptados de múltiples fuentes para ofrecer un paquete muy completo de funciones de aprendizaje que permiten la interoperabilidad, accesibilidad y la reutilización del contenido de aprendizaje.

Surge a raíz de las necesidades de mezclar contenidos de diferentes orígenes, para que estos sean intercambiables y reutilizables. Los OA creados bajo las indicaciones de SCORM pueden ser exportados para posteriormente ser interpretados por otros LMS sin presentar problemas de compatibilidad, esto constituye una de las ventajas del uso de este estándar. Además permite conocer el avance de los estudiantes en el transcurso de la actividad (a través de evaluaciones o reportes de contenidos).

Un contenido cumple con SCORM si está:

- Diseñado para ser exhibido en un navegador.
- Descrito por metadatos.
- Organizado como un conjunto estructurado de objetos más pequeños.
- Empaquetado de tal forma que pueda ser cargado en cualquier plataforma SCORM compatible.
- Creado para ser portable, de forma que pueda ser distribuido por cualquier servidor web en cualquier sistema operativo. (Cely, 2010)

SCORM ofrece grandes beneficios para el aprendizaje y a continuación se mencionan los más significativos:

- *Transferible. Capacidad de usar el contenido en cualquier LMS SCORM certificado por ADL.*
- *Reutilizable. Capacidad de buscar y recuperar objetos de contenido, incluyendo lecciones, módulos, ejercicios, actividades, medios, etc., y de reutilizarlos.*
- *Monitoreo de datos. Capacidad de monitorear datos sobre el aprendiz y su experiencia incluyendo calificaciones, tiempo, comentarios, etc.*
- *Secuenciable. Capacidad de combinar objetos de contenido para lograr estrategias de secuencia o estratificación. (Cármaco, 2010)*

SCORM posee varias versiones pero las más conocidas y empleadas son la 1.2 y 1.3. A continuación se presentan algunas de las características de estas versiones.

- **SCORM 1.2**

SCORM 1.2 es una versión muy empleada. Soportada por la mayoría de los LMS que existen y responde básicamente a la necesidad de cargar y trasladar contenidos entre diferentes plataformas, además establece un sistema de secuenciación, que permite pequeñas variaciones, aunque estas no evitan, que en definitiva, los cursos tengan una estructura demasiado lineal. Esta característica le impide al profesor establecer diferentes rutas de aprendizaje dependiendo del desempeño del alumno dentro del curso.

- **SCORM 2004**

Esta versión del estándar aparece para resolver los problemas y ambigüedades que las versiones anteriores no pudieron solucionar. En esta versión se define el modelo de secuenciamiento y navegación de contenidos (SN - Sequencing and Navigation), el cual a su vez define la secuencia de contenidos para un usuario (bien generada por el sistema gestor del aprendizaje o bien fruto de la interacción explícita del usuario). Además define cómo interpretar las reglas de secuenciación asociadas a los contenidos y basa la navegación en un árbol de actividades. (ADL, 2006)

La secuenciación y navegación constituye una nueva característica, que permite describir el flujo de navegación dentro del contenido, así como introducir restricciones para evitar que un alumno vea una sección sin haber primero aprobado la anterior. También incluye la posibilidad de especificar una secuencia de actividades flexible a partir de objetos de contenido, posibilitando compartir y usar información de acuerdo con el estatus de múltiples objetivos de aprendizaje alcanzados por los estudiantes en un mismo LMS. Esta característica permite que se puedan trazar rutas de aprendizaje personalizado. (Cármaco, 2010)

Además facilita la creación de una experiencia de aprendizaje dinámica, compatible con cualquier LMS que soporte el estándar y posibilita la reutilización de los Objetos de Aprendizaje para crear contenidos más complejos y completos de acuerdo a las necesidades del desarrollador. (Anónimo, 2010)

Las características mencionadas de SCORM 2004 hacen factible su aplicación en el módulo a desarrollar, esta idea se sustenta también en que el equipo de desarrollo posee experiencia de trabajo con el estándar SCORM.

## **1.5 Herramientas que posibilitan asignar secuencia y navegación en los OA**

### **1.5.1 Herramientas de autor**

*“La preparación de materiales para el proceso de enseñanza aprendizaje constituye el eje central de cualquier estrategia pedagógica, mucho más si las relaciones alumno-profesor son mediatizadas. Por eso, la publicación de un curso mediado por las TIC, va precedido de un proceso en el cual se preparan y seleccionan los materiales educativos digitales que se insertan en él, llamado producción de cursos en formato digital.”* (Herrero, 2008)

Con la ayuda de las herramientas de autor, el trabajo de los diseñadores instruccionales se hace mucho más sencillo, pues agrupan un conjunto de funcionalidades que permiten crear un OA, ahorrando tiempo y esfuerzo.

*“Las herramientas de autor fueron desplegadas para la creación de materiales educativos y publicación de cursos, son aplicaciones que permiten un trabajo constructivista para generar un entorno de aprendizaje dinámico, dentro de las funcionalidades que este tipo de herramientas presentan se puede destacar la posibilidad de crear actividades o pequeñas aplicaciones desde la misma herramienta.”* (CARRODEGUAS, 2008)

- **Reusable eLearning Object Authoring & Delivery (RELOAD)**

Es una herramienta para crear y editar paquetes e insertar metadatos conforme a las especificaciones de ADL e Instructional Management Systems (IMS).

Los objetivos principales de RELOAD son:

- Facilitar la creación, el intercambio y la reutilización de OA.
- Aumentar la gama de enfoques pedagógicos de realización a través de la utilización de planes de lecciones.

La versión 2.5.5 incluye soporte para IEEE LOM, SCORM 2004, IMS CP últimas versiones. La implementación de la versión 2004 de SCORM posibilita la creación de OA más dinámicos en los que el diseñador puede establecer diferentes rutas de aprendizaje dependiendo del desempeño del estudiante.

*“Un inconveniente de dicha herramienta es su editor, el cual está en inglés, y aunque posee la elección de cambiar a otros idiomas, es limitado, modifica sólo algunos términos. Además de necesitar de un navegador Web para ver el contenido de los paquetes.”* (ÁLVAREZ, 2009)

- **UDUTU**

Es una herramienta que permite la creación de cursos en línea completamente multimedia a través de la web, donde podrán añadir a sus librerías contenidos en flash, audio, video, documentos, imágenes y otros. Los cursos estarán guiados a través de una serie de opciones y se crearán sus propias páginas partiendo de una serie de plantillas.

Mediante UDUTU se puede obtener un resultado similar al que proporciona una presentación de PowerPoint. De hecho, permite importar presentaciones en este formato. Una de las virtudes de esta aplicación es que permite trabajo colaborativo. Desde el panel de administración se puede invitar a usuarios y gestionar los historiales de cambios.

Una vez creado el curso este puede ser exportado a formatos como SCORM 1.2 y SCORM 2004, lo cual constituye una gran ventaja, haciéndolo compatible con todos los LMS que permiten la importación. (Cazarez, 2010)

### **1.5.2 Repositorio de Objetos de Aprendizaje: RHODA**

Es un Repositorio de Objetos de Aprendizaje desarrollado en la Universidad de las Ciencias Informáticas (UCI). Constituye un espacio creado para apoyar a los usuarios en la gestión de recursos didácticos, utilizando la tecnología de Objetos de Aprendizaje.

Es un lugar para el trabajo metodológico colaborativo, orientado a elevar la calidad de los recursos didácticos. Los Objetos de Aprendizaje que en él se encuentran son arbitrados por revisores y su creación está asociada al proceso de producción del Laboratorio para la Producción de Recursos Didácticos. (autores, 2010)

Los Objetos de Aprendizaje pueden ser seleccionados para complementar cursos que se encuentran en el Entorno Virtual de Aprendizaje o para la auto-superación de estudiantes y profesores.

La herramienta incorpora funcionalidades como la exportación de paquetes según el estándar SCORM en sus dos versiones, de esta forma, el diseñador instruccional puede seleccionar la que desea utilizar de acuerdo a sus necesidades.

## 1.6 Estudio de las metodologías de desarrollo del software

### 1.6.1 Extreme Programming (XP)

“La Programación Extrema es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado.” (Escribano, 2002)

XP es una metodología que se propone a emplear en proyectos de riesgo, pues aumenta la productividad. Surgió como respuesta y posible solución a los problemas derivados del cambio en los requerimientos. Se caracteriza por ser una metodología ágil, en la que se combinan las mejores prácticas para el desarrollo de software. Se diferencia de las metodologías tradicionales principalmente porque pone más énfasis en la adaptabilidad que en la previsibilidad.

La metodología XP propone seis fases para el desarrollo del ciclo de vida del software:

- Exploración
- Planificación de la entrega
- Iteraciones
- Producción
- Mantenimiento
- Muerte del proyecto

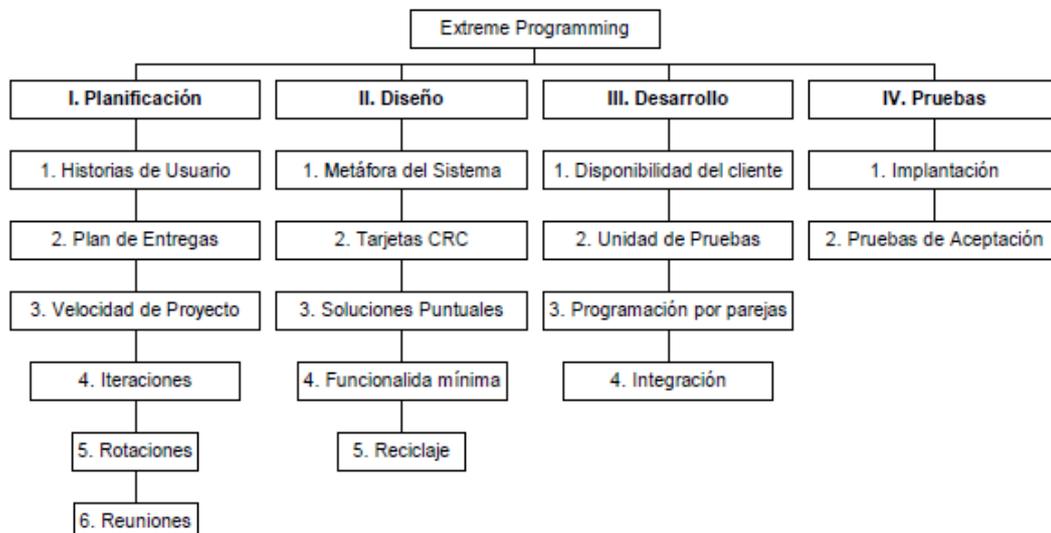


Figura 1: Trabajando con la metodología XP. (Escribano, 2002)

Extreme Programming trata de minimizar la complejidad de un proyecto y se enfoca directamente en su objetivo. Una de las características más interesantes de esta

metodología es que permite la programación en parejas, posibilitando que dos programadores trabajen en una misma estación, logrando así un desempeño del trabajo más eficiente. Dentro de las ventajas que proporciona cabe mencionar que facilita la rapidez de reacción pues hace uso de las relaciones interpersonales para desarrollar el software. Además se llevan a cabo pruebas unitarias, basadas en pruebas hechas a los procesos de mayor importancia, para de esta forma minimizar los riesgos a causa de errores. (Escribano, 2002)

### **1.6.2 Rational Unified Process (RUP)**

El Proceso Unificado de Software es una metodología tradicional pensada para la Ingeniería de Software, la cual proporciona un conjunto de técnicas que soportan el ciclo completo de desarrollo del software.

RUP unifica completamente a un equipo de desarrollo, optimizando la productividad de cada uno de los miembros del mismo. Las principales características de esta metodología de desarrollo son:

- **Iterativo e incremental:** Donde cada fase se desarrolla en iteraciones, de forma tal que se pueda dividir en pequeños proyectos mejorando su comprensión y desarrollo. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente unos más que otros.
- **Dirigido por casos de uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo, pues los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.
- **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. (Jacobson, 2000)

El desarrollo del software en RUP es dividido en cuatro fases:

- **Inicio:** Alcanzar un acuerdo entre todos los interesados respecto a los objetivos del ciclo de vida para el proyecto, generar el ámbito de trabajo, el caso de negocio, síntesis de arquitectura posible y el alcance del proyecto.

- **Elaboración:** Establecer la línea base para la arquitectura del sistema y proporcionar una base estable para el diseño y el esfuerzo de implementación de la siguiente fase, mitigando la mayoría de los riesgos tecnológicos.
- **Construcción:** Completar el desarrollo del sistema basado en la línea base de la arquitectura. En otras palabras lograr la funcionalidad operativa del software.
- **Transición:** Garantizar que el software esté listo para entregarlo a los usuarios y lograr la aprobación cuanto antes para liberar el producto al mercado.

RUP propone nueve flujos de trabajo, de los cuales seis son obligatorios o fundamentales y tres son opcionales o de apoyo.

#### **Flujos Obligatorios:**

- **Modelo del negocio:** Describe los procesos de negocio, identifica quiénes participan y las actividades que requieren automatización.
- **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- **Análisis y Diseño:** Describe cómo el sistema será realizado a partir de las funcionalidades previstas y las restricciones impuestas (requerimientos), indica con precisión lo que se debe programar.
- **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- **Pruebas:** Busca los defectos a lo largo del ciclo de vida del software.
- **Despliegue:** Produce el release del producto y realiza las actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregarle el software a los usuarios finales.

#### **Flujos de opcionales:**

- **Administración del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades del cliente.
- **Administración de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización y actualización concurrente de elementos, control de versiones, etc.
- **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto, así como el procedimiento para implementar el proceso en una organización. (Jacobson, 2000)

RUP se selecciona como metodología de desarrollo de software, por las características antes mencionadas y por ser la escogida en la versión 1.0 de la herramienta CRODA.

## 1.7 Lenguaje Unificado de Modelado (UML)

UML es un lenguaje de modelado que permite a los creadores de sistemas, generar diseños que capturen sus ideas en una forma convencional y fácil de comprender para comunicarlas a otras personas. Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas, la finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo.

*“Es importante destacar que un modelo UML describe lo que supuestamente hará el sistema, pero no dice cómo hacerlo.”* (Schmuller, 2008)

*“UML es un lenguaje de modelado visual que se usa para visualizar, especificar, construir y documentar artefactos de un sistema de software.”* (Jacobson, 2000)

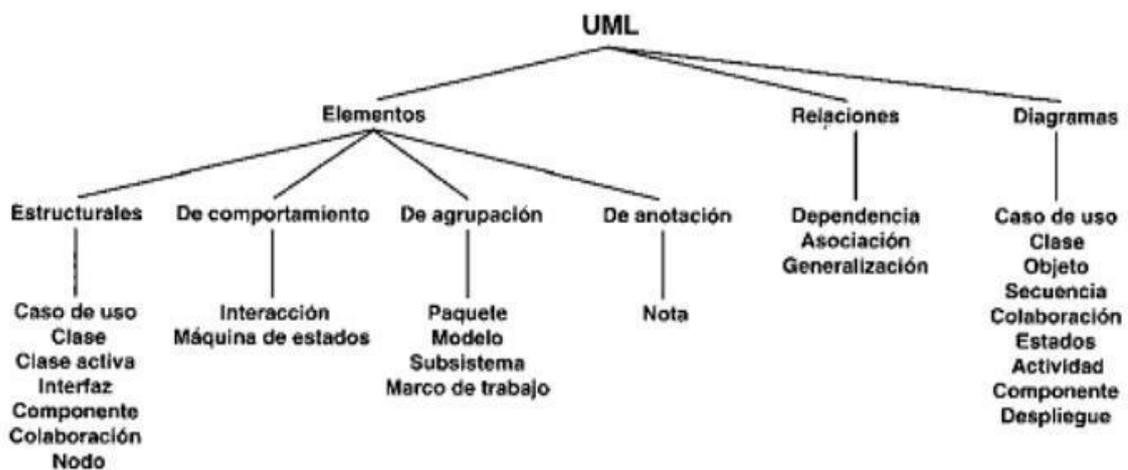


Figura 2: Aspectos de UML. (Jacobson, 2000)

Por lo anteriormente mencionado, el lenguaje unificado de modelado (UML) será utilizado como soporte de modelación para la implementación del módulo propuesto.

## 1.8 Lenguajes de programación

### 1.8.1 Lenguajes del lado del cliente

#### XHTML

XHTML es un lenguaje muy similar al HTML, con algunas diferencias que lo hacen más potente y aconsejable para la modelación de páginas web. Sus siglas se derivan de las palabras inglesas eXtensible Hypertext Markup Language y significan Lenguaje Extensible de Marcación Hipertexto.

Este lenguaje constituye la versión XML de HTML y adquiere la rigidez del XML, además incorpora una nueva concepción para la modelación de páginas web, en la cual se busca la creación de una web semántica, especificando sólo el significado de cada elemento y no cómo se debe visualizar el elemento, lo cual permite separar el contenido de la presentación. (Alvarez, 2011)

## **CSS**

Las “Hojas de estilo en cascada”, por sus siglas en inglés CSS, tienen como principio utilizar un solo documento para almacenar las características de presentación de las páginas asociadas a grupos de elementos. Entre sus utilidades se destacan lograr una apariencia uniforme en toda la aplicación y cambiar aspectos del sitio en pocas líneas de código. (Jeff, 2008)

Es un lenguaje que al avanzar en sus versiones provee una mayor flexibilidad, aplicando nuevas reglas para ahorrar tiempo y haciendo más fácil la reconstrucción de efectos previamente complejos.

## **JavaScript**

Es un lenguaje que se utiliza para desarrollar pequeños programas que son insertados en páginas web. Ofrece muchas posibilidades y permite crear diferentes efectos e interactuar con los usuarios. La mayor parte de su programación está dirigida a describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros. Existen dos tipos de JavaScript, uno que se ejecuta en el cliente denominado Navigator JavaScript y otro que se ejecuta en el servidor, conocido como LiveWire JavaScript. (Valdés, 2007)

*“Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.”* (Pérez, 2007)

### **1.8.2 Lenguajes del lado del servidor**

## **PHP**

*“PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas Web dinámicas, embebido en páginas HTML y ejecutado en el servidor.”* (Hinostraza, 2007)

Es un lenguaje ideal que entre sus principales características se destacan la rapidez, su facilidad de aprendizaje, su soporte multiplataforma tanto de diversos Sistemas Operativos, como servidores HTTP y de bases de datos, así como el hecho de que se distribuye de forma gratuita bajo una licencia abierta.

PHP está soportado por una gran comunidad de programadores y es un potente lenguaje, que consta de un código actualizado continuamente, con mejoras y extensiones que amplían sus capacidades.

## **1.9 Estudio de las tecnologías**

### **1.9.1 Servidores Web**

Un servidor web es *“un programa que sirve para atender y responder a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten usando el protocolo HTTP o el protocolo HTTPS”*. (Cibernetia, 2009)

#### **Internet Information Services (IIS)**

IIS constituye un servidor web que dispone de una infraestructura con gran fiabilidad, capacidad de manejo y escalabilidad para aplicaciones web, aumentando la disponibilidad de las mismas y reduciendo los costes administrativos. Además permite la monitorización de salud de estado automático, aislamiento de procesos y capacidad de gestión mejorada. (autores, 2007)

Su principal limitante para utilizarlo en la solución del problema es que este servidor solo funciona con el sistema operativo Windows, por lo que constituye una propiedad de Microsoft Corporation.

#### **Cherokee**

Cherokee es un servidor web de alto rendimiento, flexible y fácil de configurar. Constituye el servidor más rápido que existe. Proporciona una estabilidad sólida como una roca. Entre sus características una que merece especial crédito es su amigable interfaz de usuario, llamada cherokee-admin. Esta interfaz de administración permite configurar el servidor web sin tener que preocuparse por cómo modificar un archivo de texto escrito con una sintaxis determinada. Cherokee es software libre, tiene una arquitectura bien diseñada, es totalmente modular y tiene una base de código limpio y ordenado. (Ortega, 2010)

#### **Apache**

Apache es un potente y gratuito servidor web que proporciona flexibilidad y eficiencia, puede funcionar sobre diferentes plataformas y es desarrollado de forma abierta. Es modular, pues proporciona diferentes módulos de apoyo. (Félix, 2000)

También permite adaptarse a diversos entornos y necesidades, posibilitando que se desarrollen diferentes extensiones. En nuestros días es el servidor web más utilizado a nivel mundial.

Por las características antes mencionadas se selecciona Apache como servidor web a utilizar en el desarrollo del módulo. Además es el servidor empleado hasta el momento en el proyecto CRODA.

### **1.9.2 Gestores de base de datos**

*“Un sistema gestor de bases de datos (SGBD) es un software que proporciona a los usuarios servicios para la creación, almacenamiento y procesamiento de bases de datos de forma segura y eficiente, sirviendo de intermediario entre aplicaciones y datos”.*  
(Sicilia, 2008)

#### **Oracle(Relational Data Base Management System)**

Oracle es uno de los sistemas gestores de bases de datos más destacados, basado en la tecnología cliente/servidor que entre sus potencialidades presenta soporte multiplataforma, estabilidad y escalabilidad. (Gómez, 2009)

Es un producto muy vendido a nivel mundial pero presenta como dificultad, que por su alto precio, sólo empresas muy grandes y multinacionales puedan adquirirlo, razón por la cual no fue tomado en consideración para darle solución al problema planteado.

#### **MySQL**

MySQL constituye uno de los SGBD más robustos que existe en la actualidad, muy confiable, seguro y de gran rapidez, posee gran cantidad de librerías que facilitan su instalación y configuración, estas características lo hacen uno de los sistemas de mayor aceptación y altamente conveniente para la gestión de datos en grandes y pequeños volúmenes. (autores, 2011)

Su principal dificultad es que es propietario y está patrocinado por una empresa privada que posee el copyright de la mayor parte del código, razón por lo cual no es considerado como parte de la solución para el problema planteado.

## **PostgreSQL**

PostgreSQL es un magnífico gestor de bases de datos, “*considerado como la base de datos de código abierto más avanzada del mundo*” (John, 2010)

Mantiene la misma velocidad de respuesta para gestionar tanto bases de datos relativamente pequeñas como bases de datos de considerables dimensiones. Presenta una gran escalabilidad que le permite su uso en sitios web que posean numerosas peticiones diarias. Además es un gestor altamente extensible, que incluye la herencia entre tablas y posee una API flexible. (Pecos, 2010)

Este gestor será utilizado como parte del desarrollo para solucionar el problema planteado, pues aunque MySQL brinda mayor rapidez, PostgreSQL es mucho más potente y además es el gestor con el cual el equipo de desarrollo ha estado trabajando y sobre el cual se posee la experiencia necesaria.

### **1.9.3 Base de datos de XML nativo**

#### **EXist-db**

EXist-db es un sistema de gestión de base de datos de código abierto, construido utilizando la tecnología XML. Permite almacenar datos XML y presenta funciones de procesamiento eficiente. Es extensible y altamente compatible con el estándar XQuery, pues contiene una gran colección de módulos de función de XQuery, que proporcionan un potente entorno y mejor desempeño para desarrollar aplicaciones web basadas en XQuery. (exist, 2009)

EXist-db será utilizado para el trabajo con los XML como parte de la solución al problema planteado.

### **1.9.4 Frameworks**

Los frameworks constituyen una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Están compuestos por componentes personalizables e intercambiables para el desarrollo de aplicaciones, es decir, son una aplicación genérica incompleta y configurable a la cual podemos añadirle las últimas piezas para obtener una aplicación concreta.

Posibilitan la reutilización de código aprovechando funcionalidades ya implementadas, lo cual permite agilizar y disminuir el esfuerzo durante el proceso de desarrollo del software, logrando una concentración directa en la solución del problema. (Web, 2010)

### **Akelos**

Akelos es un framework de código abierto que permite aumentar la velocidad para desarrollar aplicaciones web y escribiendo menos código que otros frameworks de desarrollo. Es muy simple y requiere de un mínimo de configuración. Las aplicaciones desarrolladas con este framework siguen el patrón Modelo Vista Controlador (MVC). (Martínez, 2010)

### **CodeIgniter**

CodeIgniter es un framework altamente extensible y más rápido que muchos otros entornos, que posee una amplia documentación y librerías lo cual es indispensable para el desarrollo de aplicaciones web. Presenta una manera específica de codificar las páginas web y clasificar sus diferentes scripts, permitiendo que el código esté organizado y sea más fácil de crear y mantener. Utiliza además el patrón MVC. (Yepes, 2010)

### **Symfony**

Symfony es uno de los frameworks más populares entre los usuarios y empresas, permite a los programadores crear códigos con una elevada calidad y más fáciles de mantener. Es un framework maduro, estable y posee una excelente documentación, con varios libros y tutoriales distribuidos en Internet. Además emplea el patrón Modelo Vista Controlador, separando las partes que forman una aplicación: la lógica de negocio, la lógica de servidor y la presentación. También ofrece varias herramientas y clases que simplifican el desarrollo de aplicaciones web. Permite economizar el tiempo de desarrollo y centrarse en los aspectos más relevantes y específicos de cada aplicación. (Web M. d., 2010)

El sistema, al cual contribuirá la solución del problema planteado, está desarrollado sobre este framework, razón por la cual se selecciona Symfony para el desarrollo del módulo a implementar, manteniendo la línea de arquitectura seguida en el proyecto CRODA. Además se selecciona este framework, pues ofrece gran soporte y se actualiza constantemente. También se tiene un amplio conocimiento y dominio del framework por parte del equipo de desarrollo.

### **ExtJs**

ExtJs es una completa librería de JavaScript, con un excelente rendimiento que permite el desarrollo de aplicaciones web complejas y dinámicas empleando tecnologías como Ajax, DHTML, DOM. Permite realizar interfaces amigables para los usuarios, pues nos proporciona componentes con funcionalidades avanzadas de gran utilidad en la implementación. Además provee una experiencia consistente sobre cualquier navegador y evita tener que validar el código para que funcione bien en cada uno de los navegadores. Presenta una balanceada relación entre cliente-servidor. (Autores, 2007)

Aunque existen otras librerías referidas a la presentación de las aplicaciones como jQuery y Prototype, se decide el uso de ExtJs por las características y ventajas antes mencionadas, además se posee un conocimiento previo sobre esta librería pues ha sido utilizada en CRODA para el desarrollo de la apariencia de interfaz de usuario.

## **1.10 Estudio de las herramientas**

### **1.10.1 Herramientas CASE**

#### **Rational Rose**

*“Es el producto más completo de la familia Rational Rose. Al igual que todos los productos Rational Rose incluye soporte UML. Rational Rose Enterprise es la mejor elección para el ambiente de modelado que soporte la generación de código a partir de modelos en Ada, ANSI C++, C++, CORBA, Java/J2EE, Visual C++ y Visual Basic. Como todos los demás productos Rational Rose, proporciona un lenguaje común de modelado para el equipo que facilita la creación de software de calidad más rápidamente.”*  
(autores., 2009)

Es una herramienta software que posee diferentes funcionalidades que proveen el modelado visual mediante UML. Permite especificar, analizar y diseñar el sistema de software antes de codificarlo, así como mantener la consistencia de los modelos del sistema software.

Una característica importante es que permite la generación de código a partir de modelos y viceversa, posibilitando así tanto la ingeniería directa como la inversa. (Blanco, 2009)

#### **Visual Paradigm (VP)**

Visual Paradigm es una herramienta CASE de diseño, creada para ayudar en el desarrollo de software. VP-UML soporta los principales estándares de la industria, tales como Lenguaje de Modelado Unificado (UML), SysML, BPMN, XML, etc.

*“Visual Paradigm permite realizar ingeniería tanto directa como inversa, pues a partir de un modelo relacional es capaz de desplegar todas las clases asociadas a las tablas. Además, esta herramienta es colaborativa, soporta múltiples usuarios trabajando sobre el mismo proyecto. Es multiplataforma y permite el control de versiones.” (León, 2007)*

Se selecciona VP como herramienta CASE pues es la que se utiliza en las primeras versiones de la herramienta de autor CRODA, además aporta ventajas como: utiliza UML como lenguaje de modelado y permite la modelación de procesos de desarrollo de software que sigan la filosofía de software libre.

### **1.10.2 Entorno de desarrollo integrado (IDE)**

#### **ZendStudio**

ZendStudio es un IDE que se ha diseñado para maximizar la productividad de los desarrolladores, pues permite desarrollar y mantener el código más rápido, proporciona depuración completa y pruebas de apoyo, lo cual posibilita analizar y resolver los problemas de aplicación en un corto período de tiempo.

Permite el desarrollo ágil de aplicaciones, combinando características como la refactorización, la generación de código, el código de ayuda y análisis semántico. (autores, 2010)

#### **NetBeans IDE**

NetBeans IDE es un entorno de desarrollo. Una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso. (NetBeans, 2010)

Estas razones justifican que se haya seleccionado como IDE de programación, además presenta una excelente integración con Symfony y Subversion, con un completamiento de código superior a otros IDE, que le permite economizar tiempo a los desarrolladores. También aporta mejoras en cuanto a rendimiento, respecto a otros IDE como el ZendStudio.

### **1.10.3 Gestores de administración de Postgres**

#### **phpPgAdmin**

phpPgAdmin es una herramienta de administración, ideal para administradores de bases de datos PostgreSQL y servicios de hosting.

Entre sus principales características resaltan el soporte a múltiples versiones de PostgreSQL, administración de usuarios, grupos, bases de datos y esquemas, además ofrece la posibilidad de administrar varios servidores, manipular y exportar los datos a diferentes formatos e importar sentencias SQL. (autores, 2011)

### **Navicat**

Navicat es un magnífico cliente para la administración y control, de forma más sencilla y visual, de bases de datos PostgreSQL. Es multiplataforma, posibilita manejar tablas, contenidos y crear consultas, además permite crear copias de seguridad junto con otras funciones que harán más cómodo el trabajo diario con las bases de datos. (Balboa, 2009)

### **PgAdmin3**

PgAdmin3 es una herramienta muy útil y de código abierto para la administración de bases de datos PostgreSQL. Su interfaz gráfica permite que sea mucho más fácil la administración y está disponible para varios sistemas operativos incluyendo Linux. (Usuarios, 2010)

PgAdmin3 incorpora funcionalidades para realizar consultas, examinar su ejecución y trabajar con los datos. Permite realizar acciones, que vienen condicionadas con los permisos que tenga el usuario con que se conecten a la base de datos. (DataPrix, 2007)

Por sus funcionalidades y características se selecciona esta herramienta, para el trabajo con PostgreSQL en el desarrollo del módulo propuesto, además es una herramienta sobre la que se tiene experiencia, pues ya se ha trabajado con ella en el desarrollo de la herramienta de autor CRODA.

## **1.10.4 Clientes para control de versiones**

### **RapidSVN**

RapidSVN es un cliente gráfico para Subversion fácil de usar. Funciona en diferentes plataformas y dispone de un completo manual en línea. Permite el acceso a direcciones SVN, brinda funcionalidades como subir, descargar contenido y sincronizarlo con el servidor original, además posibilita comprobar su estado, crear y fusionar direcciones. (López, 2011)

## **KDESVN**

KDESVN es un cliente gráfico para el sistema de control de versiones Subversion. No presenta dificultades para su instalación y la misma se puede realizar desde los repositorios de diferentes distribuciones de Linux. Posee varias funcionalidades que facilitan la interacción con los repositorios, permitiendo descargar la versión más actualizada de datos, aplicaciones, etc., así como realizar la operación inversa desde una copia local, actualizando cambios realizados.

Posee gran similitud en términos de características funcionales, respecto a otros clientes, pero se selecciona para su uso pues en la actualidad KDESVN es la herramienta utilizada en el proyecto CRODA para el control de versiones, ofreciendo una mayor seguridad al evitar la pérdida de datos.

## **Conclusiones**

Como resultado de la investigación y del análisis bibliográfico realizado, durante el presente capítulo, se verificó que el estándar SCORM 2004 ofrece las indicaciones necesarias para implementar un módulo en la herramienta de autor CRODA que facilite la asignación de secuencia y navegación en los Objetos de Aprendizaje. Además se eligieron los lenguajes de programación, metodología de desarrollo de software, servidor web, gestor de base de datos, base datos de XML nativo, frameworks, así como las herramientas a utilizar para implementar dicho módulo.

## **Capítulo 2: Características del sistema**

### **Introducción**

En el presente capítulo se describe la propuesta de solución. Se explican conceptos relacionados con el Modelo de Dominio. Además se listan los requerimientos tanto funcionales como no funcionales del módulo propuesto. Se identifican los actores, casos de uso y las relaciones que se establecen entre ellos a través del Modelo de Caso de Uso del Sistema. También se presentan las correspondientes descripciones de los casos de uso identificados.

### **2.1 Objeto de Automatización**

Con el propósito de cumplir los objetivos anteriormente planteados se incluye en la herramienta de autor CRODA, un módulo que permitirá automatizar el proceso de asignar secuencia y navegación en los OA. Dicha automatización seguirá las especificaciones del estándar SCORM 2004. Para asignar y editar secuencia en un elemento se mostrará un área en la cual podrán gestionarse reglas de precondiciones, poscondiciones, condiciones de salida, reglas de acumulación, modo control y objetivos. Además se incluye la opción exportar un Objeto de Aprendizaje en la versión 1.2 y 1.3 del estándar SCORM, según las necesidades del profesor.

### **2.2 Descripción del sistema propuesto**

El módulo propuesto constituirá parte de una aplicación web desarrollada en PHP en una versión mayor que la 5.0, empleando como framework a Symfony y ExtJs como librería de JavaScript. Se hará uso del gestor de base de datos PostgreSQL y de EXist-db como base de datos de XML nativo.

Asignar secuencia se presentará como una opción en la creación de plantillas o generación de OA. Una vez seleccionada esta opción se mostrarán interfaces para la gestión de precondiciones, poscondiciones, condiciones de salida, reglas de acumulación, modo control y objetivos. De esta forma el diseñador instruccional podrá asignar la secuencia en el Objeto de Aprendizaje de acuerdo a sus necesidades, siguiendo objetivos pedagógicos específicos. El profesor podrá crear la cantidad de reglas necesarias para establecer diferentes rutas dentro de los OA. Estas reglas serán definidas estableciendo puntuaciones o condiciones. Una actividad será aprobada si cumple con todas las reglas definidas anteriormente por el profesor.

Dicho módulo añade características sobre la secuencia y navegación que están definidas en la documentación oficial de SCORM 2004 y que no se tienen en cuenta en las herramientas de autor estudiadas en el capítulo anterior. Básicamente se agregan otros controles de secuencia que permiten establecer un límite de intentos de una actividad y un tiempo límite para cada intento, además se incorporan otras consideraciones.

### 2.2.1 Modelo de Dominio

En el módulo propuesto no se realiza Modelo de Negocio pues en la herramienta de autor CRODA no se tiene en cuenta debido al bajo nivel de estructuración que presenta el negocio, sin embargo se realiza Modelo de Dominio representando visualmente los conceptos significativos para el tema a desarrollar.

El Modelo de Dominio es una representación visual y estática del entorno real, objeto del proyecto, ayuda a comprender los conceptos que utilizan los usuarios, los conceptos con los que trabajan y con los que deberá trabajar la aplicación. (Larman, 2008)

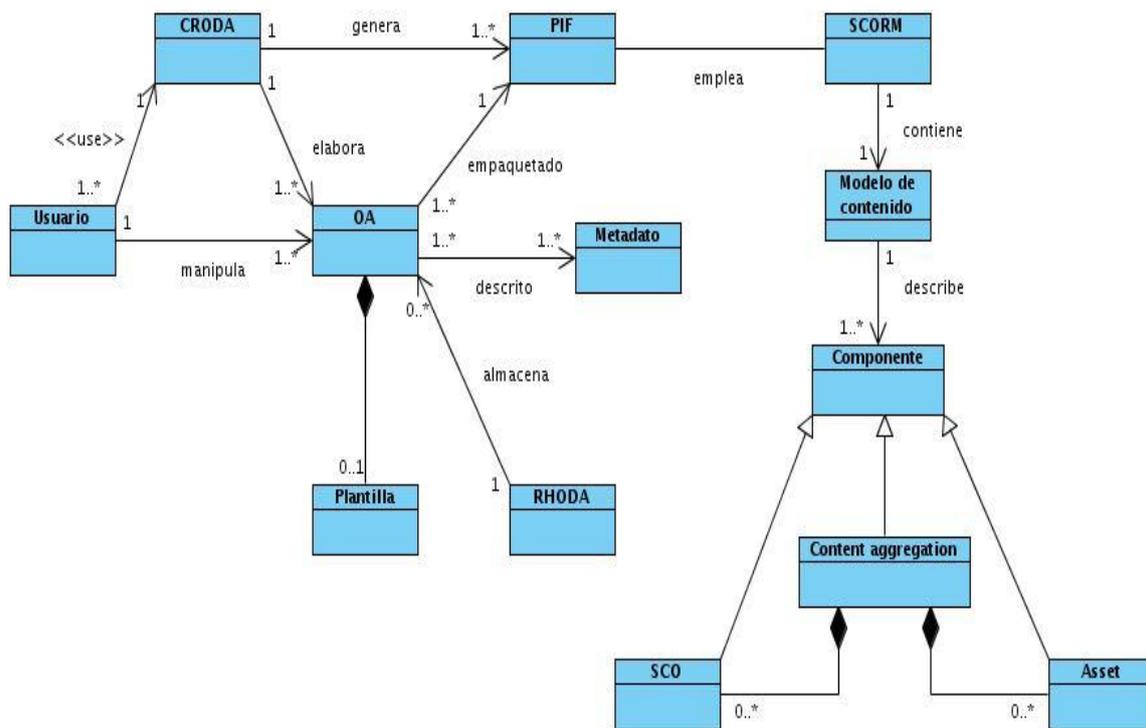


Figura 3: Modelo de Dominio.

## 2.3 Requerimientos

### 2.3.1 Requerimientos funcionales

RF1. Editar acción de “Precondiciones”.

- RF2. Editar condiciones de “Precondiciones”.
- RF3. Introducir ID del objetivo referenciado de las “Precondiciones”.
- RF4. Introducir puntuación a comparar de las “Precondiciones”.
- RF5. Editar acción de “Poscondiciones”.
- RF6. Editar condiciones de “Poscondiciones”.
- RF7. Introducir Id del objetivo referenciado para “Poscondiciones”.
- RF8. Introducir puntuación a comparar de “Poscondiciones”.
- RF9. Editar condición para “Condiciones de salida”.
- RF10. Editar condiciones de “Condiciones de salida”.
- RF11. Introducir Id del objetivo referenciado para “Condiciones de salida”.
- RF12. Introducir puntuación a comparar para “Condiciones de salida”.
- RF13. Editar estado de “Reglas de acumulación”.
- RF14. Editar condición para el estado seleccionado en “Reglas de acumulación”.
- RF15. Editar condiciones en “Reglas de acumulación”.
- RF16. Editar “Controles de secuencia”.
- RF17. Establecer límites de control.
- RF18. Establecer controles auxiliares.
- RF19. Crear regla.
- RF20. Actualizar regla.
- RF21. Eliminar regla.
- RF22. Cerrar ventana actual.
- RF23. Editar objetivo.
- RF24. Añadir objetivo.

- RF25. Adicionar mapeo de objetivo.
- RF26. Eliminar objetivo.
- RF27. Crear plantilla.
- RF28. Listar plantillas.
- RF29. Generar OA.
- RF30. Exportar Objeto de Aprendizaje.
- RF31. Asignar secuencia.
- RF32. Editar secuencia.
- RF33. Eliminar secuencia.

### **2.3.2. Requerimientos no funcionales**

#### **Usabilidad**

RNF 1. El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente Web en sentido general.

RNF 2. En la interacción con el software el sistema mostrará mensajes que sean de entendimiento común con el usuario, para de esta forma elevar la comunicación con el mismo.

RNF 3. En los campos que sean complejos se mostrarán notas indicando ejemplos concretos para guiar el proceso.

#### **Accesibilidad**

RNF 4. Las interfaces deben tener vínculos entre ellas para facilitar el proceso.

RNF 5. Protección contra acciones no autorizadas que puedan afectar la integridad de los datos.

#### **Fiabilidad**

- **Disponibilidad**

RNF 6. Una vez publicado el sistema, deberá estar disponible para los usuarios en cualquier momento.

### **Eficiencia**

RNF 7. Tiempos de respuestas rápidas como máximo 10 segundos al igual que la velocidad de procesamiento de la información.

RNF 8. El sistema funcionará de manera óptima en los navegadores Web Mozilla Firefox, Internet Explorer 6 o superior y Opera.

- **Hardware**

RNF 9. La PC del servidor debe tener al menos 1GB de RAM.

RNF 10. La PC del servidor debe ser Pentium 4 o superior.

RNF 11. La capacidad de almacenamiento del disco duro del servidor debe ser como mínimo 120GB.

RNF 12. La PC cliente debe tener como mínimo 80GB de capacidad de almacenamiento.

RNF 13. La PC cliente debe tener al menos 1GB de RAM.

RNF 14. La PC cliente debe ser Pentium 4 o superior.

- **Software**

RNF 15. Se utilizará PostgreSQL versión 8.4 como gestor de base de datos.

RNF 16. Se utilizará Exist-db versión 1.4 como base de datos de XML nativo.

### **Restricciones del diseño**

- **Herramienta CASE de modelado visual.**

RNF 17. Para el modelado visual se deberá usar Visual Paradigm 6.4 como herramienta CASE.

- **Lenguaje del lado del servidor**

RNF 18. Se utilizará PHP 5.

- **Entorno de desarrollo integrado IDE**

RNF 19. Se hará uso de Netbeans 6.9 como entorno de desarrollo IDE.

### **Interfaz**

- **Interfaces de usuario**

RNF 20. Identificación de colores y formatos con la herramienta de autor CRODA.

- **Interfaces de comunicación**

RNF 21. Las etiquetas de los campos deben presentarse de manera normalizada en todas las páginas, para asegurar que estas aparezcan con la misma tipografía.

RNF 22. Los campos obligatorios serán marcados por un indicador preferentemente de color rojo.

## **2.4 Modelo de Caso de Uso del sistema**

### **2.4.1. Definición de actores del sistema**

<b>Actor</b>	<b>Descripción</b>
Creador	Es la persona que realiza todas las funciones relacionadas con la creación y edición de plantillas (incluidas las de secuenciación).
Generador	Es la persona que realiza todas las funciones relacionadas con la generación de OA (incluidas las de secuenciación).
Generador-Creador	Generalización de los usuarios Generador y Creador.

### **2.4.2. Definición de casos de uso**

<b>CU # 1</b>	<b>Asignar secuencia.</b>
Actor	Generador/Creador
Descripción	Permite al Generador/Creador asignar secuencia a un elemento del árbol.
Referencia	RF31.

<b>CU # 2</b>	<b>Editar secuencia (extend).</b>
---------------	-----------------------------------

Actor	Generador/Creador.
Descripción	Permite al Generador/Creador editar la secuencia del elemento del árbol que se está editando.
Referencia	RF 32 y RF 33.

<b>CU # 3</b>	<b>Gestionar precondition. (extend)</b>
Actor	Generador/Creador
Descripción	Permite al Generador/Creador establecer reglas para las preconditiones de un elemento del árbol.
Referencia	RF 19, RF 20, RF 21, RF 22, RF 1, RF 2, RF 3 y RF 4.

<b>CU # 4</b>	<b>Gestionar poscondición. (extend)</b>
Actor	Generador/Creador
Descripción	Permite al Generador/Creador establecer reglas para las poscondiciones de un elemento del árbol.
Referencia	RF 19, RF 20, RF 21, RF 22, RF 5, RF 6, RF 7 y RF 8.

<b>CU # 5</b>	<b>Gestionar condición de salida. (extend)</b>
Actor	Generador/Creador
Descripción	Permite al Generador/Creador establecer reglas para las condiciones de salida de un elemento del árbol.
Referencia	RF 19, RF 20, RF 21, RF 22, RF 9, RF 10, RF 11 y RF 12.

<b>CU # 6</b>	<b>Gestionar regla de acumulación.</b>
---------------	--

	<b>(extend)</b>
Actor	Generador/Creador
Descripción	Permite al Generador/Creador establecer reglas para definir qué % de aprobado es necesario para aprobar un contenido.
Referencia	RF 19, RF 20, RF 21, RF 22, RF 13, RF 14 y RF 15.

<b>CU # 7</b>	<b>Gestionar modo control. (extend)</b>
Actor	Generador/Creador
Descripción	Permite al Generador/Creador establecer reglas para los controles de un elemento del árbol, estableciendo una serie de parámetros para regir el OA.
Referencia	RF 22, RF 16, RF 17 y RF 18.

<b>CU # 8</b>	<b>Gestionar objetivo. (extend)</b>
Actor	Generador/Creador
Descripción	Permite al Generador/Creador establecer un objetivo primario y varios objetivos secundarios para referenciar las reglas de un elemento del árbol.
Referencia	RF 22, RF 23, RF 24, RF 25 y RF 26.

<b>CU # 9</b>	<b>Exportar Objeto de Aprendizaje.</b>
Actor	Generador
Descripción	Permite exportar un Objeto de Aprendizaje en cualquiera de las versiones del estándar SCORM.
Referencia	RF 30.

**Casos de Uso que son redefinidos.**

<b>CU # 10</b>	<b>Crear plantilla.</b>
Actor	Creador
Descripción	Permite crear una nueva plantilla e

	incorporarle facilidades de secuenciación si el Creador lo desea.
Referencia	RF 27.

CU # 11	Listar plantillas.
Actor	Generador
Descripción	Muestra dos listas individuales de las platillas existentes en el sistema con las dos versiones del estándar SCORM.
Referencia	FR 28.

CU # 12	Generar OA.
Actor	Generador
Descripción	Permite generar un Objeto de Aprendizaje a partir de una plantilla previamente seleccionada y adaptarlo a sus necesidades.
Referencia	RF 29.

### 2.4.3. Diagrama de Casos de Uso del Sistema (DCUS)

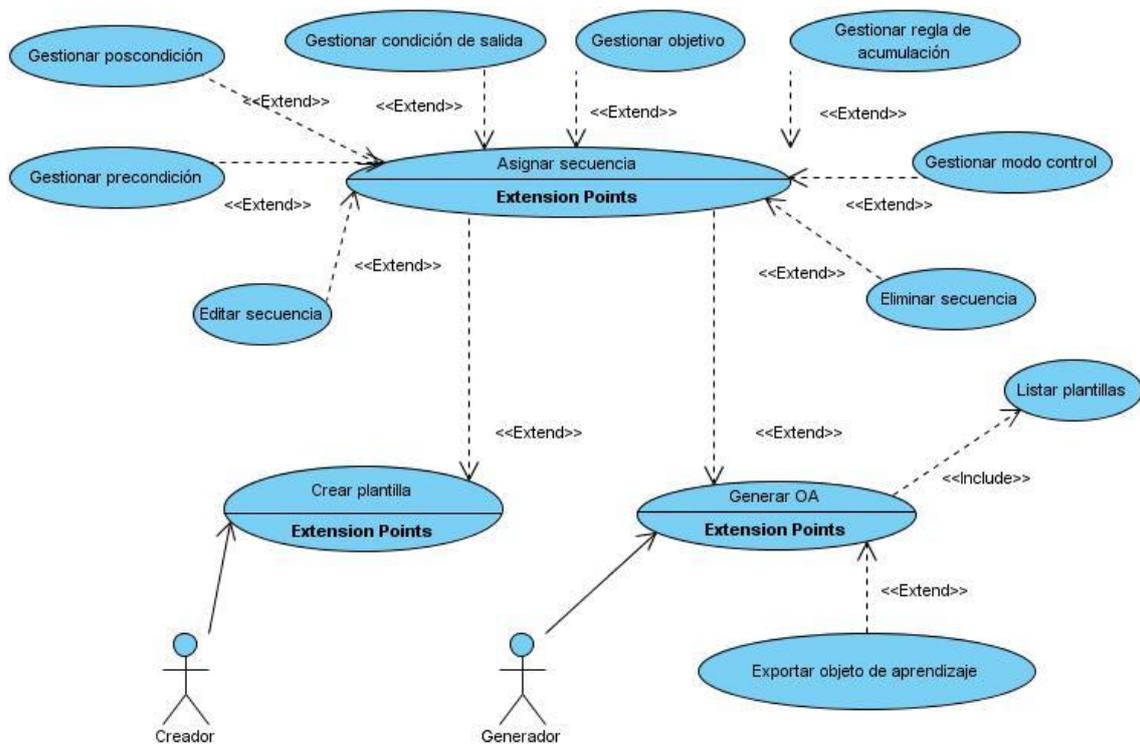


Figura 4: Diagrama de CUS módulo SCORM 2004.

## 2.5 Descripción textual de los casos de uso del sistema

CU Asignar secuencia.

Objetivo	Asignar secuencia a un elemento.	
Actores	Creador: Si se está Creando/Editando una plantilla. Generador: Si se está Creando/Editando un Objeto de Aprendizaje.	
Resumen	El caso de uso se inicia cuando el Generador/Creador está Creando/Editando una plantilla o Creando/Editando un OA y selecciona la opción Asignar secuencia sobre un Item o seleccionando la opción editar de un nodo de secuencia.	
Precondiciones	Si se está Creando/Editando una plantilla, el usuario debe estar registrado en el sistema y poseer privilegios de Creador. Si se está Creando/Editando un OA, el usuario debe estar registrado en el sistema y poseer privilegios de Generador.	
Poscondiciones	Se han creado o editado las reglas de secuencia del elemento.	
Referencia	RF33.	
Flujo de eventos		
Flujo básico: Asignar secuencia		
	Acción del actor	Respuesta del sistema
1.	El Generador/Creador selecciona la opción Asignar secuencia de un Item.	2. El sistema muestra la opciones: a) Precondiciones: Ver CU Gestionar precondición. b) Poscondiciones: Ver CU Gestionar poscondición. c) Condiciones de salida: Ver CU Gestionar condición de salida. d) Reglas de acumulación: Ver CU Gestionar regla de acumulación. e) Modo control: Ver CU Gestionar modo control. f) Objetivos: Ver CU Gestionar objetivo.
3.	El Generador/Creador realiza los cambios deseados en la secuenciación del elemento y selecciona la opción Cerrar.	4. El sistema guarda los cambios y crea el nodo secuencia para el elemento seleccionado, finalizando así el caso de uso.

CU Editar secuencia.

Objetivo	Editar secuencia.	
Actores	Creador: Si se está Creando/Editando una plantilla. Generador: Si se está Creando/Editando un Objeto de Aprendizaje.	
Resumen	El caso de uso se inicia cuando el Generador/Creador está Creando/Editando una plantilla o Creando/Editando un OA y selecciona la opción Editar secuencia sobre un nodo de secuencia.	
Precondiciones	Si se está Creando/Editando una plantilla, el usuario debe estar registrado en el sistema y poseer privilegios de Creador. Si se está Creando/Editando un OA, el usuario debe estar registrado en el sistema y poseer privilegios de Generador.	
Poscondiciones	Se han editado las reglas de secuencia del elemento.	
Referencia	RF32.	
Flujo de eventos		
Flujo básico: Editar secuencia		
	Acción del actor	Respuesta del sistema
1.	El Generador/Creador selecciona la opción Editar secuencia de un nodo secuencia.	2. El sistema muestra la opciones: a) Precondiciones: Ver CU Gestionar precondición. b) Poscondiciones: Ver CU Gestionar poscondición. c) Condiciones de salida: Ver CU Gestionar condición de salida. d) Reglas de acumulación: Ver CU Gestionar regla de acumulación. e) Modo control: Ver CU Gestionar modo control. f) Objetivos: Ver CU Gestionar objetivo.
3	El Generador/Creador realiza los cambios deseados en la secuenciación del elemento y selecciona la opción Cerrar.	4. El sistema guarda los cambios y muestra el mensaje "La secuencia ha sido editada con éxito". Finalizando así el caso de uso.
Flujo alternativo: Borrar secuencia.		
	Acción del actor	Respuesta del sistema

3.1 El Generador/Creador selecciona la opción Borrar de un nodo secuencia.	3.2 El sistema elimina las reglas y el nodo secuencia. Finalizando así el caso de uso.
--	--

CU Gestionar precondition.

Objetivo	Configurar las preconditiones de la secuenciación.
Actores	Creador: Si se está Creando/Editando una plantilla. Generador: Si se está Creando/Editando un Objeto de Aprendizaje.
Resumen	El caso de uso se inicia cuando el Generador/Creador está Creando/Editando una plantilla o Creando/Editando un OA y selecciona la opción Asignar secuencia sobre un Item o seleccionando la opción editar de un nodo secuencia y escogiendo la opción Gestionar preconditiones. El caso de uso finaliza cuando se ha realizado esta acción.
Precondiciones	Si se está Creando/Editando una plantilla, el usuario debe estar registrado en el sistema y poseer privilegios de Creador. Si se está Creando/Editando un OA, el usuario debe estar registrado en el sistema y poseer privilegios de Generador.
Poscondiciones	Se han creado o actualizado las preconditiones del nodo de secuenciación.
Referencia	RF 19, RF 20, RF 21, RF 22, RF 1, RF 2, RF 3 y RF 4.

Flujo de eventos

Flujo básico: Gestionar precondition

	Acción del actor	Respuesta del sistema
1.	El Generador/Creador selecciona la opción Asignar secuencia de un Item o selecciona la opción Editar en un nodo secuencia.	2. El sistema muestra la opciones: a) Precondiciones b) Poscondiciones: Ver CU Gestionar poscondición. c) Condiciones de salida: Ver CU Gestionar condición de salida. d) Reglas de acumulación: Ver CU Gestionar regla de acumulación. e) Modo control: Ver CU Gestionar modo control. f) Objetivos: Ver CU Gestionar

		objetivo.
3.	El Generador/Creador selecciona la opción Precondiciones (seleccionada predeterminadamente por el sistema).	4. El sistema muestra una interfaz para Editar las precondiciones: Y las opciones: Crear regla. Ver sección 1. Actualizar. Ver sección 2. Eliminar. Ver sección 3. Cerrar.
5.	El Generador/Creador realiza los cambios deseados y selecciona la opción Cerrar.	6. El sistema cierra la ventana guardando los cambios en la secuencia del elemento. Finaliza así el caso de uso.

Sección 1: "Crear regla"

Flujo básico: Gestionar Precondición

	Acción del actor	Respuesta del sistema
1.	<p>El Generador/Creador selecciona una de las acciones: (Campos de selección).</p> <p>a) Saltar: (Marcada predeterminadamente por el sistema)</p> <p>b) Deshabilitar</p> <p>c) Ocultar</p> <p>d) Parar recorrido</p> <p>Y la condición correspondiente:</p> <p>a) Todas: (Marcada predeterminadamente por el sistema)</p> <p>b) Cualquiera</p>	<p>2. El sistema muestra las siguientes condiciones con un campo de selección para que el Generador/Creador seleccione la que desee:</p> <ul style="list-style-type: none"> <li>- Se intentó</li> <li>- Estado del objetivo conocido.</li> <li>- Satisfecho.</li> <li>- Progreso de actividad conocido.</li> <li>- Completado.</li> <li>- Puntuación del objetivo conocido.</li> <li>- Puntuación del objetivo menor que.</li> <li>- Puntuación del objetivo mayor que.</li> <li>- Tiempo límite de intentos.</li> <li>- Verdadero siempre.</li> </ul> <p>El sistema muestra además los campos:</p> <ul style="list-style-type: none"> <li>- Id objetivo referenciado (Formato: alfabético y numérico)</li> </ul>

		<p>(Válido sólo para las condiciones Estado del objetivo conocido, Satisfecho, Puntuación del objetivo conocido, Puntuación del objetivo mayor que, Puntuación del objetivo menor que)</p> <ul style="list-style-type: none"> <li>- Puntuación a comparar (Formato: numérico, valor único: -1;1, cifras decimales:4) (Válido sólo para los campos: Puntuación del objetivo mayor que y Puntuación del objetivo menor que)</li> </ul>
3.	El Generador/Creador selecciona las condiciones para la regla, introduce los valores deseados en los campos y selecciona la opción Crear regla.	4. El sistema crea la regla según las condiciones seleccionadas y la muestra en un panel inferior.

Flujos alternos

Datos Incorrectos

	Acción del actor	Respuesta del sistema
2.1	El Generador/Creador introduce en el campo Puntuación a comparar valores distintos a los definidos. <ul style="list-style-type: none"> <li>- Puntuación a comparar (Formato: numérico, valor único: -1; 1, cifras decimales: hasta 4).</li> </ul>	2.2 El sistema muestra el mensaje "Datos incorrectos", en la parte inferior del campo indicando que este debe ser cambiado.
2.3	El Generador/Creador selecciona la opción Crear regla sin haber corregido los errores señalados.	2.4 El sistema muestra el mensaje "Presenta un campo con errores".

Sección 2: "Actualizar"

Flujo básico: Gestionar Precondición

	Acción del actor	Respuesta del sistema
1.	El Generador/Creador edita los valores deseados y selecciona la opción Actualizar indicando la regla	2. El sistema permite que el Generador/Creador agregue la(s) condiciones deseadas a la regla

	deseada.	seleccionada y una vez adicionadas las muestra en el panel inferior.
Flujo alternativo: Actualizar regla.		
	Acción del actor	Respuesta del sistema
1.1	El Generador/Creador presiona el botón Actualizar sin haber seleccionado ninguna regla.	1.2 El sistema muestra el mensaje “Seleccione una regla para actualizar”.
Sección 3: “Eliminar regla”		
Flujo básico: Gestionar precondition		
	Acción del actor	Respuesta del sistema
1.	El Generador/Creador selecciona la regla que se desea eliminar y escoge la opción Eliminar.	2. El sistema elimina la regla y actualiza la información del panel inferior.
Flujo alternativo: Eliminar regla.		
	Acción del actor	Respuesta del sistema
1.1	El Generador/Creador presiona el botón Eliminar sin haber seleccionado ninguna regla.	1.2 El sistema muestra el mensaje “Seleccione una regla para eliminar”.

Las restantes descripciones textuales de los casos de uso del sistema pueden ser consultados en Anexos (#1 al #6).

### **Conclusiones**

En el presente capítulo se inició el desarrollo de la propuesta de solución, analizando las principales características del sistema para asignar secuencia y navegación en los OA. Se definió en el levantamiento de los requerimientos funcionales y no funcionales, qué es lo que el sistema debe hacer y cómo debe hacerlo. También se identificaron los actores y casos de usos, así como las relaciones entre ellos, logrando una descripción detallada para cada caso de uso. Además fue necesaria la redefinición de algunos casos de uso, incorporándoles las especificaciones del estándar SCORM 2004. Los artefactos generados en este capítulo posibilitarán la entrada al flujo de trabajo Análisis y Diseño.

## *Capítulo 3: Análisis y Diseño del módulo*

### **Capítulo 3: Análisis y Diseño del módulo**

#### **Introducción**

Como un paso fundamental para la posterior implementación, en el presente capítulo se realizan los artefactos requeridos en el flujo de trabajo Análisis y Diseño para los casos de uso identificados anteriormente. Tanto el análisis como el diseño constituyen elementos básicos en el proceso de desarrollo de software pues guían al desarrollador, mostrándole mediante los artefactos del Modelo de Diseño un conjunto de elementos que serán posteriormente implementados.

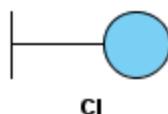
#### **3.1 Modelo de Análisis**

Teniendo en cuenta los requerimientos capturados en el capítulo anterior, en el Modelo de Análisis se realiza un esbozo preliminar del sistema a implementar. El principal objetivo del análisis es comprender perfectamente los requisitos del software y no precisar cómo se implementará la solución, por lo que en el mismo no se tienen en cuenta los lenguajes de programación, herramientas ni tecnologías a utilizar. Básicamente se enfoca en los requerimientos funcionales del sistema.

##### **3.1.1 Diagrama de Clases del Análisis**

El primer paso para la elaboración del Diagrama de Clases del Análisis es la identificación de las clases del análisis, las cuales no son más que la representación de los roles de los elementos del modelo. Las clases del análisis se dividen en tres estereotipos que se presentan a continuación:

**Clases Interfaz:** Se utilizan para modelar la interrelación entre el sistema y sus actores (es decir usuarios y sistemas externos). Las clases interfaz modelan las partes del sistema que dependen de sus actores lo que implica que clarifiquen y reúnan los requisitos en los límites del sistema.



**Figura 5: Estereotipo de las clases interfaz.**

**Clases Control:** Representan la coordinación, secuencia, transacciones y control de objetos, se usan con frecuencia para encapsular el control de un caso de uso en concreto.



Figura 6: Estereotipo de las clases control.

**Clases Entidad:** Se utilizan para modelar información que posee una vida larga y que es a menudo persistente. En la mayoría de los casos se derivan directamente de una clase entidad del negocio. (Jacobson, 2000)

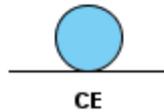


Figura 7: Estereotipo de las clases entidad.

A continuación se presentan los Diagramas de Clases del Análisis del módulo.



Figura 8: Diagrama de Clases del Análisis CU Asignar Secuencia.



Figura 9: Diagrama de Clases del Análisis CU Editar secuencia.



Figura 10: Diagrama de Clases del Análisis CU Gestionar precondition.

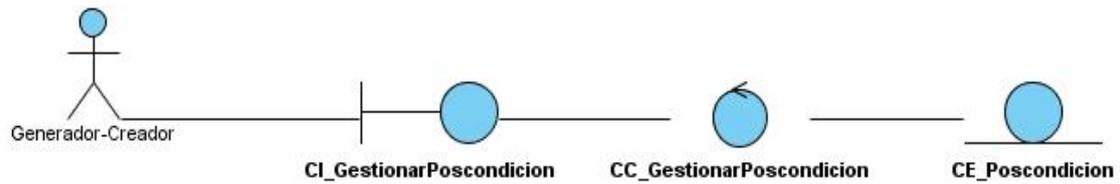


Figura 11: Diagrama de Clases del Análisis CU Gestionar poscondición.

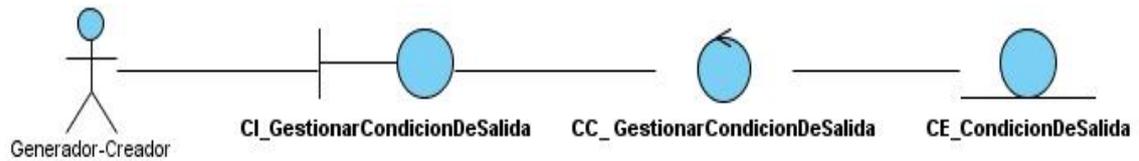


Figura 12: Diagrama de Clases del Análisis CU Gestionar condición de salida.

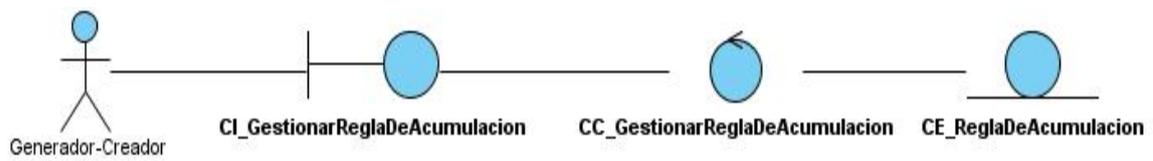


Figura 13: Diagrama de Clases del Análisis CU Gestionar regla de acumulación.

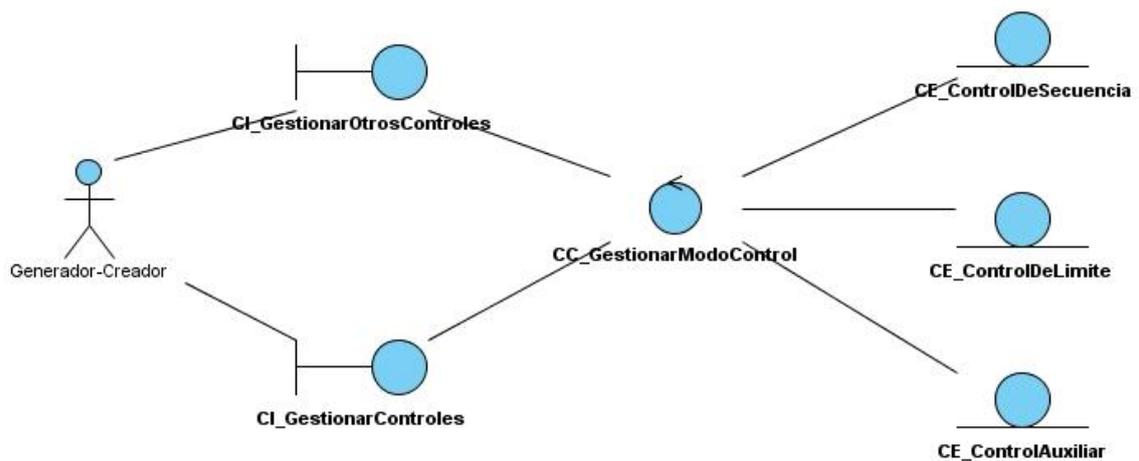


Figura 14: Diagrama de Clases del Análisis CU Gestionar modo control.

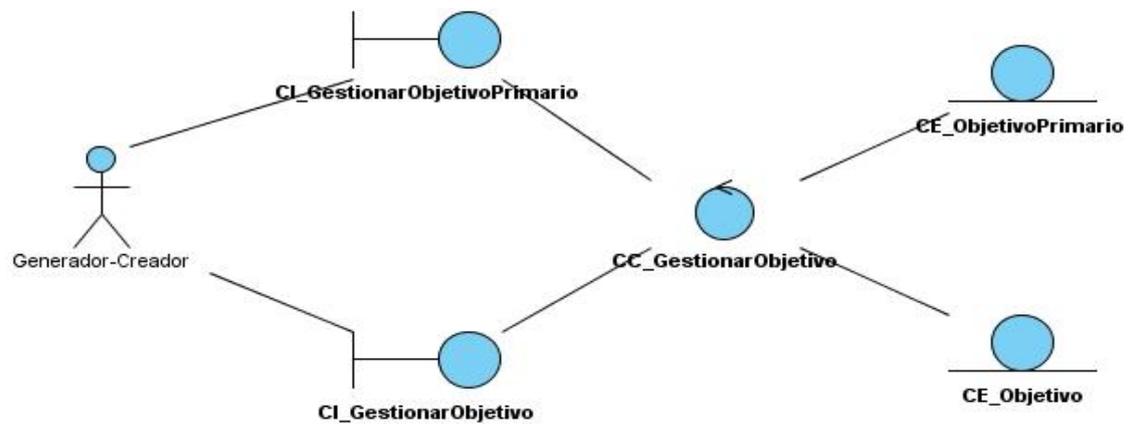


Figura 15: Diagrama de Clases del Análisis CU Gestionar objetivo.



Figura 16: Diagrama de Clases del Análisis CU Exportar Objeto de Aprendizaje.

### 3.1.2 Diagramas de Interacción

Los Diagramas de Interacción son modelos que describen la manera en que colaboran grupos de objetos para cierto comportamiento. Un Diagrama de Interacción capta el comportamiento de un solo caso de uso. El diagrama muestra cierto número de ejemplos de objetos y los mensajes que se pasan entre estos objetos dentro del caso de uso. (Alzate, 2009)

Existen dos tipos de Diagramas de Interacción: Diagramas de Secuencia y Diagramas de Colaboración. En el presente trabajo se realizarán diagramas de colaboración pues estos aportan una idea más clara de la dimensión estructural del sistema.

#### Diagramas de Colaboración

En los Diagramas de Colaboración, los objetos ejemplo se muestran como íconos. Las flechas indican, como en los diagramas de secuencia, los mensajes enviados dentro del caso de uso dado. En los Diagramas de Colaboración la secuencia se indica enumerando los mensajes.

A continuación se muestran los Diagramas de Colaboración del módulo SCORM 2004.

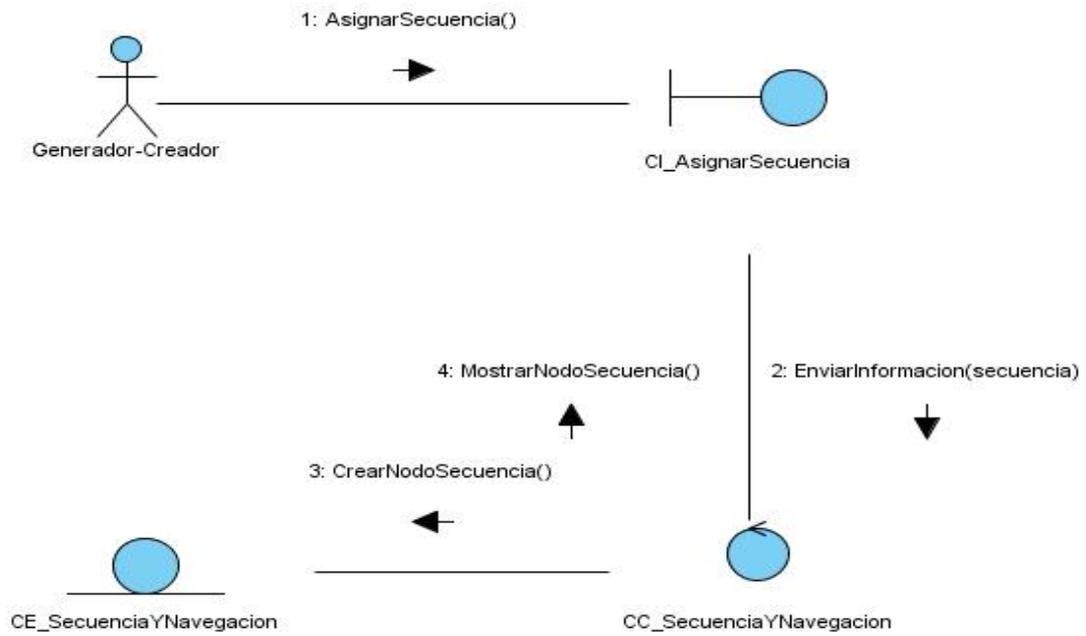


Figura 17: Diagrama de Colaboración CU Asignar Secuencia.

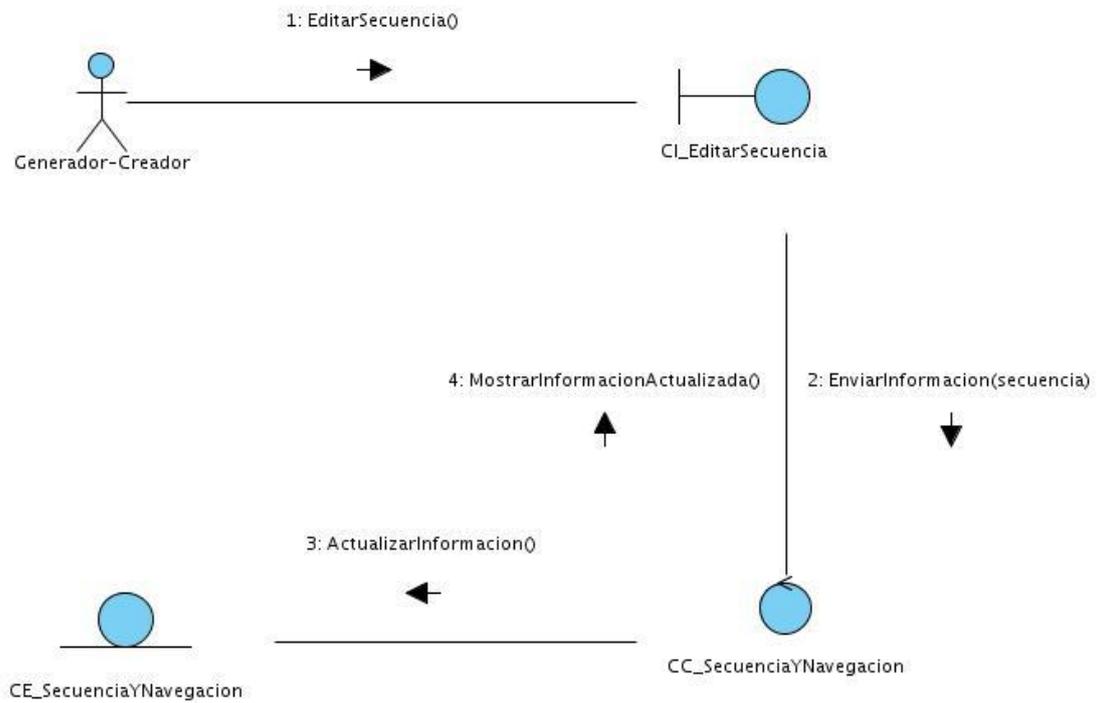


Figura 18: Diagrama de Colaboración CU Editar secuencia Sección Editar.

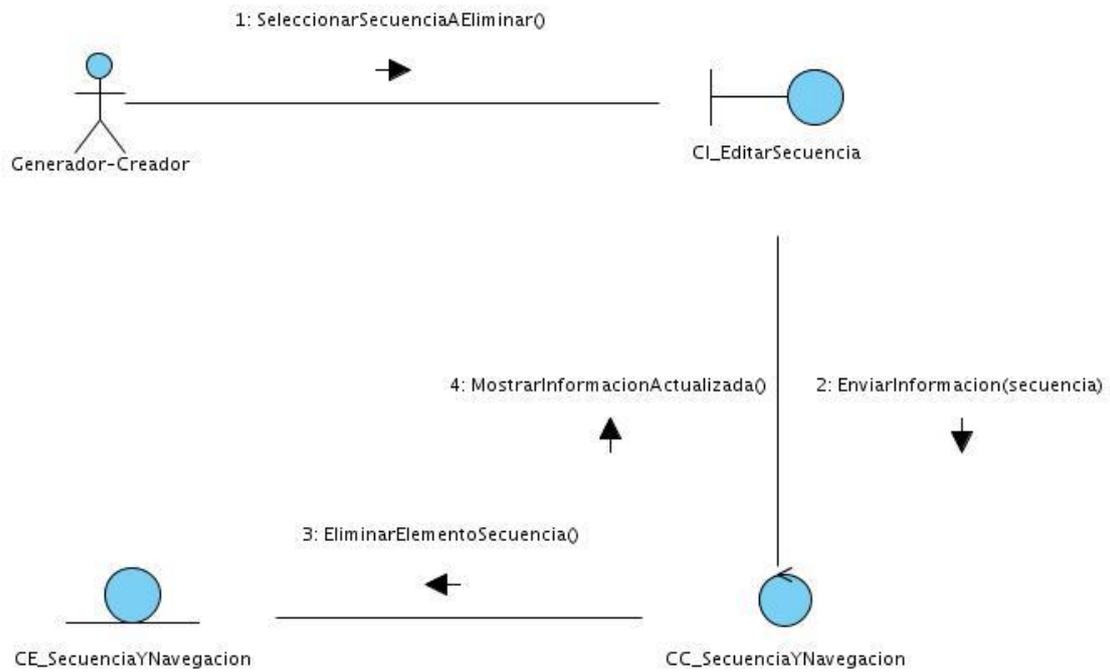


Figura 19: Diagrama de Colaboración CU Editar secuencia Sección Eliminar.

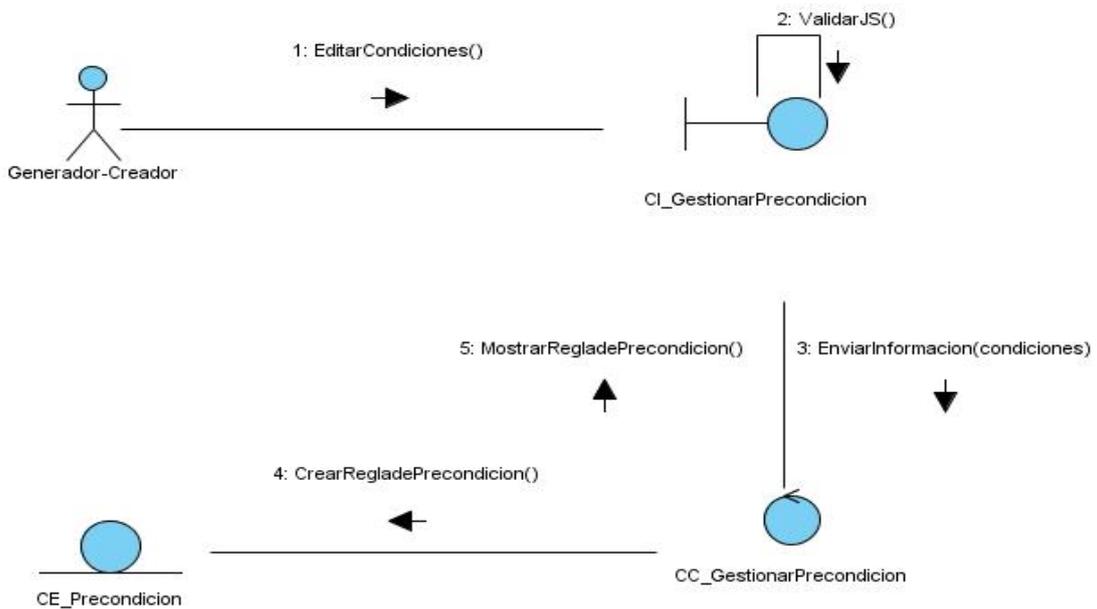
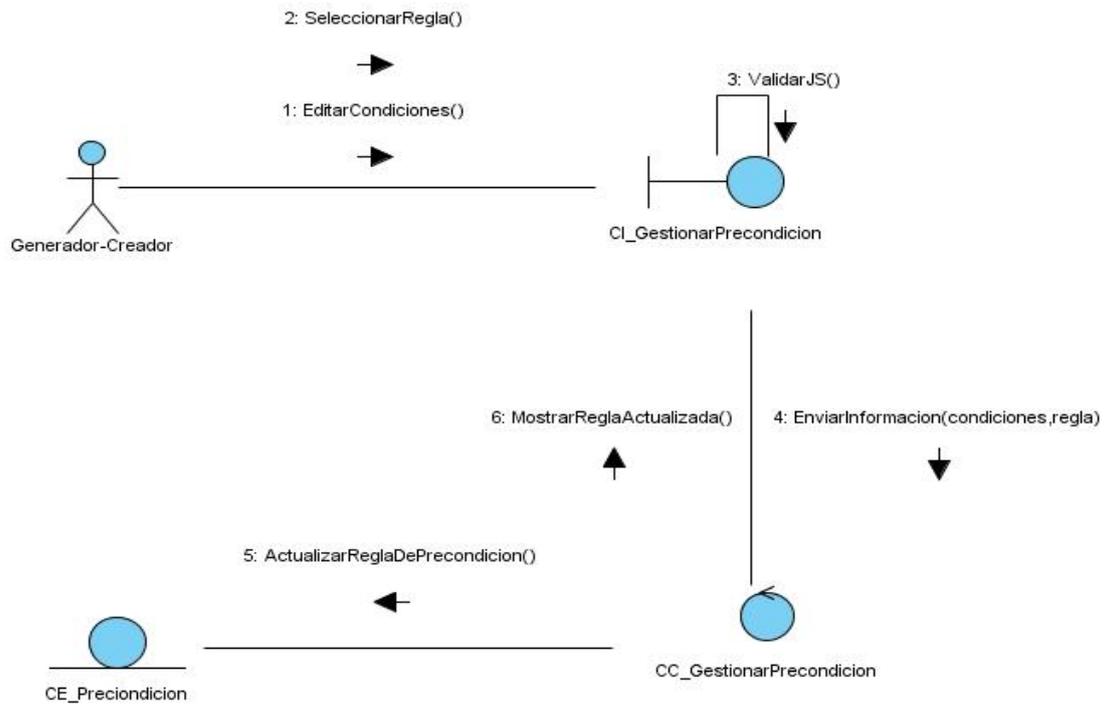
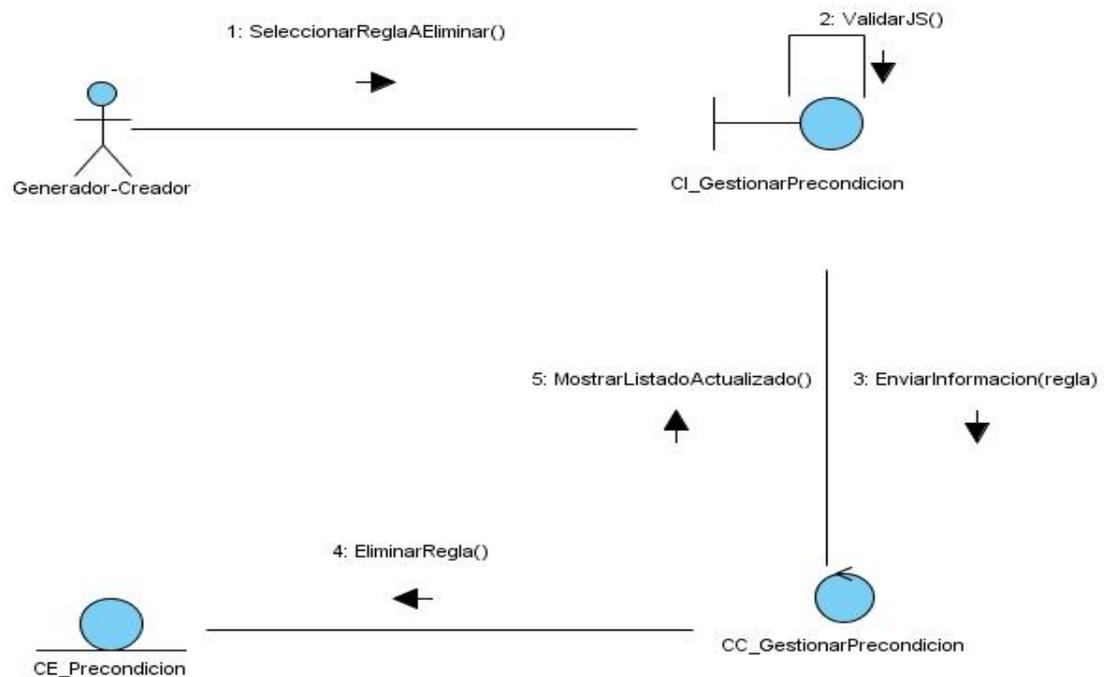


Figura 20: Diagrama de Colaboración CU Gestionar precondition Sección Crear regla.



**Figura 21: Diagrama de Colaboración CU Gestionar precondición Sección Actualizar regla.**



**Figura 22: Diagrama de Colaboración CU Gestionar precondición Sección Eliminar regla.**

Los restantes Diagramas de Colaboración del Análisis de los demás casos de uso se encuentran en Anexos (#7 al #25).

### **3.2 Modelo de Diseño**

Al igual que el análisis, el diseño es realizado en la fase de elaboración. En esta parte del flujo de trabajo se realiza una modelación del sistema para definir una arquitectura que soporte todos los requisitos, incluyendo los no funcionales. La entrada esencial para el Modelo de Diseño es el resultado del análisis. En resumen el diseño proporciona una entrada apropiada y un punto de partida para las actividades de implementación subsiguientes, capturando los requisitos, interfaces y clases.

#### **3.2.1 Diagrama de Clases del Diseño**

En el Modelo de Diseño se requiere confeccionar los Diagramas de Clases del Diseño, los cuales contienen clases y relaciones entre ellas. Seguidamente se representan los Diagramas de Clases del Diseño correspondiente a los casos de usos descritos con anterioridad.

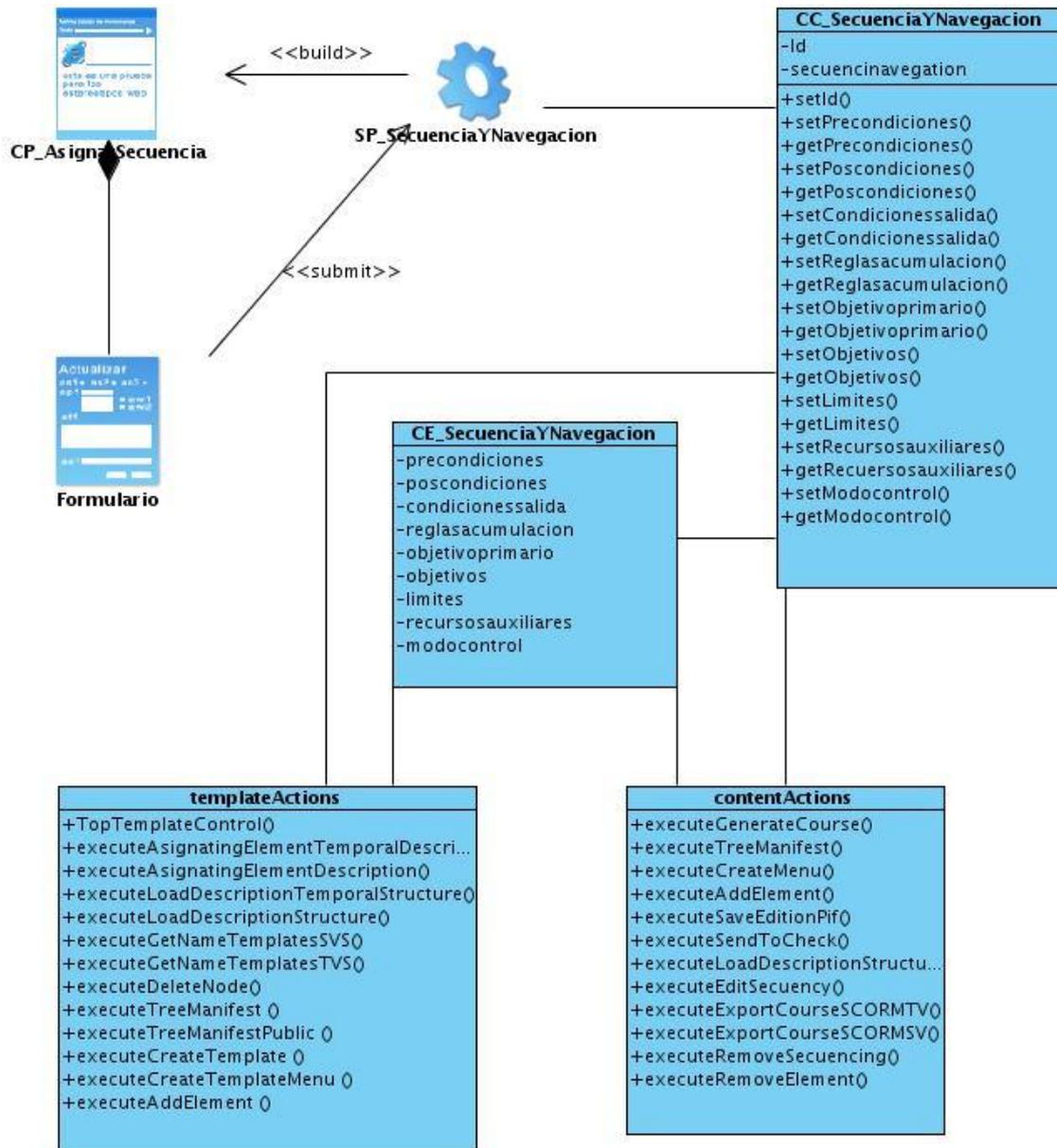


Figura 23: Diagrama de Clases del Diseño CU Asignar Secuencia.

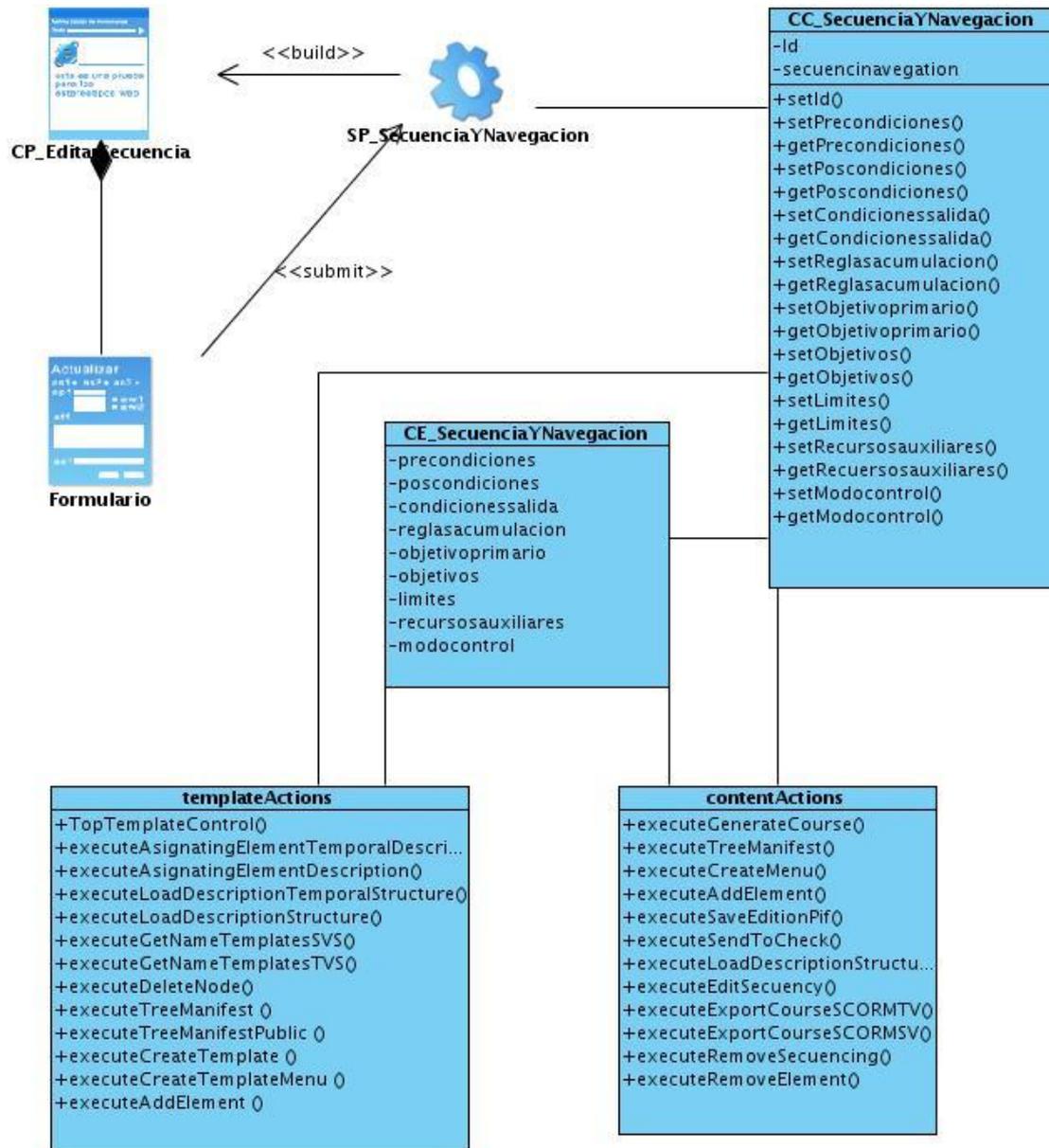


Figura 24: Diagrama de Clases del diseño CU Editar secuencia.

Los Diagramas de Clases de Diseño restantes, pueden ser consultados en Anexos (#26 al #32).

### 3.3 Diseño de la base de datos.

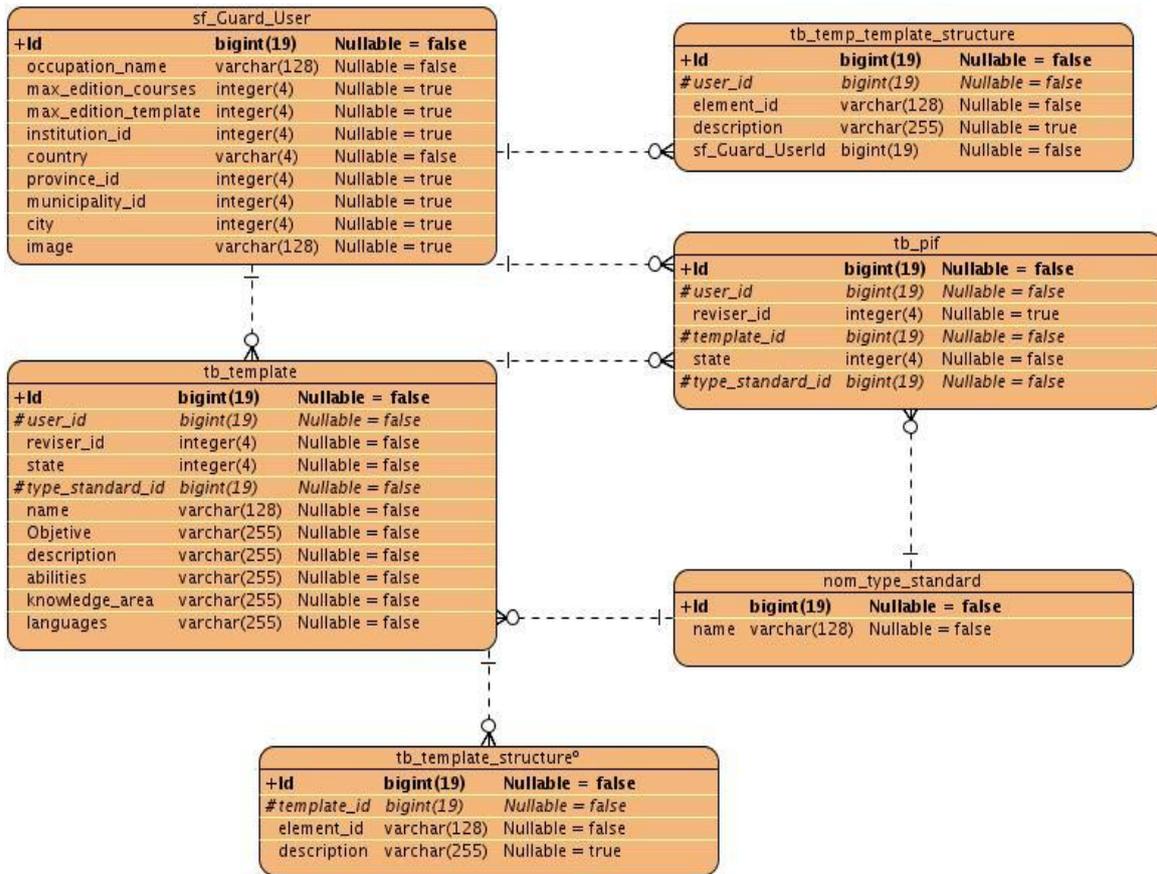


Figura 25: Diseño de la base de datos.

### 3.4 Descripción de las tablas de la base de datos.

Nombre:	sf_Guard_User.		
Descripción:	Tabla referente a los usuarios de la aplicación, la cual contiene los datos generales y específicos de los mismos, nombre de usuario, contraseña, nombre y apellidos, rol que ocupará en la aplicación ( creador, generador ), el tipo de acceso a la herramienta, a través de LDAP o manual así como identificador del propio usuario.		
Atributo:	Tipo:	Descripción:	
<ul style="list-style-type: none"> <li>▪ Id.</li> <li>▪ City.</li> <li>▪ occupation_name.</li> <li>▪ max_edition_courses.</li> <li>▪ max_edition_template.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Integer.</li> <li>▪ Varchar.</li> <li>▪ Varchar.</li> <li>▪ Integer.</li> <li>▪ Integer.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Identificador del usuario.</li> <li>▪ Ciudad de origen.</li> <li>▪ Identificador del rol.</li> </ul>	

<ul style="list-style-type: none"> <li>▪ institution_id.</li> <li>▪ Country.</li> <li>▪ province_id.</li> <li>▪ municipality_id.</li> <li>▪ Image.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Integer.</li> <li>▪ Integer.</li> <li>▪ Integer.</li> <li>▪ Integer.</li> <li>▪ Varchar.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Máximo de cursos en edición.</li> <li>▪ Máximo de plantillas en edición.</li> <li>▪ Identificador de la institución.</li> <li>▪ País.</li> <li>▪ Identificador de la provincia.</li> <li>▪ Identificador del municipio.</li> <li>▪ Imagen.</li> </ul>
---	--	--

Nombre:	tb_template.	
Descripción:	Tabla referente a las plantillas creadas en la aplicación, la cual contiene nombre, autor y descripción de la misma, condición, fecha de creada y el identificador de la plantilla.	
Atributo:	Tipo:	Descripción:
<ul style="list-style-type: none"> <li>▪ Id.</li> <li>▪ user_id.</li> <li>▪ reviser_id.</li> <li>▪ State.</li> <li>▪ type_standard_id.</li> <li>▪ Name.</li> <li>▪ Objective.</li> <li>▪ Description.</li> <li>▪ Abilities.</li> <li>▪ knowledge_area.</li> <li>▪ Languages.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Integer.</li> <li>▪ Integer.</li> <li>▪ Integer.</li> <li>▪ Integer.</li> <li>▪ Varchar.</li> <li>▪ Varchar.</li> <li>▪ Varchar.</li> <li>▪ Varchar.</li> <li>▪ Varchar.</li> <li>▪ Varchar.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Identificador.</li> <li>▪ Identificador del usuario.</li> <li>▪ Identificador del revisor.</li> <li>▪ Estado de la plantilla.</li> <li>▪ Identificador del tipo de estándar.</li> <li>▪ Nombre de la plantilla.</li> <li>▪ Objetivo.</li> <li>▪ Descripción.</li> <li>▪ Habilidades.</li> <li>▪ Área del conocimiento.</li> <li>▪ Idioma.</li> </ul>

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

	del OA, autor, fecha de creado, fecha de culminación, estado del OA, así como el identificador del mismo.	
Atributo:	Tipo:	Descripción:
<ul style="list-style-type: none"> <li>▪ Id.</li> <li>▪ user_id.</li> <li>▪ reviser_id.</li> <li>▪ template_id.</li> <li>▪ State.</li> <li>▪ type_standard_id.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Integer.</li> <li>▪ Integer.</li> <li>▪ Integer.</li> <li>▪ Integer.</li> <li>▪ Integer.</li> <li>▪ Integer.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Identificador.</li> <li>▪ Identificador del usuario.</li> <li>▪ Identificador del revisor.</li> <li>▪ Identificador de la plantilla.</li> <li>▪ Estado.</li> <li>▪ Identificador del tipo de estándar.</li> </ul>

Nombre:	nom_type_standard.	
Descripción:	Tabla referente al tipo de estándar utilizado en la plantilla, que contiene el nombre del mismo que puede ser SCORM 1.2 o SCORM 2004.	
Atributo:	Tipo:	Descripción:
<ul style="list-style-type: none"> <li>▪ Id.</li> <li>▪ Name.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Integer.</li> <li>▪ Varchar.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Identificador del estándar.</li> <li>▪ Nombre.</li> </ul>

Nombre:	tb_temp_template_structure.	
Descripción:	Tabla temporal referente a la descripción de los elementos que conforman la estructura de la plantilla, la cual contiene id del usuario, id del elemento, descripción.	
Atributo:	Tipo:	Descripción:
<ul style="list-style-type: none"> <li>▪ Id.</li> <li>▪ user_id.</li> <li>▪ element_id.</li> <li>▪ Description.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Integer.</li> <li>▪ Integer.</li> <li>▪ Varchar.</li> <li>▪ Varchar.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Identificador.</li> <li>▪ Identificador del usuario.</li> <li>▪ Identificador del elemento.</li> <li>▪ Descripción.</li> </ul>

Nombre:	tb_template_structure.		
Descripción:	Tabla referente a la descripción de los elementos que conforman la estructura de la plantilla, la cual contiene id de la plantilla, id del elemento, descripción.		
Atributo:	Tipo:	Descripción:	
<ul style="list-style-type: none"><li>▪ Id.</li><li>▪ template_id.</li><li>▪ element_id.</li><li>▪ Description.</li></ul>	<ul style="list-style-type: none"><li>▪ Integer.</li><li>▪ Integer.</li><li>▪ Varchar.</li><li>▪ Varchar.</li></ul>	<ul style="list-style-type: none"><li>▪ Identificador.</li><li>▪ Identificador de la plantilla.</li><li>▪ Identificador del elemento.</li><li>▪ Descripción.</li></ul>	

### **Conclusiones**

En este capítulo se realizaron los diagramas requeridos en el flujo de trabajo Análisis y Diseño de la metodología de desarrollo de software RUP, logrando una mejor comprensión de los requerimientos identificados y transformando los mismos en un diseño que indica cómo debe implementarse el módulo propuesto. Se elaboró el diseño de la base de datos y se describieron las tablas de la base de datos para brindar al desarrollador mayor información referente a las mismas.

## *Capítulo 4: Implementación del módulo*

### **Capítulo 4: Implementación del módulo**

#### **Introducción**

En el presente capítulo se realiza la implementación de la propuesta de solución, teniendo en cuenta los requerimientos capturados en capítulos anteriores, así como el pasado Análisis y Diseño del módulo. Se realizan además los Diagramas de Despliegue y de Componentes, para lograr una entrega final de la investigación desarrollada.

#### **4.1 Implementación**

El flujo de trabajo Implementación tiene como principal entrada los artefactos generados en el flujo de trabajo Análisis y Diseño. La implementación trata de adaptar el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares. El propósito fundamental de este flujo de trabajo es desarrollar la arquitectura y el sistema como un todo, logrando implementar las clases y subsistemas encontrados durante el diseño. (Jacobson, 2000)

#### **4.2 Diagrama de Despliegue**

El Diagrama de Despliegue ayuda a comprender la arquitectura de un sistema. Es la forma de mostrar la configuración de nodos de procesamientos en tiempo de ejecución y los componentes que en ellos residen. Estos nodos forman la topología de hardware sobre el que se ejecuta el sistema. Este diagrama se preocupa principalmente de la distribución, entrega e instalación de las partes que constituyen el sistema físico. (Cossio, 2009)

A continuación se muestra el Diagrama de Despliegue correspondiente al módulo propuesto.

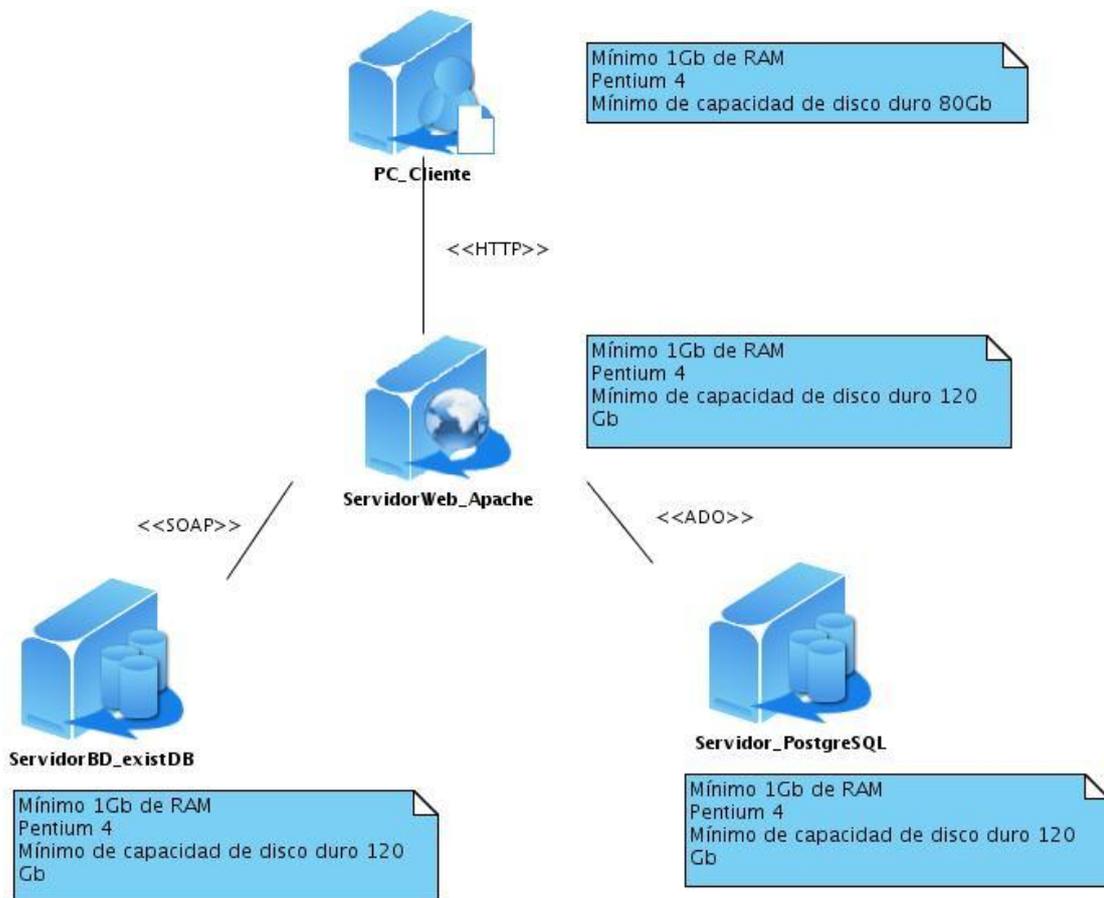


Figura 26: Diagrama de Despliegue Módulo SCORM 2004.

Para acceder al sistema el usuario necesita una PC Cliente que se conecta al servidor web Apache, donde se encuentran montadas las funcionalidades que se implementaron, esta conexión se realizará a través del protocolo HTTP. A su vez el servidor web intercambia información con un servidor de base de datos PostgreSQL mediante el protocolo de acceso a base de datos ADO. El servidor web también establece una conexión haciendo uso del protocolo SOAP con EXist-db, un servidor de base de datos de XML nativo donde se guardan las estructuras de las plantillas y de los OA.

### 4.3 Diagrama de Componentes

Los Diagramas de Componentes muestran las dependencias lógicas entre componentes de software, sean éstos componentes fuentes, binarios o ejecutables, ilustran las piezas del software, controladores embebidos, etc. En general, describen elementos físicos de un sistema así como sus relaciones. Prevalecen en el campo de la arquitectura de software pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema. (Ramírez, 2009)

A continuación se presenta el Diagrama de Componentes, modelando la vista de implementación estática del sistema desarrollado y mostrando los subsistemas de implementación por capas.

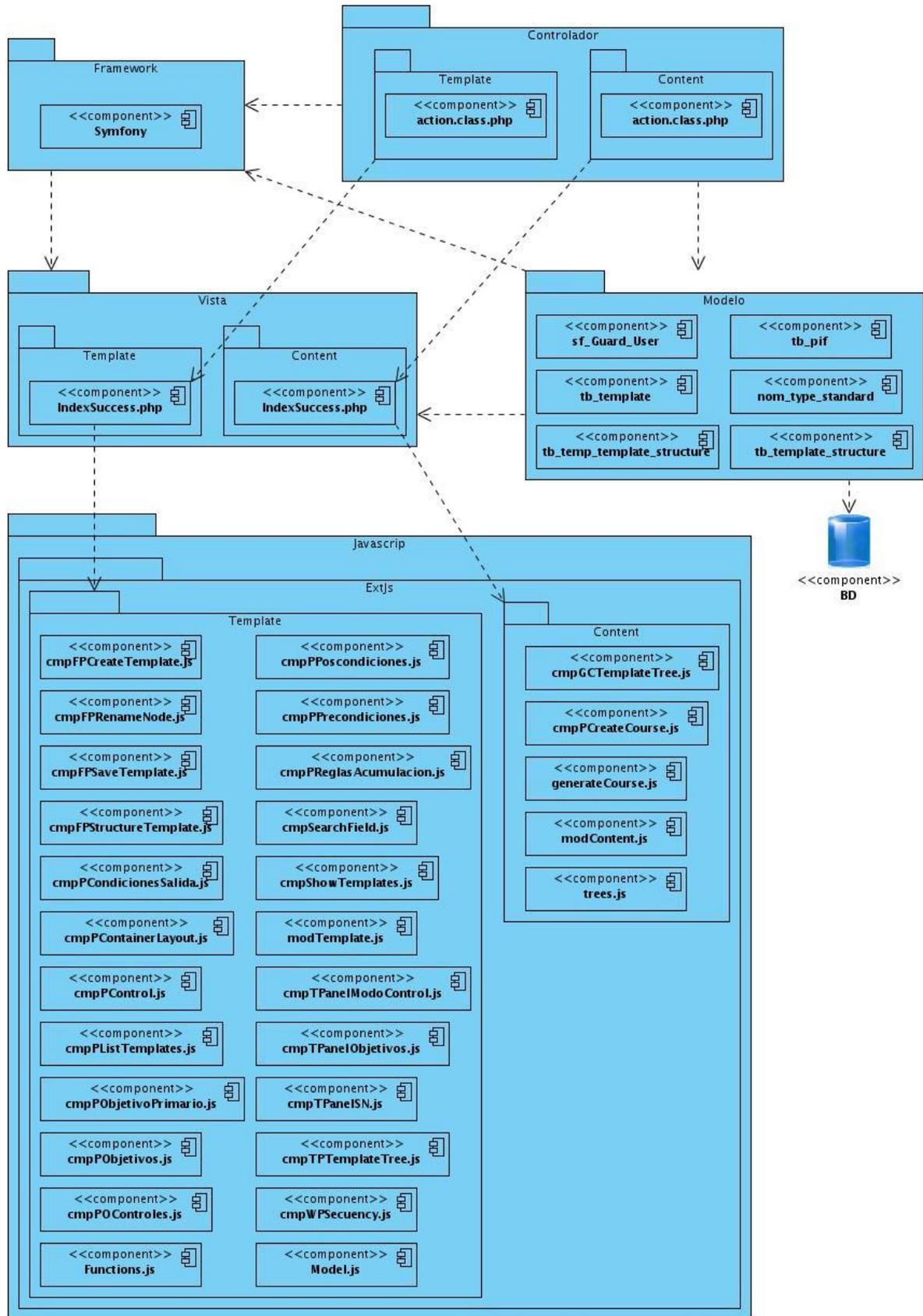


Figura 27: Diagrama de Componentes del módulo SCORM 2004.

## **Conclusiones**

En el presente capítulo se han implementado las funcionalidades identificadas. Además se completaron los artefactos del flujo de trabajo Implementación con el modelado de los Diagramas de Despliegue y de Componentes, los cuales brindan una vista física del sistema.

### *Conclusiones generales*

Al finalizar la investigación se puede afirmar que los objetivos trazados fueron cumplidos y se arribaron a las siguientes conclusiones:

- \*\* El análisis y diseño del módulo SCORM 2004, permitió crear un punto de partida para su implementación en la herramienta de autor CRODA.
- \*\* La solución presentada facilita la asignación de secuencia y navegación en los OA de la herramienta de autor CRODA, posibilitando al diseñador instruccional modelar el recorrido del estudiante por el Objeto de Aprendizaje, teniendo en cuenta su desempeño.
- \*\* Se le incorpora a la herramienta de autor CRODA la opción de exportar los Objetos de Aprendizaje en las versiones 1.2 y 1.3 del estándar SCORM.

## *Recomendaciones*

Una aplicación informática requiere de una constante actualización por lo que al concluir la investigación realizada se recomienda:

- \*\* Enriquecer la herramienta en cuanto a funcionalidades que permitan una mayor flexibilidad en la creación y edición de los OA.
- \*\* Mantener una constante investigación referida a los estándares utilizados para preservar la herramienta de autor actualizada.
- \*\* Elaborar un manual de ayuda para la creación de las reglas de secuencia.

## *Bibliografía*

- », **Giancarlo Corzo Deja una respuesta. 2008.** Desarrollo en Web. *ExtJS lo bueno, lo malo y lo feo.* [En línea] 22 de Octubre de 2008. <http://blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo/>.
- ADL. 2006.** *Sequencig and Navigation (SN).* 2006.
- ÁLVAREZ, Luís A. 2009.** *Creando OA con Reload.* 2009.
- Alvarez, Miguel Angel. 2011.** Desarrolloweb.com. *Qué es XHTML.* [En línea] 10 de Enero de 2011. <http://www.desarrolloweb.com/articulos/que-es-xhtml.html>.
- Alzate, Richard. 2009.** [En línea] 2009. <http://www.gratisblog.com/index.php?itemid=130280>.
- . **2010.** [En línea] 2010. <http://tecnofilos.aprenderapensar.net/2010/01/29/scorm-1-2-y-scorm-2004/>.
- anónimos, Autores.** [En línea] <http://www.visual-paradigm.com/product/vpuml/>.
- Autores. 2007.** Ecured. *Extjs.* [En línea] 1 de abril de 2007. [http://www.ecured.cu/index.php/Sencha\\_Ext\\_JS](http://www.ecured.cu/index.php/Sencha_Ext_JS).
- autores, Colectivo de. 2006.** [En línea] 2006. <http://hosting.udlap.mx/estudiantes/jose.ferrercz/reload%20editor.pdf>.
- . **2010.** [En línea] 2010. <http://roa.uci.cu/roa.php>.
- . **2009.** ADL. [En línea] Noviembre de 2009. [www.ADLNet.gov](http://www.ADLNet.gov).
- . **2011.** IMS Global Learning Consortium. [En línea] 3 de Enero de 2011. <http://www.imsglobal.org/simplesequencing/>.
- . **2007.** Internet Information Services. [En línea] 11 de Julio de 2007. <http://www.microsoft.com/spain/windowsserver2003/technologies/webapp/iis.mspx>.
- . **2011.** My SQL. [En línea] 4 de Enero de 2011. <http://dev.mysql.com/doc/refman/5.0/es/features.html>.
- . **2011.** What is phppgAdmin? [En línea] 2011. <http://phppgadmin.sourceforge.net/>.
- . **2010.** Zend Studio The Leoding PHP IDE. [En línea] 2010. <http://www.zend.com/en/products/studio/>.
- autores., Colectivo de. 2009.** Rational Rose Enterprise. [En línea] 2009. [www.essi.upc.edu/.../LESE-2%20-%20Introduccion%20a%20Rational%20Rose.ppt](http://www.essi.upc.edu/.../LESE-2%20-%20Introduccion%20a%20Rational%20Rose.ppt).
- Balboa, Marcos Dacosta. 2009.** Linwind. *Navicat-cliente MySQL, PostgreSQL y Oracle.* [En línea] 29 de Agosto de 2009. <http://www.dacostabalboa.com/es/navicat-cliente-mysql-postgresql-y-oracle/4722>.

**Blanco, Rubén González. 2009.** LESE-2 Introducción a Rational Rose. [En línea] 2009. [www.essi.upc.edu/.../LESE-2%20-%20Introduccion%20a%20Rational%20Rose.ppt](http://www.essi.upc.edu/.../LESE-2%20-%20Introduccion%20a%20Rational%20Rose.ppt).

**Cármaco, Jose L. 2010.** SCORM . [En línea] 20 de Mayo de 2010. <http://www.joseluisarcamo.info/?p=69>.

**CARRODEGUAS, RICARDO. 2008.** *Análisis y Diseño de una herramienta de autor Web Interoperable.* 2008.

**Cazarez, José Luis. 2010.** [En línea] enero de 2010. <http://www.slideshare.net/jcazare/softwre-de-educacin-udutu>.

**Cely, Milton. 2010.** [En línea] 2010. <http://www.moodlemoot.org.co/file.php/1/Memorias/Conferencia%20Milton%20Cely%20-%20SCORM%20%28Sharable%20Content%20Object%20Reference%20Model%29%20%5BModo%20de%20compatibilidad.pdf>.

**Cibernetia. 2009.** [En línea] 2009. [http://www.cibernetia.com/manuales/instalacion\\_servidor\\_web/1\\_conceptos\\_basicos.php](http://www.cibernetia.com/manuales/instalacion_servidor_web/1_conceptos_basicos.php).

**Cossio, Miguel Humberto. 2009.** *Diagrama de despliegue.* [En línea] 2009. [virtual.usalesiana.edu.bo/web/practica/archiv/ddespli2.doc](http://virtual.usalesiana.edu.bo/web/practica/archiv/ddespli2.doc) .

**Crysfel. 2009.** ExtJS Framework. [En línea] 2009. <http://www.quizzpot.com/2009/01/ext-js-framework/>.

**DataPrix. 2007.** DataPrix. *Cliente grafico: pgAdmin3.* [En línea] 2 de Enero de 2007. <http://www.dataprix.com/8-cliente-grafico-pgadmin3>.

**Escribano, Gerardo Fernández. 2002.** Introducción a Extreme Programming. [En línea] 9 de diciembre de 2002. <http://www.info-ab.uclm.es/asignaturas/42551/trabajosAnteriores/Presentacion-XP.pdf>.

**Etxeberría, Ana Landeta. 2007.** Libro de buenas práctica de e-learning. [En línea] 2007. <http://www.buenaspracticas-elearning.com/capítulo-16-estandares-e-learning.html>.

**exist, Comunidad. 2009.** [En línea] 2009. <http://exist.sourceforge.net/>.

**Félix, Alvaro del Castillo San. 2000.** El servidor de web Apache: Introducción práctica. [En línea] 2000. <http://acsblog.es/articulos/trunk/LinuxActual/Apache/html/x31.html>.

**Gómez, Alexandra Del Valle Brito. 2009.** Oracle. [En línea] 23 de Noviembre de 2009. <http://www.monografias.com/trabajos25/oracle/oracle.shtml>.

**Guzmán, Clara López. 2005.** *Los Repositorios de Objetos de Aprendizaje como soporte a un entorno e-learning.* . España. : s.n., 2005.

**Hinostroza, Raul Rodas. 2007.** LinuxCentro.net. *Características de PHP.* [En línea] 22 de Febrero de 2007. <http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>.

**Jacobson. 2000.** El proceso unificado de Desarrollo de Software. [En línea] 2000. <http://bibliodoc.uci.cu/pdf/reg00060.pdf>.

**Jeff. 2008.** Kioskea.net. [En línea] 16 de Octubre de 2008. <http://es.kioskea.net/contents/css/cssintro.php3>.

**John. 2010.** Manuales de ayuda. *Características de PostgreSQL*. [En línea] 2010. <http://www.manualesdeayuda.com/manuales/bases-de-datos/postgresql/caracteristicas-de-postgresql-01844.html>.

**Larman, Craig. 2008.** Modelo de dominio. [En línea] 27 de Abril de 2008. [http://migueljaque.com/index.php/tecnicas/tecnicasmodnegocio/37-modelado\\_negocio/46-modelo-de-dominio?tmpl=component&print=1&page=](http://migueljaque.com/index.php/tecnicas/tecnicasmodnegocio/37-modelado_negocio/46-modelo-de-dominio?tmpl=component&print=1&page=).

**Herrero, José L. Montero O'Farrill y Elsa. 2008.** *LAS HERRAMIENTAS DE AUTOR EN EL PROCESO DE...* Revista de Medios y Educación, Cuba : Pixel-Bit., 2008.

**Learning, Advanced Distributed. 2006.** *Content Aggregation Model (CAM)*. 2006.

**López, José María. 2011.** Softonic. *RapidSVN Cliente para subversion para principiantes y expertos*. [En línea] 2011. <http://rapidsvn.softonic.com/>.

**MaríaAgudelo, Mónica. 2006.** PLATAFORMAS EDUCATIVAS. [En línea] Agosto de 2006. <http://aprendeonline.udea.edu.co/banco/html/plataformaseducativas/>.

**Martínez, Bermi Ferrer. 2010.** Akelos Framework. [En línea] 2010. <http://www.akelos.org/>.

**Mendoza, Jorge A. 2003.** [En línea] 10 de Junio de 2003. <http://www.informaticamilenium.com.mx/paginas/mn/articulo78.htm>.

**NetBeans, Comunidad. 2010.** Bienvenido a NetBeans y [www.netbeans.org](http://www.netbeans.org). [En línea] 2010. [http://netbeans.org/index\\_es.html](http://netbeans.org/index_es.html).

**Ochoa, Sergio. 2005.** Introducción a los Patrones. [En línea] 2005.

**Ortega, Alvaro López. 2010.** Cherokee. [En línea] 2010. [http://www.cherokee-project.com/doc/basics\\_why\\_cherokee.html](http://www.cherokee-project.com/doc/basics_why_cherokee.html).

**Pecos, Daniel. 2010.** Características de PostGreSQL. [En línea] 2010. [http://danielpecos.com/docs/mysql\\_postgres/x15.html#AEN30](http://danielpecos.com/docs/mysql_postgres/x15.html#AEN30).

**Pérez, Javier Eguíluz. 2007.** librosweb.es. *Introducción a JavaScript*. [En línea] 2007. <http://www.librosweb.es/javascript/>.

**Potencier, Fabien. 2009.** *Symfony 1.2, la guía definitiva*. 2009.

**Ramírez, Elisa Arizaca. 2009.** Diagrama de componentes. [En línea] 2009. [virtual.usalesiana.edu.bo/web/practica/archiv/compon.doc](http://virtual.usalesiana.edu.bo/web/practica/archiv/compon.doc).

**Ros, Miguel Zapata. 2009.** *Secuenciación de contenidos. Especificaciones para la secuenciación instruccional de objetos de aprendizaje*. 2009.

**Schmuller, Joseph. 2008.** Aprender UML en 24 horas. [En línea] 2008. <http://www.ivlabs.org/home/?p=1800>.

**Sicilia, Miguel Angel. 2008.** Funciones de los Sistemas Gestores de Bases de Datos. [En línea] 23 de Septiembre de 2008. <http://cnx.org/content/m17543/latest/>.

**Usuarios, Grupo de. 2010.** pgAdmin. [En línea] 2010. <http://www.arpug.com.ar/trac/wiki/PgAdmin>.

**Valdés, Damián Pérez. 2007.** ¿Qué es Javascript? *Maestros del Web*. [En línea] 3 de Julio de 2007. <http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>.

**Web, El equipo de Maestros del. 2010.** Maestros del Web. *Los Frameworks de PHP agilizan tu trabajo*. [En línea] 2010. <http://www.maestrosdelweb.com/editorial/los-frameworks-de-php-agilizan-tu-trabajo/>.

**Web, Maestros del. 2010.** Maestros del Web. *El framework Symfony, una introducción práctica (I parte)*. [En línea] 2010. <http://www.maestrosdelweb.com/editorial/el-framework-symfony-una-introduccion-practica-i-parte>.

**Yepes, Jesús. 2010.** Blog and Web. [En línea] 2010. <http://blogandweb.com/php/introduccion-a-code-igniter-i/>.