



**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**FACULTAD 4**

**ANÁLISIS, DISEÑO, IMPLEMENTACIÓN Y PRUEBA DE UNA EXTENSIÓN PARA LA PLATAFORMA  
ZERA, QUE PERMITA LA INCORPORACIÓN DEL ESTÁNDAR SCORM**



**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS  
INFORMÁTICAS**

**Autores:**

Lissette Feria Sotto.

Yerandy Manso Guerra.

**Tutores:**

**Ing.** Jorge Antonio Díaz Gutiérrez

**MCs.** Roberto López Dosagües



## Declaración de autoría

Declaramos que somos los únicos autores del trabajo Análisis, Diseño Implementación y Prueba de una extensión para la plataforma ZERA, que permita la incorporación del estándar SCORM, y autorizamos a la Facultad 8 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Autores:

\_\_\_\_\_  
Lissette Feria Sotto

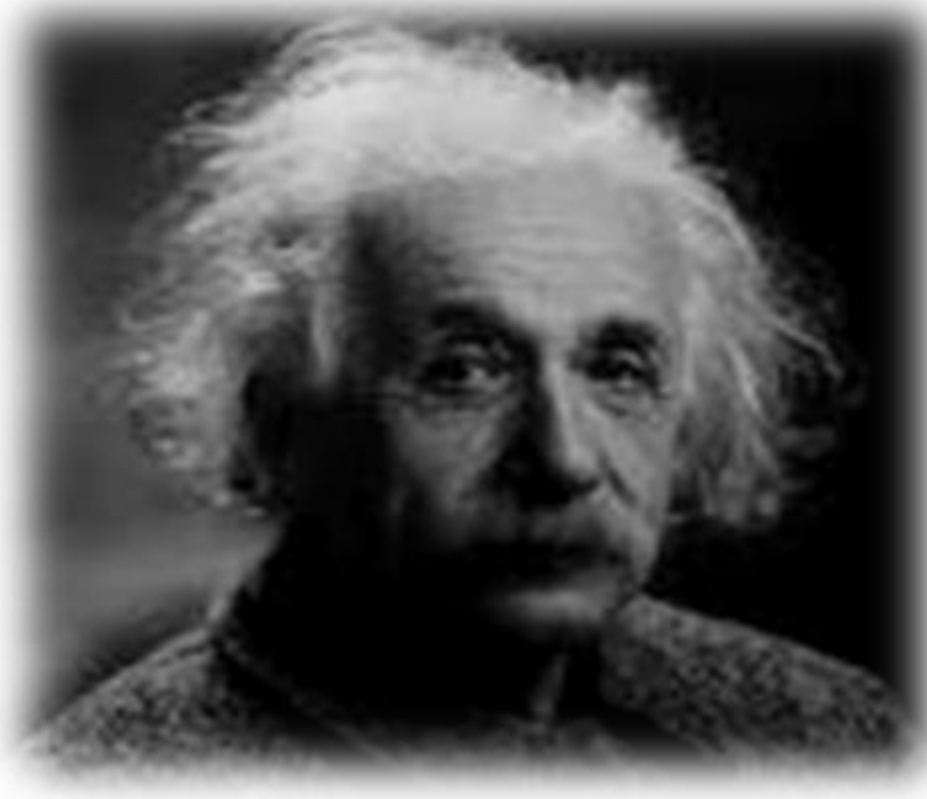
\_\_\_\_\_  
Yerandy Manso Guerra

\_\_\_\_\_  
Ing. Jorge Antonio Díaz Gutiérrez

\_\_\_\_\_  
MCs. Roberto López Dosagües

Análisis, diseño, implementación y prueba de una extensión para la plataforma ZERA, que permita la incorporación del estándar SCORM.

---



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

Albert Einstein



## DEDICATORIA

### LISSETTE

A mi papá Soto, que aunque no esté entre nosotros, siempre quiso que cumpliera su sueño más deseado, el ser ingeniera.

A mi mamá Nilda que la quiero con todo mi corazón y aún no es suficiente. Por haberme educado y brindado su cariño durante toda mi vida.

A mi papá por creer siempre en mí y darme todo su amor en cada momento. Por ser el hombre más importante de mi vida.

A mi mamá por ser mi mayor inspiración y ejemplo a seguir. A la que le debo todo lo que he logrado por el empeño que ha puesto en hacer de mí una gran mujer y mejor persona.

A mi hermano, lo quiero mucho y espero poder servirle de ejemplo. A toda mi familia que siempre ha estado pendiente de cada uno de mis pasos.

### YERANDY

A mi abuela Ercilia y abuelo Alfonso siempre los recordaré y los amaré no como abuelos sino como padres que es lo que fueron para mí.

A mis padres por darme la vida, por quererme y sacrificarse por mí, por darme todo su amor y apoyo esta tesis es lo primero de muchas cosas que quiero dedicarle.

A mi hermana por ser tan buena conmigo, por complacerme en todo, por ser especial, para ella todo mi corazón.



## AGRADECIMIENTOS

### LISSETTE

A mis compañeros de estudio, los viejos y los nuevos por compartir junto a mí estos maravillosos 5 años de carrera. En especial a Adrian, Raidel, Lia, Idalmis, Lisandra, Edith, Pachi y Yoyi por haberse comportado como mis hermanos, por comprenderme y apoyarme siempre.

A Yerandy por ser mi mejor pareja y compañero de tesis.

### YERANDY

A todos mis amigos por ser geniales.

A Pedro y a Maravilla locos sin comparación y que tanto me han ayudado en cada pase.

A Portales y Magalis los quiero mucho, gracias por ser especiales.

A Lissette, por compartir los últimos 3 años conmigo, sin ti no lo hubiéramos logrado.

### JUNTOS

A los tutores Jorge y Roberto que siempre estuvieron para apoyarnos y criticarnos, que más que tutores son amigos.

A Abel por ser un líder de proyecto sin igual y un gran amigo.

A todos los que dieron un granito de arena para obtener este triunfo.



## RESUMEN

El uso de las tecnologías Web en la educación, ha sido excelente medio para romper con las limitantes geográficas y temporales de los esquemas tradicionales del proceso enseñanza - aprendizaje. Con el desarrollo de nuevas tecnologías, han surgido diferentes herramientas, las que han posibilitado el progreso de los Sistema de Administración de Aprendizaje (LMS). Dada la heterogeneidad de los LMS, han surgido estándares que posibilitan la comunicación y reutilización de los recursos y contenidos , entre los que se encuentra, el Modelo de Referencia de Contenidos Compatibles (SCORM).

La investigación muestra el desarrollo de una extensión, que permite la interoperabilidad, reusabilidad y escalabilidad con distintas plataformas, haciendo uso del estándar SCORM 2004 3ra Edición, como parte de los esfuerzos que realiza La Universidad de las Ciencias Informáticas, en el proceso de enseñanza y aprendizaje.

En el desarrollo de la solución, se utilizó el marco de trabajo definido por el Framework de PHP Symfony. Se realiza un análisis detallado sobre los lenguajes de programación empleados, como los gestores de base de datos y los marcos de trabajo que Symfony define para la presentación, acceso a datos y lógica del negocio. Se describen los algoritmos más importantes utilizados.

Se detalla el resultado del estado del arte realizado, que permitió definir la metodología empleada, así como un levantamiento bibliográfico del objeto de estudio.

El resultado de la investigación, muestra una extensión que permite la importación y exportación de recursos educativos, lo que garantiza la estandarización y reutilización de los mismos.

Se presentan recomendaciones que permiten dar continuidad al desarrollo de esta investigación, así como su mantenimiento.



ÍNDICE	PÁGINAS
RESUMEN.....	6
INTRODUCCIÓN.....	9
<b>CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.....</b>	<b>12</b>
1.1 - Conceptos y definiciones asociados al problema de investigación.....	12
1.2.- Organismos e instituciones que participan en los procesos de estandarización en e-learning .....	17
1.3.- Especificaciones y Estándares más utilizados en e-learning.....	18
1.4.- Análisis de sistemas existentes que emplean el estándar SCORM.....	20
1.6.- Aplicaciones cliente-servidor .....	22
1.7.- Análisis de diversas Metodologías de Desarrollo para la propuesta de solución del producto .....	25
1.8.- Análisis de Herramientas CASE para la propuesta de solución del producto .....	27
1.9.- Para la construcción de bocetos o maquetas de los prototipos de interfaz de usuario, se selecciona la herramienta Balsamiq Mockups (versión 1.6.58).....	28
1.10.- Análisis de Entornos e IDEs de Desarrollo para la propuesta de solución del producto .....	29
1.11.- Análisis de Sistemas Gestores de Base de Datos para la propuesta de solución del producto ..	30
1.12.- Metodología, Herramientas, IDE y lenguajes de programación seleccionados para el desarrollo del producto.....	31
1.13.- Conclusiones parciales.....	33
<b>CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA .....</b>	<b>34</b>
2.1.- Modelo de Dominio .....	34
2.2.- Descripción del sistema propuesto .....	35
2.3.- Requisitos del Software .....	35
2.4.- Definición de los actores del sistema.....	38
2.5.- Diagrama de Casos de Uso del Sistema.....	38
2.6.- Descripción de los Casos de Uso pertenecientes al DCUS. ....	39
2.7.- Conclusiones parciales.....	42
<b>CAPÍTULO III: ANÁLISIS Y DISEÑO.....</b>	<b>43</b>
3.1.- Estructura del paquete SCORM.....	43
3.2.- Modelo de análisis .....	46



3.4.- Modelo de diseño.....	49
3.5.- Diseño de la base de datos .....	52
3.6.- Conclusiones parciales.....	57
<i>CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBA.....</i>	<i>58</i>
4.1.- Estándares de código.....	58
4.2.- Descripción de las principales clases utilizadas. ....	58
4.3.- Proceso de Implementación. ....	59
4.4.- Descripción de los algoritmos más importantes. ....	61
4.5.- Modelo de Implementación .....	62
4.6.- Pruebas del Software .....	63
4.7.- Conclusiones parciales.....	74
<i>CONCLUSIONES GENERALES.....</i>	<i>75</i>
<i>RECOMENDACIONES.....</i>	<i>76</i>
<i>BIBLIOGRAFÍA.....</i>	<i>77</i>





## INTRODUCCIÓN

El uso de las tecnologías Web en la educación ha sido excelente medio para romper con las limitantes geográficas y temporales de los esquemas tradicionales de la enseñanza-aprendizaje. Su adopción y uso han sido amplios, lo que ha permitido un desarrollo rápido y consistente en el que la Web ha ido tomando distintas formas dentro de los procesos educativos, por lo que se ha convertido en una infraestructura básica que facilita el desarrollo de los procesos de enseñanza-aprendizaje, tanto presenciales como no presenciales, lo que ha dado lugar a un modelo conocido como e-formación o e-learning, este modelo a tomado cada vez más auge y ha sido utilizado como un complemento que se ha de adaptar según las necesidades de cada institución.

A raíz del desarrollo de estas nuevas modalidades, han surgido diferentes herramientas, las mismas han evolucionado hasta los actuales Sistema de Administración de Aprendizaje (Learning Management System, en adelante LMS), estos proveen las funciones administrativas y de seguimiento necesarias para posibilitar y controlar el acceso a los contenidos, implementar recursos de comunicaciones y llevar a cabo el seguimiento de quienes utilizan la herramienta. En general, los LMS facilitan la interacción entre los docentes y los estudiantes, aportan herramientas para la gestión de contenidos académicos y permiten el seguimiento y la evaluación. Es decir, facilitan la “simulación” del modelo real en el mundo virtual.

## SITUACIÓN PROBLÉMICA

Dada la heterogeneidad de los LMS, se han desarrollado estándares que garantizan interoperabilidad, reusabilidad, manejabilidad, accesibilidad, durabilidad y escalabilidad entre las distintas plataformas. Entre estos estándares se encuentra el Modelo de Referencia Compartible (Shareable Content Object Reference Model, en adelante SCORM), siendo este un conjunto de especificaciones para el desarrollo, empaquetamiento y distribución de material educativo en cualquier momento y en cualquier lugar. El estándar SCORM, asegura que este material es: reutilizable, accesible, interoperable y durable.

En la Universidad de las Ciencias Informáticas se lleva a cabo, el desarrollo de un LMS llamado ZERA. Dado el nivel de complejidad que posee, se encuentra en construcción, por lo que, a lo largo del desarrollo del mismo se continúan identificando necesidades. Una de ella se convierte en primordial, refiérase a la no existencia de una funcionalidad que permita la interoperabilidad e intercambio de contenidos y recursos



tales como imágenes, videos, contenidos, animaciones, simulaciones y audios con otras plataformas utilizando las especificaciones del estándar SCORM.

Teniendo en cuenta lo expuesto anteriormente, se identificó como **problema a resolver** *¿Cómo permitir la interoperabilidad e intercambio de contenidos y recursos entre plataformas que cumplan con el estándar SCORM?*

**Objetivo General:** *Desarrollar una extensión para la plataforma ZERA que permita la exportación e importación de recursos educativos o partes de los mismos, que se basen en las normas del estándar SCORM.*

**Objeto de Investigación:** *Los procesos de interoperabilidad e intercambio de contenidos y recursos entre plataformas Web.*

**Campo de Acción:** *Procesos de exportación e importación de recursos para las plataformas educativas.*

**Idea a defender:** *El desarrollo de una extensión para la plataforma ZERA permite la importación y exportación de recursos educativos y garantiza la estandarización y reutilización de los mismos, así como la interoperabilidad entre distintos LMS que soporten el estándar SCORM.*

### **Métodos de Investigación**

Como **métodos teóricos** se utilizan:

- ✓ Análisis - Síntesis: En el análisis de documentos para la captura y levantamiento de requisitos que permitan obtener los rasgos que caractericen al negocio y que sirvan de ayuda para procesar la información y elaborar conclusiones.
- ✓ Histórico - Lógico: En el estudio realizado sobre las tecnologías que existen actualmente para poder realizar la selección de las que se van a utilizar de acuerdo con las características propias del sistema a desarrollar.
- ✓ Modelación: En la confección de los diagramas que permitirán representar la propuesta de solución.

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



- ✓ Realizar el levantamiento bibliográfico de la literatura a emplear para el estudio y desarrollo del problema de investigación.
- ✓ Estudiar las políticas y normas del estándar SCORM y LOM.
- ✓ Analizar y diseñar la propuesta de visualización de las funcionalidades correspondientes a la extensión de la Plataforma ZERA.
- ✓ Implementar la extensión.
- ✓ Integrar la extensión creada con la plataforma ZERA.
- ✓ Realizar las pruebas de calidad a lo largo de todo el desarrollo de la extensión.

### Estructura Capítular

En el **Capítulo I: “Fundamentación Teórica”**, se expone el estado del arte y los conceptos vinculados al objeto de estudio, con el fin de lograr mejor comprensión del problema de la investigación. Se describe además, los lenguajes de programación y sistemas para realizar el análisis y diseño de la aplicación, así como las tendencias y tecnologías actuales sobre las que se basa. Finalmente se da a conocer la selección de los elementos que van a permitir el desarrollo del producto informático.

En el **Capítulo II: “Características del Sistema”**, se especifican los requisitos funcionales que debe cumplir el sistema, así como el diagrama de CU del sistema y las descripciones de los mismos.

En el **Capítulo III: “Análisis y Diseño”**, se realiza el análisis y diseño que sustenta la propuesta de solución del sistema a desarrollar.

En el **Capítulo IV: “Implementación y Prueba”**, se lleva a cabo la implementación del sistema a través de los lenguajes de programación y las metodologías seleccionados. También se arriban a conclusiones sobre las verificaciones efectuadas de las pruebas realizadas, para comprobar que el sistema brinde las funcionalidades acordadas.



## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se expone el estado del arte y los conceptos vinculados al objeto de estudio, con el fin de lograr una mejor comprensión del problema de la investigación. Se describe además, los lenguajes de programación y sistemas para realizar el análisis y diseño de la aplicación así como las tendencias y tecnologías actuales sobre las que se basa. Finalmente se da a conocer la selección de los elementos que van a permitir el desarrollo del producto informático con tecnología WEB.

### 1.1 - Conceptos y definiciones asociados al problema de investigación.

Para el desarrollo de la presente investigación, es necesario definir los conceptos básicos que permiten mejor comprensión del problema de investigación.

#### 1.1.1 - Software educativo

Del estudio realizado sobre el concepto que se expone tanto en Cuba como en el mundo, se arriba a la conclusión de que existen muchas definiciones, entre las que se destacan los siguientes:

*“Los software educativos, se definen de forma genérica como aplicaciones o programas computacionales que faciliten el proceso de enseñanza aprendizaje”. (1)*

*“Programas educativos y programas didácticos como sinónimos para designar genéricamente los programas para ordenador creados con la finalidad específica de ser utilizados como medio didáctico, es decir, para facilitar los procesos de enseñanza y de aprendizaje”.(2)*

De los conceptos expuestos, los autores coinciden en que cuando se habla de software educativo (en adelante SE), se refiere a programas educativos o didácticos, creados con la finalidad de ser utilizados, para facilitar los procesos de enseñanza - aprendizaje.

#### 1.1.2. - Base de Datos

De igual forma, Base de Datos (en adelante BD) lo define en diferentes artículos varios investigadores, como:

*“...una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular”. (3)*

*“Es un conjunto exhaustivo no redundante de datos estructurados organizados independientemente de su*



*utilización y su implementación en máquina accesibles en tiempo real y compatibles con usuarios concurrentes con necesidad de información diferente y no predicable en tiempo". (4)*

Para la presente investigación, como BD se define, un sistema formado por un conjunto de datos almacenados y estructurados, en el cual se almacena información en campos o delimitadores, teniendo acceso a ella posteriormente, tanto de forma separada como de forma conjunta. Se utiliza normalmente para recoger grandes cantidades de información.

### **1.1.3. - Extensión**

Existen muchas definiciones de Extensión (o Plug-in) en la WEB, tales como:

*"Accesorio adicional al browser que permite llevar a cabo tareas adicionales concretas". (5)*

*"Complemento que permite ejecutar otra aplicación desde una página web, las más típicas son los reproductores de audio y video". (6)*

*"Programas que se agregan a un navegador del World Wide Web (en lo adelante WWW) los cuales realizan funciones determinadas. Producen la visualización de archivos multimedia y dan soporte a archivos gráficos no estándares con el visualizador". (7)*

Estas definiciones van dirigidas principalmente hacia los navegadores web, por lo que para la investigación se define, como Plug-in, un programa que es independiente a la aplicación siguiendo la arquitectura establecida para el producto, puede ser instalado y utilizado como parte de un software agregándole funciones adicionales que anteriormente no poseía.

### **1.1.4. - Sistema e-learning**

El e-learning es un sistema de educación electrónico o a distancia, en el que se integra el uso de las tecnologías de la información y elementos pedagógicos (didácticos) para la formación, capacitación y enseñanza de los usuarios o estudiantes en línea, es decir, se basa en adquirir conocimientos por medios electrónicos. Se puede entender como una modalidad de aprendizaje dentro de la educación a distancia y se define como e-learning. Emplea herramientas y medios diversos como Internet, Intranets, CD-ROM, producciones multimedia (textos, imágenes, audios, videos, etc.), entre otros. Literalmente e-learning, es aprendizaje con medios electrónicos: enseñanza dirigida por la tecnología.



### 1.1.5. - Plataforma e-learning

Una plataforma e-learning o también conocido por el término "enseñanza virtual", es un sistema de formación interactivo para desarrollar programas de enseñanza, que hace uso masivo de los medios electrónicos, para llegar a un alumnado generalmente remoto. O sea, es una capacitación no presencial que, a través de plataformas tecnológicas, posibilita y flexibiliza el acceso y el tiempo en el proceso de enseñanza-aprendizaje, adecuándolos a las habilidades, necesidades y disponibilidades de cada docente.

#### **Criterios fundamentales de los sistemas e-learning**

Los criterios de los sistemas e-learning son referenciados en varios artículos por diferentes investigadores sobre todo de países del primer mundo. A continuación se puntualizan algunos de estos criterios, por ser los que más se identifican con la investigación:

- ✓ Los sistemas e-learning trabajan en red, lo que lo hace capaz de ser instantáneamente actualizado, almacenado, recuperado, distribuido y permite compartir instrucción o información.
- ✓ Es entregado al usuario final a través del uso de ordenadores utilizando tecnología estándar de Internet.
- ✓ Se enfoca en la visión más amplia del aprendizaje que van más allá de los paradigmas tradicionales de capacitación". Desde la perspectiva que ofrece la experiencia en el desarrollo y explotación de plataformas e-learning.(8)

### 1.1.6. - Learning Management System

Los Sistemas de Gestión de Aprendizaje son uno de los elementos fundamentales del *e-learning*. Un LMS permite la publicación de contenidos, el acceso a los contenidos, la gestión de los recursos y la comunicación entre todos los actores implicados (alumnos, profesores, administradores del sistema y creadores de contenidos). Además, el sistema gestiona habitualmente los accesos, las actividades y los permisos del usuario.

### 1.1.7. - Objetos de Aprendizaje

El elemento central en la nueva forma de desarrollar los cursos es el objeto de aprendizaje (en adelante OA). La definición más citada en la literatura, es la del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, Asociación técnico-profesional mundial dedicada fundamentalmente a la estandarización), propuesta en uno de los pocos estándares relacionados con e-learning que han sido aprobados:



*“Un objeto es cualquier entidad digital o no digital que puede ser usada, re-usada o referenciada para el aprendizaje soportado en tecnología”. (9)*

Esta es una definición excesivamente genérica y que ha hecho que se proporcionen otras definiciones más específicas como la de Wiley (2000) (10):

*“Cualquier recurso digital que pueda ser reutilizado como soporte para el aprendizaje.”*

Wiley también matiza que *“...se usa para designar material educativo diseñado y creado en pequeñas unidades con el propósito de maximizar el número de situaciones educativas en las que se puede utilizar dicho recurso”.* (10)

Esta idea está directamente recogida en la definición proporcionada por Polsani (2003) que lo define como:

*“Unidad didáctica de contenido, auto – contenida e independiente, predispuesta para su reutilización en múltiples contextos instruccionales.”* (11)

En la concepción de un OA debe pensarse que sean recursos con atributos específicos para su interacción en un entorno e-learning, fácil de localizar, utilizar, almacenar y compartir. Para ello, estos recursos deben ser (Rehak & Mason, 2003):

**Accesibles:** Pueden ser indexados para una localización y recuperación más eficiente, utilizando esquemas estándares de metadatos.

**Interoperables:** Pueden operar entre diferentes plataformas de hardware y software.

**Portables:** Pueden moverse y albergarse en diferentes plataformas de manera transparente, sin cambio alguno en estructura o contenido.

**Durables:** Deben permanecer intactos a las actualizaciones de software y hardware. (12)

### 1.1.8.- Estándares en e-learning

Una de las principales funciones de los estándares en e-learning es servir como facilitadores de la durabilidad y de la reutilización en el tiempo de los contenidos y de la interoperabilidad, es decir, facilitar el intercambio de los contenidos entre diversas plataformas y sistemas.



Con la aparición de los estándares, a partir del año 2001, se garantizaba la independencia de los contenidos y los LMS, de forma que se cumplan ciertas especificaciones sobre las que basar el desarrollo de herramientas y contenidos.

Las ventajas de la estandarización posibilitan que se pueda elegir libremente los proveedores de contenidos y herramientas, la reutilización de los cursos en plataformas diferentes, abaratando considerablemente las inversiones que hay que realizar en planes de formación.

Actualmente hay diversos estándares utilizables, como son el AICC (desarrollado por la industria de la aviación de Estados Unidos), IEEE LTSC (Instituto de Ingenieros Electrónicos e Informáticos), IMS (del inglés Global Learning Consortium), y el más utilizado y extendido SCORM.

Estos estándares abordan aspectos relativos a los contenidos, cómo se empaquetan los cursos, cómo se describen tanto los cursos, como los propios elementos OA y cómo se describen las evaluaciones, o exámenes de modo que puedan ser intercambiables entre sistemas.

Los estándares favorecen la máxima integración entre distintas tecnologías, convirtiéndose en un componente facilitador de la accesibilidad y la reutilización de materiales, la creación de cursos más ricos y una formación más global, dado que los materiales creados en una herramienta pueden ser vistos por otra y exportados para una tercera.(13)

#### **1.1.9.- Framework**

Un Framework es una estructura de archivos y utilidades que aceleran la programación de una aplicación informática, proveyendo una metodología de trabajo que sistematiza y facilita la generación de formularios, funciones y módulos de uso común, permitiendo al desarrollador dedicar su atención hacia los aspectos específicos de cada aplicación.

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio. (14)

#### **1.1.10.- Extensible Markup Language**





Todas las especificaciones, normas y estándares tecnológicos actuales usan el lenguaje Extensible Markup Language (en adelante XML). Este sirve para describir y transportar por la red las páginas Web. En el ámbito de eLearning es el lenguaje que se usa para describir prácticamente cualquier cosa. La ventaja del uso del XML es su gran aceptación y el alto número de herramientas tecnológicas que hacen muy fácil el desarrollo y tratamiento informático (interpretar un documento XML).

## **1.2.- Organismos e instituciones que participan en los procesos de estandarización en e-learning**

Existen organismos e instituciones que contribuyen de una forma u otra en la estandarización de los procesos llevados a cabo en los e-learning, entre ellas se encuentran las siguientes:

### **1.2.1.- Advanced Distributed Learning**

En Noviembre de 1997 el Departamento de Defensa de EE.UU. y la oficina de Ciencia y Tecnología de la Casa Blanca lanzaron la iniciativa Advanced Distributed Learning (en adelante ADL). El propósito de ADL es desarrollar el e-learning para asegurar el acceso a materiales educativos y de alta calidad que puedan ser adaptados a las necesidades individuales y que se puedan distribuir de forma sencilla. ADL surge como respuesta a las necesidades de uno de los mayores consumidores de software del mundo, y forma parte del esfuerzo que el gobierno norteamericano viene realizando con el objetivo de conseguir una enseñanza de calidad, en el que también están implicados los Departamentos de Educación y Trabajo.

ADL ha sido una de las organizaciones más activas en el esfuerzo de la estandarización de las tecnologías de aprendizaje, en colaboración con otras iniciativas principalmente IEEE, IMS y AICC. Su principal resultado es un conjunto de especificaciones que, bajo la denominación SCORM (ADL SCORM, 2002, 2006) propone un modelo de agregación de contenidos (Content Aggregation Model, CAM), un entorno de tiempo de ejecución (Run-Time Environment, RTE) y la secuenciación y navegación (Sequencing and Navigation, SN) de los contenidos. Actualmente SCORM es la norma que está teniendo un mayor impacto en la industria, ya que es la que se ha implementado en un mayor número de sistemas.

(15)

### **1.2.2.- IMS Global Consortium**

IMS Global Learning Consortium es un grupo independiente, sin ánimo de lucro que inició su labor en 1997 impulsado por el National Learning Infrastructure Initiative (en adelante NLII) que es una



organización apoyada por Educase. Aunque inicialmente surgió como una iniciativa en EEUU, ahora en IMS participan instituciones educativas de todo el mundo (desde universidades a pequeñas empresas de formación), fabricantes, y vendedores de aplicaciones software para la educación.

Actualmente es el principal promotor y desarrollador de especificaciones abiertas orientadas a la enseñanza electrónica. Su objetivo es que, a partir de estas especificaciones, se consiga la interoperabilidad de aplicaciones y servicios en la enseñanza electrónica para que los autores de contenidos y de entornos puedan trabajar conjuntamente. (15)

### **1.3.- Especificaciones y Estándares más utilizados en e-learning**

Sobre las especificaciones de los sistemas e-learning más empleados en el empaquetamiento de cursos y recursos para el desarrollo de los LMS, de la bibliografía consultada y de los estudios realizados se arriba a la conclusión de que existen varias especificaciones las que a continuación se mencionan: IMS Content Packaging, IMS Question & Test Interoperability Specification, IMS Learning Design, IMS Learner Information Package Specification, IEEE Learning Object Metadata / IMS Learning Resources Metadata Specification, Learning Object Metadata y SCORM. De todas ellas nos detenemos en Learning Object Metadata (en adelante LOM) y SCORM, por constituir ella la base del desarrollo de nuestra investigación.

#### **1.3.1. - IEEE Learning Object Metadata / IMS Learning Resource Metadata Specification**

Actualmente IEEE Learning Object Metadata (Metadatos para Objetos de Aprendizaje), es el estándar de e-learning formalmente aprobado que goza de mayor aceptación, y que ha sido adoptado en la especificación de IMS Learning Resources Metadata. De hecho LOM se basa en los esfuerzos previos hechos para la descripción de recursos educativos en los proyectos ARIADNE, IMS y DublinCore. (16)

Los metadatos LOM, son información añadida a los materiales digitales que facilitan su clasificación y posterior recuperación. La especificación de metadatos adecuados para los materiales educativos es indispensable a fin de añadir valor a los mismos, en el sentido de facilitar su reutilización. Los materiales enriquecidos convenientemente con metadatos podrán almacenarse en bibliotecas digitales de contenidos educativos (por ejemplo, repositorios de OA). Estas bibliotecas soportarán, entonces, consultas significativas que permitirán la recuperación de aquellos materiales almacenados que cubran una determinada necesidad pedagógica. Estos metadatos proporcionan descripciones, propiedades e



información sobre los objetos de aprendizaje que permiten caracterizarlos, de forma que se simplifica su uso y gestión. De forma coloquial, lo que se busca mediante esta información complementaria es poder saber cuál es el contenido y el propósito de un OA sin tener que acceder a dicho contenido. Por tanto, los metadatos aportan información orientada a hacer más eficiente la búsqueda y utilización de los recursos. (16)

### 1.3.2.- Estándar SCORM

El estándar SCORM define una forma específica de la construcción de sistemas de gestión de aprendizaje y contenidos de formación para que trabajen bien con otros sistemas. Básicamente, las distintas versiones de SCORM que rigen: el contenido de los envases y el intercambio de datos en tiempo de ejecución. El contenido del paquete determina cómo una pieza de contenido debe ser entregado en un sentido físico. El centro de envasado SCORM es un documento XML titulado "*imsmanifest.xml*". Este archivo contiene cada pieza de información requerida por el LMS para importar y lanzar contenido sin la intervención humana. Este archivo de manifiesto contiene un XML que describe la estructura de un curso, tanto desde la perspectiva del alumno y desde una perspectiva del sistema de archivos físicos. En concreto SCORM se sustenta sobre las siguientes especificaciones:

- ✓ IEEE Data Model for Content Object Communication.
- ✓ IEEE ECMAScript Application Programming Interface for Content to Runtime Services Communication.
- ✓ IEEE Learning Object Metadata.
- ✓ IEEE Extensible Markup Language Schema Binding for Learning Object Metadata Data Model.
- ✓ IMS Content Packaging.
- ✓ IMS Simple Sequencing.

#### Existen varias versiones de SCORM:

**SCORM 1.1:** fue fundamentalmente el primer paso, y nunca ha ganado una amplia aceptación. Algunos productos aún lo apoyan, pero no se ha adoptado ampliamente.

**SCORM 1.2:** resuelve muchos de los problemas 1.1, es la versión ampliamente adoptada. A partir de octubre de 2005, todos los principales LMS continúan su apoyo, y la mayoría de los proveedores de contenido aún producen contenido que cumpla con la especificación 1.2.



**SCORM en su versión del 2004:** este modelo ha pasado a incluir también la secuenciación y navegación (Sequencing and Navigation SN) de los contenidos. Esta secuenciación define como se aplica y extiende IMS Simple Sequencing para un sistema SCORM.

SCORM 2004 define un modelo de software que describe el modelo de agregación de contenidos, las interrelaciones establecidas entre las componentes de los cursos, los modelos de datos y los protocolos de comunicación, de manera que los OA definidos en un LMS puedan compartirse entre diferentes LMS, a medida que este estándar ha sido aprobado por la comunidad internacional han surgido algunas mejoras conocidas por ediciones, en la actualidad SCORM 2004 va por su 4ta Edición, aunque está pobremente documentada y se ha puesto poco en práctica, por estas cuestiones la más usada es la 3ra Edición.

Esto permite interacciones de contenido sustancialmente más rico y un enorme aumento en la reutilización de recursos. La adopción ha sido lenta, a este punto, pero el número de LMS y proveedores de contenido SCORM 2004 de apoyo está aumentando considerablemente. (16)

### **1.3.3.- ¿Por qué utilizar el estándar SCORM 2004 en la plataforma ZERA?**

El estudio realizado acerca de los estándares e-learning más usados, arroja la conclusión que SCORM 2004 3ra Edición, por ser el más ampliamente aceptado y por aportar el complejo requisito de la secuenciación (conjunto de reglas que especifican el orden en el que un usuario puede visitar el contenido), además de definir un entorno de comunicación entre el contenido y el LMS, y proveer un seguimiento continuo del usuario, es seleccionado para ser aplicado en ZERA.

## **1.4.- Análisis de sistemas existentes que emplean el estándar SCORM**

### **1.4.1.- Sistemas a nivel internacional que utilizan el estándar SCORM**

El uso del estándar SCORM en sistemas LMS se ha extendido a lo largo del mundo, a continuación expondremos algunas plataformas que poseen entre sus funcionalidades este estándar:

#### **1.4.1.1.- Plataforma Claroline**

Se inició en el año 2001 por la Universidad Católica de Louvain, Bélgica (UCL) y ha sido desarrollado siguiendo las necesidades y experiencia pedagógica de los profesores. El proyecto fue subvencionado financieramente por la Fondation Louvain. Desde el 2004, el CERDECAM está contribuyendo significativamente al desarrollo de la plataforma. (18)



Publicada bajo una licencia de código abierto o software libre (Open Source), Claroline permite crear y administrar cursos a cientos de organizaciones de 93 países diferentes y la colaboración de espacios online.

La mayoría de los docentes se familiarizan muy rápido con Claroline, en muchos casos en tan solo dos o tres horas sin que estos tengan algún tipo de conocimientos técnicos de computación. De esta forma, ellos solamente se pueden concentrar en las cosas que realmente importan para enseñar tales como: el contenido, un buen escenario educacional, pero recordando la autonomía y la no necesidad de un equipo técnico para administrar el sitio Web del curso.

#### **1.4.1.3.- Plataforma MOODLE:**

Es un Sistema de Gestión de Cursos de Código Abierto (Open Source Course Management System). Este es una herramienta para producir cursos basados en Internet, páginas Web y procedimientos que permitan fácilmente la comunicación a través de Internet y el trabajo colaborativo.

Características generales de la plataforma MOODLE (20)

- ✓ Moodle es software libre, con Licencia pública GNU. Básicamente, esto significa que los usuarios de Moodle tienen algunas libertades: pueden copiar, usar y modificar Moodle siempre que acepten proporcionar el código fuente a otros, no modificar o eliminar la licencia original y los derechos de autor.
- ✓ Los recursos que el docente entrega a sus estudiantes pueden ser de cualquier fuente y con cualquier formato, puesto que su programación está orientada a objetos.
- ✓ Lleva registro y seguimiento completo de los accesos del alumno. Se dispone de informes de actividad de cada estudiante, con gráficos y detalles sobre su paso por cada módulo.
- ✓ Responde a los estándares internacionales SCORM el cual permite importar y exportar los contenidos a otras plataformas.
- ✓ El profesor tiene control total sobre todas las opciones de un curso. Se puede elegir entre varios formatos de curso tales como semanal, por temas o el formato social, basado en debates.

#### **1.5.- Resultado del análisis de los sistemas que utilizan el estándar SCORM**

El estudio de los sistemas analizados, proyecta la característica en común de exportar o importar cursos y recursos, siguiendo las especificaciones del estándar SCORM. Sin embargo la forma en que estos



productos incorporan esta funcionalidad, no permite que el usuario aprecie el contenido del paquete SCORM hasta que no esté en la plataforma y no es posible reutilizar el código que estas plataformas poseen debido a la complejidad de ZERA. Producto de estas deficiencias la investigación presente se hace necesaria, con el fin de lograr que ZERA lime las dificultades antes señaladas y facilite el intercambio del usuario con los contenidos de los paquetes SCORM permitiendo el intercambio de OA con otros LMS.

## **1.6.- Aplicaciones cliente-servidor**

### **1.6.1.- Tecnologías del lado del servidor**

#### **1.6.1.1.- PHP (versión 5)**

##### **Características de PHP 5**

Fácil de usar: PHP5 es un lenguaje fácil de aprender. No es necesario hacer un estudio muy concienzudo de sus funciones para realizar programas sencillos que nos resuelvan la mayoría de los problemas diarios.

Multiplataforma: Se ejecuta en multitud de plataformas, Sistemas Operativos y servidores existentes. Es compatible con los tres servidores líderes del mercado: Apache, Microsoft Internet Information Server y Netscape Enterprise.

Sistemas Operativos compatibles con PHP5: MacOS X, GNU Linux, Solaris, Windows 98/Me, Windows NT/2000/XP/2003, entre otros.

Licencia Open Source: La licencia de Código Abierto implica que el código fuente de PHP5 es libre de ser descargado e inspeccionado por nosotros. La consecuencia principal es que el coste del producto en la mayoría de los casos es de 0 Euros. Tener el código fuente de PHP5 sirve, entre otras cosas, para poder hacer nuestro servidor a medida, es decir, podemos compilar el programa con las opciones que realmente utilicemos. (31)

#### **1.6.1.2.- Active Server Pages**

Active Server Pages (en adelante, ASP) es una tecnología dinámica funcionando del lado del servidor, algunas de las características de ASP son:

- ✓ ASP es totalmente gratuito para Microsoft Windows NT o Windows 95/98.
- ✓ El código ASP se puede mezclar con el código HTML en la misma página (no es necesario compilarlo por separado).



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

Las principales ventajas que ofrece ASP son:

- ✓ Permite acceder a bases de datos de una forma sencilla y rápida.
- ✓ Las páginas se generan dinámicamente mediante el código de scripts, (guiones).
- ✓ El código de script se ejecuta en el servidor, y no depende del navegador que se emplee.

### 1.6.1.3.- Apache

Es el servidor web hecho por excelencia, su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa. La historia de Apache se remonta a febrero de 1995, donde empieza el proyecto del grupo Apache, el cual está basado en el servidor Apache HTTPD. El servidor web Apache es uno de los mayores triunfos del software libre. En Diciembre de 1997 tenía una cuota de mercado cercana al 45% y en la actualidad (enero 2011) ya está por encima del 65%, según los estudios de Netcraft (32) que ya se han establecido como la referencia dentro del mercado de servidores web. Esta es la primera cifra que hace que cualquier responsable de la estrategia internet de una empresa tenga que tomar a Apache como el servidor de referencia.

Apache es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos. Entre sus características se destacan las siguientes:

- ✓ Multiplataforma
- ✓ Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona.
- ✓ Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.



- ✓ Se desarrolla de forma abierta
- ✓ Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP.

#### **1.6.1.4.- Microsoft Internet Information Server (versión 5.0)**

EL servidor Internet Information Server (en adelante IIS), fue desarrollado por Microsoft desde sus inicios se han desarrollado diferentes versiones que constituyen mejoras al mismo y presenta las siguientes características: (39)

- ✓ Asegurar los datos es mediante Secure Sockets Layer (SSL). Esto proporciona un método para transferir datos entre el cliente y el servidor de forma segura, permitiendo también que el servidor pueda comprobar al cliente antes de que inicie una sesión de usuario.
- ✓ Presenta autenticación implícita que permite a los administradores autenticar a los usuarios de forma segura a través de servidores de seguridad y proxy.
- ✓ Es capaz de impedir que aquellos usuarios con direcciones IP conocidas obtengan acceso no autorizado al servidor, permitiendo especificar la información apropiada en una lista de restricciones.

#### **1.6.2.- Tecnologías del lado del cliente**

##### **1.6.2.1.- JavaScript (versión 1.5)**

JavaScript es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web. Es un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado.

Es un lenguaje de programación bastante sencillo y pensado para hacer las cosas con rapidez, a veces con ligereza. Incluso las personas que no tengan una experiencia previa en la programación podrán aprender este lenguaje con facilidad y utilizarlo en toda su potencia con sólo un poco de práctica.

##### **1.6.2.3.- Hyper Text Markup Language (versión 4.0)**

El Lenguaje de Hipertexto Marcado (en adelante HTML) es el lenguaje con el que se escriben las páginas





web. Es un lenguaje de hipertexto, es decir, un lenguaje que permite escribir texto de forma estructurada, y que está compuesto por etiquetas, que marcan el inicio y el fin de cada elemento del documento.

Un documento hipertexto no sólo se compone de texto, puede contener imágenes, sonido, vídeos, entre otros, por lo que el resultado puede considerarse como un documento multimedia. Los navegadores se encargan de interpretar el código HTML de los documentos, y de mostrar a los usuarios las páginas web resultantes del código interpretado.

#### **1.6.2.4.- Cascading Style Sheets (versión 2.0)**

*Hojas de estilo en cascada (en adelante CSS)*, es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML.

Las ventajas de utilizar CSS son:

- ✓ Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.
- ✓ Los Navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web, con lo que aumenta considerablemente la accesibilidad.
- ✓ El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño.

### **1.7.- Análisis de diversas Metodologías de Desarrollo para la propuesta de solución del producto**

El objetivo primordial del proceso de desarrollo de software, es construir un software eficiente y eficaz, que cumpla los requisitos del cliente y elevar la calidad del software en todas las fases por las que este atraviesa, a través de una mayor transparencia y control sobre el proceso. A continuación se analizan las variantes de metodologías para el desarrollo de software conocidas por los investigadores, con el propósito de, posteriormente, tomar partida.

#### **1.7.1.- Rational Unified Process**

Rational Unified Process (en adelante RUP) es una Metodología Tradicional, posee un proceso formal: Provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que satisfaga los requerimientos de los usuarios finales (respetando cronograma y presupuesto). Puede ser adaptado y



extendido para satisfacer las necesidades de la organización que lo adopte. RUP posee un ciclo de vida que consta de 4 fases Concepción, Elaboración, Construcción y Transición. Contiene además 3 características fundamentales:

**Guiado por Casos de uso:** sirven para describir el comportamiento del sistema y elaborar los casos de prueba con los que se comprueba que el sistema desarrollado “hace lo que el cliente quiere”.

**Centrado en la arquitectura:** la arquitectura del sistema es la columna vertebral de todo el desarrollo del mismo. Cada iteración gira en torno a la misma, fortaleciendo y corrigiendo sus características.

**Iterativo e incremental:** en cada ciclo de iteración se produce una nueva versión del software. Utiliza UML como lenguaje de modelado y cuenta con varias fases de trabajo en las cuales se desarrolla una serie de flujos fundamentales del desarrollo del proyecto. (25)

### 1.7.2.- Extreme Programming

Extreme Programming (*Programación Extrema*, en adelante XP), es una Metodología Ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Se define como, especialmente adecuada, para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

### 1.7.3.- La Metodología Iconix

Iconix es dirigida por casos de uso, como RUP. Es un proceso pequeño y ligero como XP, pero no descarta las fases del análisis y el diseño. Este proceso usa el Lenguaje de Modelado Unificado (en adelante UML) y un seguimiento a los requisitos. Al ser dirigido por casos de uso, produce un resultado concreto, específico y casos de uso fácilmente entendibles, que un equipo de un proyecto puede usar para conducir el esfuerzo hacia un desarrollo real. El método cubre todas las fases del ciclo del desarrollo del software.

Lo original de la metodología es la definición de un proceso ágil para obtener la especificación de requerimientos y modelar el comportamiento de sistemas, utilizando el UML. Es una alternativa para la comunidad informática dedicada al desarrollo de sistemas de gestión pequeños y medianos, que favorece



la participación de los usuarios finales y la documentación de todo el proceso.

### 1.7.5.- Lenguaje de Modelado Unificado

Lenguaje Unificado de Modelado (en adelante UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software, y entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir.

Características del UML

- ✓ UML es un lenguaje de modelado de propósito general que pueden usar todos los modeladores. No tiene propietario y está basado en el común acuerdo de gran parte de la comunidad informática.
- ✓ UML no pretende ser un método de desarrollo completo. No incluye un proceso de desarrollo paso a paso. UML incluye todos los conceptos que se consideran necesarios para utilizar un proceso moderno iterativo, basado en construir una sólida arquitectura para resolver requisitos dirigidos por casos de uso.
- ✓ UML necesita ser lo suficientemente expresivo para manejar todos los conceptos que se originan en un sistema moderno, tales como la concurrencia y distribución, así como también los mecanismos de la ingeniería de software, como son la encapsulación y componentes.
- ✓ Es orientado a objetos, permitiéndole al programador que organice su programa de acuerdo con abstracciones de más alto nivel, siendo estas más cercanas a la forma de pensar de las personas (Mariño 2007).

## 1.8.- Análisis de Herramientas CASE para la propuesta de solución del producto

### 1.8.1.- Rational Rose Enterprise Edition

Rational Rose es una de las más poderosas herramientas de modelado visual para el análisis y diseño de sistemas basados en objetos. Se utiliza para modelar un sistema antes de proceder a construirlo. Cubre todo el ciclo de vida de un proyecto.

Características:

- ✓ Permite especificar, analizar, diseñar el sistema antes de codificarlo.
- ✓ Mantiene la consistencia de los modelos del sistema software.
- ✓ Generación de código a partir de los modelos.



- ✓ Ingeniería inversa (crear modelo a partir código).
- ✓ Genera documentación del sistema.
- ✓ Disponible en múltiples plataformas. (26)

### 1.8.2.- Visual Paradigm for UML (versión 6.3)

Es una potente herramienta CASE empleada para visualizar y diseñar elementos de software, para ello utiliza el lenguaje UML, proporciona a los desarrolladores una plataforma que les permite diseñar un producto con calidad de forma rápida. Facilita la interoperabilidad con otras herramientas como Rational Rose. Se integra con diversos IDE como: NetBeans, entre otros.

Características:

- ✓ Disponibilidad en múltiples plataformas.
- ✓ Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- ✓ Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- ✓ Capacidades de ingeniería directa e inversa.
- ✓ Licencia: gratuita y comercial.
- ✓ Soporta aplicaciones Web.
- ✓ Fácil de instalar y actualizar. (27)

### 1.9.- Para la construcción de bocetos o maquetas de los prototipos de interfaz de usuario, se selecciona la herramienta Balsamiq Mockups (versión 1.6.58)

Balsamiq Mockups es una aplicación en Flex / AIR (por lo que es posible instalarlo en Windows, Mac y Linux) que sirve para crear de un modo dinámico, sencillo, práctico de bocetos o Mockups de interfaces de usuario, no solo para páginas web, sino también para aplicaciones de escritorio, o cualquier cosa que requiera interfaz gráfica.

Razones de suma importancia para la selección del Balsamiq Mockups:

- ✓ Ya que usa componentes prefabricados, la tarea de incluirlos y si se requiere modificar su posición o proporciones.
- ✓ Los componentes pueden ser modificados desde un panel que se mantiene cerca del objeto sin interponerse entre el cliente y el proyecto.



- ✓ El boceto puede ser transportado en algún dispositivo de almacenamiento por lo que no hay posibilidad de perderlo. (28)

## **1.10.- Análisis de Entornos e IDEs de Desarrollo para la propuesta de solución del producto**

### **1.10.1.- Symfony**

Symfony es un framework PHP, que facilita el desarrollo de las aplicaciones web. Symfony se encarga de todos los aspectos comunes de las aplicaciones web, dejando que el programador se dedique a aportar valor desarrollando las características únicas de cada proyecto.

Características:

- ✓ Fácil de instalar y configurar.
- ✓ Funciona con todas las bases de datos comunes (MySQL, PostgreSQL, SQLite, Oracle, MS SQL Server).
- ✓ Preparado para aplicaciones empresariales, ya que se puede adaptar con facilidad a las políticas y arquitecturas propias de cada empresa u organización.
- ✓ Flexible hasta cualquier límite y extensible mediante un completo mecanismo de plug-in. (29)

### **1.10.2.- Zend Studio**

Zend Studio se ha diseñado para maximizar la productividad de los desarrolladores por lo que le permite desarrollar y mantener el código más rápido, resolver los problemas de aplicación de forma rápida y mejorar la colaboración en equipo.

El programa entero está escrito en Java, lo que a veces supone que no funcione tan rápido como otras aplicaciones de uso diario. Proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código.

### **1.10.3.- NetBeans (versión 6.9)**

El IDE NetBeans es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación. El IDE NetBeans es un producto libre y gratuito sin restricciones de uso.

Es un IDE de código abierto, posee una interfaz muy amigable e intuitiva, característica fundamental en un IDE, goza de herramientas para crear aplicaciones profesionales ya sean de escritorio, empresariales,



Web, móviles y aplicaciones SOA, además permite hacer diagramas UML.

#### **1.10.4.- Zend Framework**

Zend Framework (en adelante ZF) es un framework de código abierto para desarrollar aplicaciones web y servicios web con PHP 5. ZF es una implementación que usa código 100% orientado a objetos. La estructura de los componentes de ZF es algo único; cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado. A menudo se refiere a este tipo de diseño como "use-at-will" (uso a voluntad). (21)

El principal patrocinador del proyecto Zend Framework es Zend Technologies, pero muchas empresas han contribuido con componentes o características importantes para el marco. Empresas como Google, Microsoft y Strikelron se han asociado con Zend para proporcionar interfaces de servicios web y otras tecnologías que desean poner a disposición de los desarrolladores de Zend Framework. (21)

### **1.11.- Análisis de Sistemas Gestores de Base de Datos para la propuesta de solución del producto**

#### **1.11.1.- MySQL**

MySQL es un sistema de administración de bases de datos (Database Management System, en adelante DBMS) para bases de datos relacionales. Está licenciado bajo la GPL de la GNU.

MySQL como gestor de bases de datos es uno de los más usados en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Las principales características de este gestor de bases de datos son las siguientes:

- ✓ Gran portabilidad entre sistemas.
- ✓ Gestión de usuarios y contraseñas, manteniendo un muy buen nivel de seguridad en los datos.
- ✓ Se puede descargar su código fuente. Esto ha favorecido muy positivamente en su desarrollo y continuas actualizaciones.
- ✓ Fácil instalación y configuración. (30)

#### **1.11.2.- PostgreSQL (versión 8.4)**



Está considerado como el SGBD de código abierto más avanzado del mundo. Proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales tales como DB2 u Oracle.

PostgreSQL aproxima los datos a un modelo objeto-relacional y es capaz de manejar complejas reglas. Posee soporte para lenguajes procedurales internos (son aquellos en los cuales, el usuario instruye al sistema para que lleve a cabo una serie de operaciones en la base de datos, con el fin de calcular el resultado deseado y están fundamentados en la utilización de variables para almacenar valores y en la realización de operaciones con los datos almacenados), incluyendo un lenguaje nativo denominado PL/PGSQL.

### 1.12.- Metodología, Herramientas, IDE y lenguajes de programación seleccionados para el desarrollo del producto

El estudio realizado sobre las Metodologías para el desarrollo del software arrojó la conclusión de que la metodología Iconix se encuentra en una etapa experimental y gran parte de su información es encontrada en inglés, lo que frena el proceso de desarrollo; con respecto a MSF es un modelo de Microsoft que implica que se tienen que emplear herramientas solo de Microsoft. La metodología XP ofrece una mayor ventaja pero con ciertas limitaciones, ya que aún no ha sido explotada a gran escala como lo es RUP que posee alto soporte y herramientas integrales que nos guían a través del mismo, lo que facilita aplicar con mayor efectividad esta metodología, y permite aprovecharla al máximo.

Se decidió utilizar **RUP**, por el hecho de que es perfecta, tanto para el tipo de proyecto a desarrollar como por las condiciones de trabajo creadas. Existen razones más específicas las cuales planteamos a continuación:

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

Realizada la valoración de Herramientas CASE se decide utilizar **Visual Paradigm** por su compatibilidad con el Sistema Operativo Linux donde se implementará la solución propuesta, a diferencia del Rational



Rose Enterprise Edition que es soportado solamente por Sistema Operativos tales como Windows 2000, NT, XP y Windows 7, característica que nos hace desecharla a la hora de confeccionar los diagramas de clases web, que les pueden ser útil al programador. Visual Paradigm soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Utiliza UML como lenguaje de modelado y se integra con herramientas Java, tales como: Eclipse, JBuilder, NetBeans, entre otras. Además de ser una herramienta que genera la documentación del proyecto automáticamente en varios formatos como Web o PDF, soporta la realización de ingeniería tanto directa como inversa y permite control de versiones.

De igual forma la investigación realizada sobre los Entornos de Desarrollo provocó la selección de **Symfony** por el grado de complejidad que contiene el proyecto, donde se desarrollan aplicaciones en las que es importante la productividad de su desarrollo, la facilidad de su mantenimiento, su seguridad y la garantía de utilizar librerías probadas en algunos de los sitios web más grandes de Internet.

#### **Razones para utilizar Symfony:**

- ✓ Probado: ha sido probado con éxito durante años en varias aplicaciones gigantescas y en otros miles de sitios pequeños y medianos.
- ✓ Licencia: se publica bajo licencia MIT, con la que puedes desarrollar aplicaciones web comerciales, gratuitas y/o de software libre.
- ✓ Código: desde su primera versión Symfony ha sido creado sólo para PHP 5, para obtener el máximo rendimiento de PHP y aprovechar todas sus características.
- ✓ Documentado: es el framework mejor documentado, ya que ha publicado cinco libros gratuitos de calidad y siempre actualizados. Además, toda la documentación está traducida al español. (29)

Según los IDEs analizados se selecciona al **IDE NetBeans**, se integra con el framework Symfony a la perfección, dando la posibilidad de ejecutar comandos propios de Symfony (limpiar la cache, crear módulos, entre otros), desde el mismo IDE, se pueden usar herramientas de control de versiones sin necesidad de abrir una aplicación externa, brinda completamiento de código para todas las funciones que el Symfony provee, es altamente configurable, incluso, permite la creación de plantillas de códigos para facilitar, de esta forma, el trabajo a los desarrolladores, sin dudas es uno de los IDEs más completo en la actualidad y a la vez uno de los más sencillos de usar.





La investigación realizada sobre los SGBD proyecta la selección de **PostgreSQL** (versión 8.4), por poseer la característica de manejo con grandes volúmenes de datos, además de la durabilidad, siendo la propiedad que asegura que una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema. Se ajusta con el framework Symfony, corre en casi todos los principales sistemas operativos: Linux, Unix, BSDs, entre otros, cuenta con una documentación muy bien organizada, pública y libre, con comentarios de los propios usuarios. Comunidades muy activas, varias comunidades en español. Emplea la licencia BSD al contrario que la GPL que usa MySQL esta permite el uso del código fuente en software no libre.

Como servidor Web se empleará **Apache** por ser altamente confiable, estable, independiente de la plataforma, Apache es una tecnología gratuita de código fuente abierto, y desde que surgió en 1995 es el servidor web más usado en el mundo, como lenguaje de programación del lado del servidor se usará **PHP5** por las mejoras de rendimiento que este posee, es un lenguaje diseñado específicamente para desarrollar aplicaciones Web, dispone de muchos más recursos que ASP, además combina excelentemente con otras herramientas como es el caso del servidor Apache y el SGBD PostgreSQL y es un lenguaje conocido por el equipo de desarrollo.

Se utilizará como lenguaje de desarrollo del lado del cliente **JavaScript**, ya que todos los navegadores interpretan este código a diferencia de Visual Basic Script que solo es soportado por IE, es compatible con PHP y HTML.

### 1.13.- Conclusiones parciales

En el presente capítulo se realizó un estudio sobre el estado del arte del Objeto de Investigación y un análisis crítico del marco teórico conceptual, que permite un análisis más profundo de las herramientas y plataformas existentes, las pautas a seguir para lograr la correcta incorporación del estándar SCORM a la plataforma ZERA, así como las distintas plataformas que ya tienen incorporado este estándar para definir la forma correcta de incorporarlo a la plataforma ZERA. Además se seleccionaron las herramientas que se emplearán en el análisis, diseño e implementación de la solución del problema de investigación.





## 2.2.- Descripción del sistema propuesto

La extensión propuesta posee como principal objetivo, lograr el intercambio e interoperabilidad de recursos educativos, entre varios LMS que empleen el estándar SCORM. El componente cuenta con tres unidades principales: Exportar, Importar y Listar paquetes SCORM. La unidad de Exportar paquetes SCORM es iniciada a partir de la selección del capítulo, tema o subtema, debe permitir la selección y pre-visualización de los recursos necesarios a empaquetar por cada categoría existente (categorías predefinidas: imágenes, videos, audios, animaciones, simulaciones y páginas). Además de, una vez seleccionados los elementos sean incorporados a la lista de recursos a empaquetar, para luego ser formado y descargado para su posterior transportación.

La unidad de importar paquetes SCORM de igual forma es iniciada a partir de la selección del capítulo, tema o subtema, debe de permitir la selección del paquete SCORM y validarlo, con el objetivo de comprobar su compatibilidad según la estructura del estándar empleado. Una vez validado el paquete, el sistema deberá visualizar todos los recursos que contiene el mismo, así como ofrecer la posibilidad de pre-visualizarlos y seleccionarlos para luego ser importarlos a la plataforma.

La unidad de listar los paquetes SCORM es iniciada a partir del acceso a la aplicación *resources* módulo “*imsScorm*”, debe de permitir la visualización de todos los paquetes, los importados y exportados. Es necesaria además la visualización de los recursos que contiene el paquete seleccionado, con el fin de verificar si se desea eliminar o descargar.

## 2.3.- Requisitos del Software

Una vez conocidos todos los conceptos que rodean al objeto de estudio, se puede analizar qué debe hacer el sistema para que se cumplan los objetivos planteados al inicio de este trabajo. Para ello se enumeran, a través de requerimientos funcionales, las prestaciones que el sistema será capaz de brindar. Los requerimientos del software son las características y cualidades que el sistema debe tener. Estos se dividen en dos grupos, los funcionales y los no funcionales. Una vez definidos e identificados los mismos se tiene la visión general de lo que se quiere hacer en el sistema.

### 2.3.1.- Requisitos Funcionales.



Los Requisitos Funcionales (en adelante RF), se definen como las condiciones o capacidades que el sistema debe cumplir. Se desea que el sistema permita:

**RF 1: Exportar Paquete SCORM.**

RF 1.1 Buscar los recursos pertenecientes al Capítulo, Tema o Subtema que van a ser seleccionados.

RF 1.2 Mostrar los recursos que se van a incluir en el paquete.

RF 1.3 Seleccionar los recursos a exportar.

RF 1.4 Incluir los recursos al paquete.

RF 1.5 Pre-visualizar los recursos a seleccionar.

RF 1.6 Generar paquete SCORM a exportar.

RF 1.7 Descargar paquete SCORM creado.

**RF 2: Listar recursos seleccionados.**

RF 2.1 Ofrecer un resumen al usuario de la cantidad de recursos seleccionados por cada categoría.

RF 2.2 Mostrar metadatos asociados a cada recurso.

**RF 3: Importar Paquete SCORM.**

RF 3.1 Seleccionar el paquete SCORM a importar.

RF 3.2 Validar el paquete para su posterior importación a la plataforma.

RF 3.3 Mostrar los recursos que se van a incluir en la plataforma.

RF 3.4 Seleccionar los recursos a importar.

RF 3.5 Pre-visualizar los recursos a importar.

RF 3.6 Importar recursos que contiene el paquete SCORM a la base de datos de la plataforma.

**RF 4: Listar paquetes SCORM.**

RF 4.1 Visualizar paquetes importados o exportados a la plataforma.

RF 4.2 Mostrar los recursos que posee un paquete.

RF 4.3 Eliminar el o los paquetes existentes en la plataforma.

RF 4.4 Descargar el o los paquetes existentes en la plataforma.

RF 4.5 Realizar el filtrado de paquete.

**RF 5: Mostrar los recursos del paquete SCORM.**



RF 5.1 Visualizar la cantidad de recursos distribuidos por categoría que contiene el paquete.

**RF 6: Buscar paquetes SCORM.**

**RF 7: Buscar recursos en el paquete SCORM.**

**RF 8: Buscar recursos del tópico seleccionado.**

### **2.3.2.- Requisitos no Funcionales**

Los Requisitos No Funcionales (en adelante RNF) se definen como las cualidades o propiedades que el software debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Normalmente están vinculados a los requerimientos funcionales, es decir, una vez que se conozca lo que el sistema debe hacer, podemos determinar cómo ha de comportarse.

#### **RNF 1 Usabilidad.**

- ✓ Facilidad de uso y de comprensión.
- ✓ El sistema podrá ser usado por cualquier usuario que tenga conocimientos básicos de informática.

#### **RNF 2 Soporte (Restricciones de diseño e implementación)**

- ✓ Usar el lenguaje PHP 5.2 o superior
- ✓ Desarrollar bajo Symfony 1.4
- ✓ Utilizar como Servidor Apache 2.x.
- ✓ El sistema usará como motor de Base de Datos a PostgreSQL 8.2 o superior. Siendo necesario la elaboración de una copia de rescate de las tablas para no afectar la información almacenada antes de poner en total funcionamiento al sistema.

#### **RNF 3 Portabilidad.**

- ✓ El sistema debe de ser portable entre diferentes sistemas para poder ser usado en cualquier aplicación web que use el framework Symfony.

#### **RNF 4 Seguridad**

- ✓ El sistema brindará permisos para la ejecución, acceso o visualización de las funcionalidades al usuario autenticado en dependencia de los perfiles asignados a este último.

#### **RNF 5 Software**

- ✓ Máquina con sistemas Operativos Windows XP o superior, GNU Linux (Distribución Ubuntu, versión 6.06).



- ✓ Usar un navegador web como: Mozilla Firefox (versión 3.5.0 o superior), Internet Explorer (versión 7.0 o superior), Chrome (versión 7.0 o superior) y Opera (versión 10.0 o superior).

#### RNF 6 Hardware

- ✓ Las computadoras locales que brindarán el servicio cliente del sistema no deberán presentar potencias menores a las brindadas por una Pentium 4, con al menos 512 MB de RAM y la conexión de red deberá ser de 100 Mbps de velocidad mínima.

#### RNF 7 Interfaz.

- ✓ El sistema brindará una interfaz amigable para sus usuarios.
- ✓ El sistema aplicará normas de diseño que permitan la distinción visual entre los recursos seleccionados y los que no lo están.
- ✓ Ante la ocurrencia de un error, el sistema mostrará un mensaje de invalidez, lo que permite darle a conocer al usuario los problemas que se presentan.
- ✓ El sistema proporcionará claridad y buena organización de la información, permitiendo la interpretación correcta e inequívoca de esta.

### 2.4.- Definición de los actores del sistema

Actores	Justificación
Docente	Realiza la selección del paquete SCORM para la posterior importación y visualización de su contenido.
Gestor de Paquetes	Selecciona los recursos educativos existentes en la plataforma para posteriormente crear y exportar el paquete, realiza la selección del paquete SCORM para la posterior importación y visualización de su contenido, lista, elimina, descarga y visualiza los paquetes importados o exportados.
Administrador Central	Realiza las mismas operaciones que el Gestor de Paquetes y en la plataforma ZERA es el actor con más privilegios, pudiendo realizar todas las operaciones dentro de esta.
Editor	Realiza las mismas operaciones que el Gestor de Paquetes y en la plataforma ZERA este actor es un moderador teniendo acceso solamente a algunas funcionalidades.

**Tabla 2.1** Descripción de los Actores del Sistema

### 2.5.- Diagrama de Casos de Uso del Sistema

Un diagrama de casos de usos muestra la relación que existe entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa.

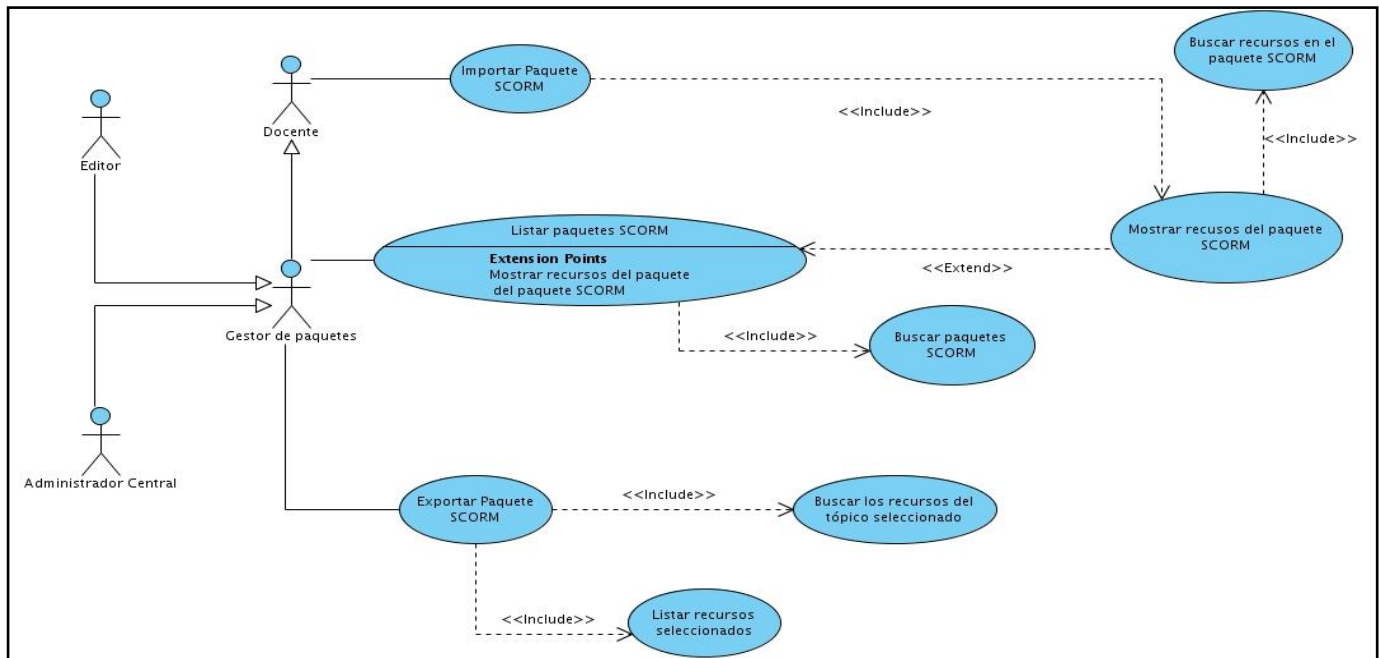


Figura 2.2 Diagrama de Casos de Uso del Sistema.

## 2.6.- Descripción de los Casos de Uso pertenecientes al DCUS.

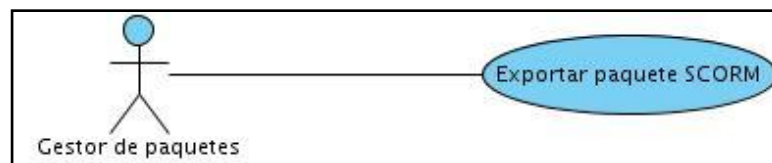


Figura 2.3 Diagrama del Casos de Uso del Sistema, Exportar paquete SCORM<sup>1</sup>.

CU-1	Exportar paquete SCORM
Actor	Editor, Administrador Central, Gestor de paquete (inicia).
Descripción	El caso de uso se inicia cuando el actor selecciona la opción exportar SCORM, que le permite realizar una acción sobre distintos tópicos (Capítulo, Tema o Subtema). El actor puede incluir, ver, seleccionar de manera aleatoria, además de seleccionar y desmarcar todos los recursos: imágenes,

<sup>1</sup>Para observar las demás descripciones ver Anexo I.



	videos, audios, simulaciones, animaciones y contenidos, que desea exportar, mediante los botones asociados a cada una de estas acciones. La acción ver se activa cuando se accede alguna de las categorías pertenecientes a alguno de los recursos existentes. Cuenta con un visor para cada tipo de recurso, se activa mediante la selección de un recurso en particular. Si el actor elige la opción de seleccionar todos, el sistema marcará todos los recursos pertenecientes a esa categoría, en el caso contrario el sistema desmarcará todos los recursos previamente marcados. Una vez el actor realiza la selección de los recursos de una categoría, el sistema los adiciona a la lista de recursos seleccionados. Cuenta además con la acción que le permite crear y generar el paquete SCORM.
Precondiciones	Debe haberse generado el escritorio de trabajo del usuario autenticado. Debe de seleccionarse el o los recursos a exportar.
Referencia	RF.1, RF.2, RF.8

**Tabla 2.2** Descripción del Caso de Uso del Sistema,  
Exportar paquete SCORM.

### 2.6.1.- Flujo Básico. CU Exportar paquete SCORM.

Acciones del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor selecciona la opción de Exportar Scorm para un Capítulo, Tema ó Subtema.	
	2. El sistema realiza una búsqueda de los recursos pertenecientes a estos tópicos. Por lo que se activa el <u>CU Buscar los recursos del tópico seleccionado.</u>
	3. Se levanta una ventana modal donde se cargan con todos los recursos asociados a estos tópicos, además de una breve descripción de cómo se debe de llevar a cabo el proceso seleccionado.
	4. Brinda la posibilidad de: <ul style="list-style-type: none"> <li>✓ Titular el paquete que se desea exportar.</li> <li>✓ Seleccionar los recursos de manera aleatoria.</li> <li>✓ Seleccionar todos los recursos mediante el botón Select All.</li> <li>✓ Desmarcar todos los recursos mediante el botón Deslect All.</li> <li>✓ Mostrar los recursos por cada categoría existente.</li> <li>✓ Ver listado de los recursos seleccionados. Se</li> </ul>





	<p>activa el <u>CU Listar recursos seleccionados</u>.</p> <p>Y permite:</p> <ul style="list-style-type: none"> <li>✓ Generar un paquete SCORM.</li> <li>✓ Descargar el paquete SCORM generado.</li> <li>✓ Cerrar las operaciones en cualquier momento.</li> </ul>
5. El actor selecciona la pestaña Summary e introduce el título del paquete.	
6. El actor selecciona una de las pestañas perteneciente a los recursos existentes.	
	<p>7. Brinda la posibilidad de realizar las acciones:</p> <ul style="list-style-type: none"> <li>✓ Seleccionar todos los recursos mediante el botón Select All.</li> <li>✓ Desmarcar todos los recursos mediante el botón Deslect All.</li> <li>✓ Seleccionar aleatoriamente los recursos.</li> <li>✓ Pre-visualizar el o los recursos.</li> </ul>
8. El actor selecciona la opción SelectAll.	
	9. El sistema selecciona de forma automática todos los recursos pertenecientes a la categoría seleccionada.
10. El actor selecciona la opción DeslectAll.	
	11. El sistema desmarca todos los recursos de la categoría escogida.
12. El actor opta por la pre-visualización de un recurso dando click en uno de ellos.	
	13. El sistema muestra en un visor de recurso, el que ha sido seleccionado, esto es válido para todos los recursos.
14. El actor selecciona la opción de Export Package SCORM.	
	<p>15. Brinda la posibilidad de:</p> <ul style="list-style-type: none"> <li>✓ Incluir los recursos seleccionados al paquete.</li> <li>✓ Crear el paquete SCORM, en el servidor (el paquete posee una extensión .ZIP). Este contiene carpetas que contienen los elementos seleccionados, además de una carpeta llamada Metadata que contiene los metadatos</li> </ul>



	asociados a cada recurso. Se crea también un XML llamado <code>imsmanifest.XML</code> el cual contiene la estructura definida para la creación de un paquete SCORM y otros ficheros que este paquete trae por defecto.
	16. Se crea un vínculo de descarga llamado <a href="#"><u>Click here to download.</u></a>
17. El actor selecciona la opción Click here to download.	
	18. Se levanta el cuadro de diálogo del navegador que le permite al usuario guardar el paquete creado en la dirección deseada.
19. El actor selecciona la opción Cerrar.	
	20. Brinda la posibilidad de cerrar la ventana modal.
	21. El caso de uso termina.

### Flujo alterno.

1.a El actor opta por marcar el checkbox de los recursos	
Acciones del actor	Respuesta del sistema
	1.a.1 Separa los elementos seleccionados de los que no lo están. 1.a.2 Se activa el <i>CU Listar recursos seleccionados.</i>
2.a El actor no selecciona el o los recursos a exportar	
Acciones del actor	Respuesta del sistema
	2.a.1 El sistema muestra un mensaje de información.
3.a El actor selecciona la flecha que se encuentra en una de las categorías	
Acciones del actor	Respuesta del sistema
	3.a.1 El sistema oculta o muestra los elementos existentes.

### 2.7.- Conclusiones parciales

La confección de este capítulo permitió identificar los CU necesarios para definir las funcionalidades básicas del sistema, teniendo en cuenta los requerimientos funcionales especificados y la acción entre actores y CU. Estas acciones se especifican en las descripciones de los CU definidos en el sistema. Con el desarrollo del presente capítulo se dejan creadas las bases para comenzar con la construcción de la propuesta, velando por el cumplimiento de todos los requerimientos y funcionalidades consideradas.

## CAPÍTULO III: ANÁLISIS Y DISEÑO

En el presente capítulo se realiza el análisis y diseño que sustenta la propuesta de solución del sistema a desarrollar. Se modelan además los diagramas de clases del análisis y los del diseño. Se define el patrón arquitectónico, y los patrones de diseño que regirán el proceso. Para garantizar el almacenamiento de la información, se modela la estructura de la base de datos.

### 3.1.- Estructura del paquete SCORM

El paquete SCORM es un fichero ZIP que contiene:

- ✓ Uno o varios ficheros que describen por medio de XML el índice de los contenidos de un paquete (estos ficheros se denominan MANIFEST) que, además, aporta información sobre la forma de presentarlos.
- ✓ El contenido (archivos físicos).

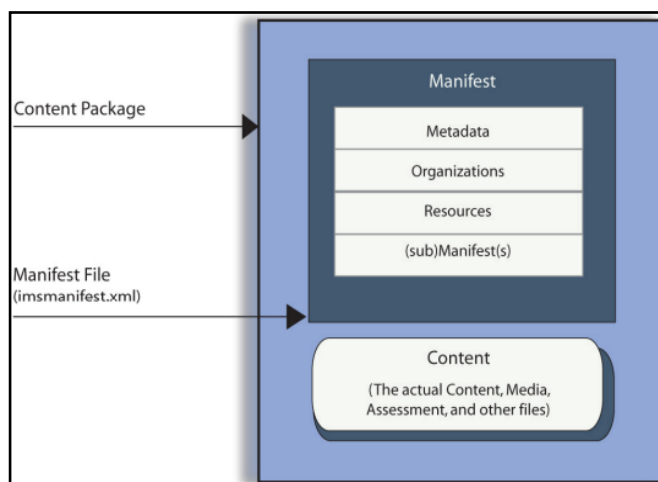


Figura 3.1 Estructura de un paquete SCORM.

#### Estructura del fichero *imsmanifest.xml*:

Para el mejor entendimiento de la estructura de este XML se usará la siguiente nomenclatura “<etiqueta XML> [cardinalidad del elemento]” (donde la cardinalidad del elemento se define: 1 es que debe de estar de forma obligatoria una vez; 0:1 no es obligatorio y de estar solo sería una vez; 1:many tiene que estar una vez de forma obligatoria y puede estar más de una vez).

<manifest>[1] Elemento raíz del manifiesto, posee como atributos *identifier*[1], *version*[0:1], *xml:base*[0:1].



**<metadata>[1]** Contenedor de los metadatos sobre todo el manifiesto.

**<schema>[1]** Define cual esquema es usado por el manifiesto. En SCORM 2004 siempre es “ADL SCORM”.

**<schemaversion[1]>** Define la versión del manifiesto.

**<lomradlcp:location[0:1]>** Contiene en el caso de ser *<lom>* los metadatos embebidos o una URL a un fichero externo en el caso de ser *<adlcp:location>*.

**<organizations[1]>** Contiene todas las formas en las que los contenidos del manifiesto se mostrarán. Generalmente solo se define una organización, múltiples organizaciones sirven para definir formas alternativas de presentar los contenidos, posee como atributos a *default[0:1]*.

**<organization[1:many]>** Contiene un elemento del contenido del manifiesto, posee como atributos a *identifier[1]*, *structure[0:1]*, *adlseq:objectivesGlobalToSystem[0:1]*.

**<title[1]>** El nombre de la organización, generalmente es usado como título del curso entero.

**<item[1:many]>** Cada *ítem* representa una actividad del curso. Un *ítem* puede tener otro *ítem* como hijo creando una estructura jerárquica, además tiene como atributo *identifier[1]*, *identifierref[0:1]*, *isvisible[0:1]*, *parameters[0:1]*.

**<title[1]>** Título de esta actividad.

**<item[0:many]>** Hijos de esta actividad para crear la estructura jerárquica.

**<metadata[0:1]>** Metadatos para describir esta actividad en particular, raramente usados.

**<lomradlcp:location[0:1]>** Contiene en el caso de ser *<lom>* los metadatos embebidos o una URL a un fichero externo en el caso de ser *<adlcp:location>*.



**<adlcp:timeLimitAction[0:1]>**Este elemento es el encargado de decirle al SCO lo que el autor del contenido quiere hacer cuando el estudiante excede el máximo de tiempo permitido.

**<adlcp:dataFromLMS[0:1]>**Datos que serán enviados al SCO viacmi.launch\_dataruntime data model element.

**<adlcp:completionThreshold[0:1]>**El umbral mínimo que el estudiante debe de realizar para considerar que ha completado la actividad, posee como atributos a *completedByMeasure[0:1]*, *minProgressMeasure[0:1]* y *progressWeight[0:1]*.

**<imsss:sequencing[0:1]>**Define las reglas de secuenciación para esta actividad.

**<adlnav:presentation[0:1]>**Contenedor para definir las reglas acerca de que controles de navegación deben de ser mostrado al usuario cuando esta actividad sea mostrada, solo es aplicable a las actividad que no posea una estructura jerárquica.

**<navigationInterface[0:1]>**Elemento contenedor adicional para especificar los controles de navegación disponibles.

**<hideLMSUI[0:many]>**Cada elemento hideLMSUI especifica un control de navegación que el autor desea que el LMS oculte cuando se entregue esta actividad.

**<metadata[0:1]>**Metadatos para describir esta organización en particular, usado solamente en cursos con múltiples organizaciones.

**<lomoradlcp:location[0:1]>**Contiene en el caso de ser *<lom>* los metadatos embebidos o una URL a un fichero externo en el caso de ser *<adlcp:location>*.

**<imsss:sequencing[0:1]>**Secuenciación de datos aplicables a la organización del nodo.

**<resources[1]>**Contiene una lista de todos los recursos (SCOs y Assets) utilizados en este manifiesto,



posee como atributo a *xml:base[0:1]*.

**<resource[1:many]>**Contenedor para un recurso, tiene como atributo *identifier[1]* (identificador del recurso, debe ser único en el manifiesto), *type[1]* (siempre será fijado a “webcontent”), *href[0:1]* (si el recurso es lanzable en este atributo se especifica la URL al fichero que se debe de lanzar primero), *xml:base[0:1]*, *adlcp:scormType[1]* (Tipo de recurso: sco o asset) .

**<metadata[0:1]>**Metadatos para describir este recurso en particular.

**<lomradlcp:location[0:1]>**Contiene en el caso de ser *<lom>* los metadatos embebidos o una URL a un fichero externo en el caso de ser *<adlcp:location>*.

**<file[1:many]>**Toda etiqueta resource debe de tener un fichero, tiene como atributo *href[1]* (URL donde se encuentra este fichero).

**<metadata[0:1]>**Metadatos para describir este fichero, raramente usado.

**<lomradlcp:location[0:1]>**Contiene en el caso de ser *<lom>* los metadatos embebidos o una URL a un fichero externo en el caso de ser *<adlcp:location>*.

**<dependency[0:many]>**Puntero a otro recurso que contiene ficheros necesarios por este recurso, usado para representar ficheros compartidos, tiene como atributo *identifierref[1]*.

**<imsss:sequencingCollection[0:1]>**Contiene reglas de secuenciación que pueden ser referenciadas desde otras actividades, muy útil para representar reglas de secuenciación comunes.

**<sequencing[0:many]>**Conjunto de reglas de secuenciación, tiene un atributo, *ID [0:1]*.

### 3.2.- Modelo de análisis

El modelo del análisis es una vista interna del sistema, estructurado por clases y paquetes estereotipados. Es utilizado fundamentalmente por los desarrolladores para comprender cómo debería darse forma al sistema, es decir, cómo debería ser diseñado e implementado. Sirve como una primera aproximación del diseño y su objetivo es comprender perfectamente los requisitos del software y no precisar cómo se

implementará la solución. Define realizaciones de casos de uso, y cada una de ellas representa el análisis de un caso de uso del modelo de casos de uso.

### 3.2.1.- Diagramas de clases del análisis

Un Diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa los objetos del mundo real, no de la implementación automatizada de estos objetos. Las clases del análisis se centran en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones del dominio. RUP propone clasificar a las mismas en interfaz, control y entidad. A continuación se presentan los diagramas de clases del análisis de los casos pertenecientes a la solución propuesta:

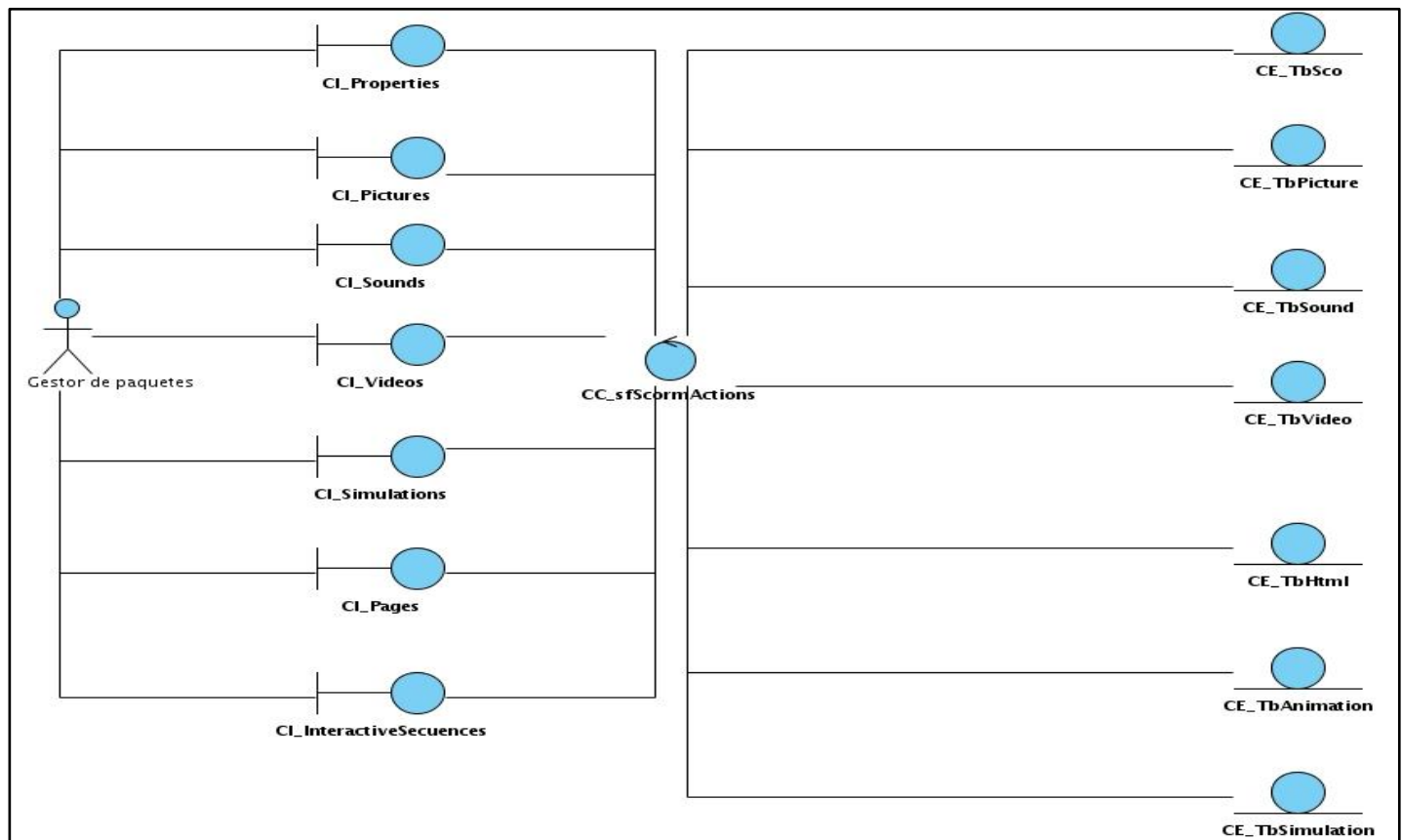


Figura 3.2 DCA Exportar paquete SCORM<sup>1</sup>.

### 3.3.- Paquetes del Sistema

Los paquetes reflejan la arquitectura de alto nivel de un sistema: su descomposición en subsistemas y sus dependencias. Una dependencia entre paquetes resume las dependencias entre los contenidos del paquete. Los mismos están organizados de manera funcional, siguiendo un cierto principio racional, tal como: funcionalidad común, implementación estrechamente relacionada, y un punto de vista común. Los paquetes pueden contener otros paquetes. A continuación se muestra el diagrama y la explicación de paquetes diseñados para el sistema:

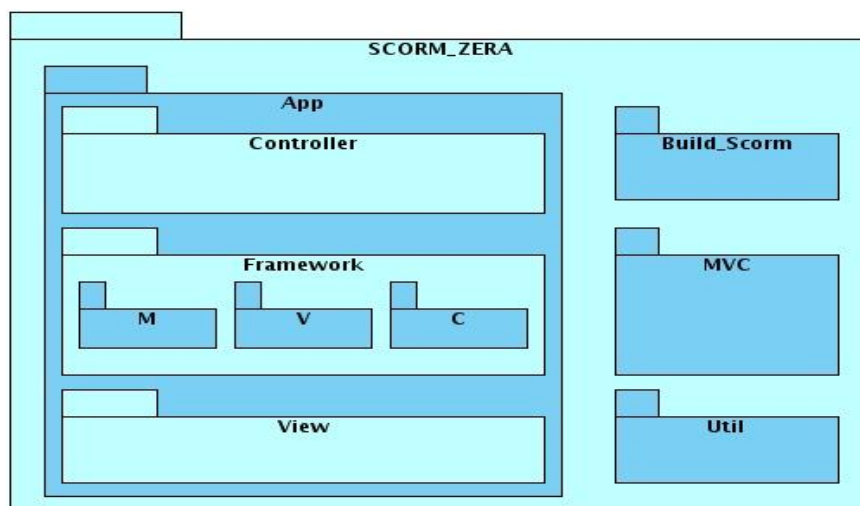


Figura 3.3 Diagrama de Paquetes del Sistema.

**App.Controller:** Contiene las clases controladoras encargadas de gestionar el negocio.

**App.View:** Contiene todas las interfaces que se utilizarán para elaborar la solución.

**App.Framework:** Contiene las clases relacionada con el Framework Symfony, separadas en tres paquetes que representan la forma en que el framework implementa el Patrón Arquitectónico *Modelo Vista Controlador*.

<sup>1</sup>Artefactos generados a partir del Análisis, Ver Anexos II.





**MVC:** Contiene la estructura del patrón Modelo Vista Controlador, así puede ser reutilizado libremente sin cambio alguno, es representado como un paquete genérico

**Build\_Scorm:** Contiene las clases que se emplean para crear la estructura del fichero *imsmanifest.xml*, implementa el patrón *Composite*.

**Util:** Contiene todas las clases auxiliares que van a ser utilizadas por las clases del paquete *Controller*, con el objetivo de complementar la implementación del negocio.

### 3.4.- Modelo de diseño

Para la confección de los diagramas de clases del diseño se emplearon diferentes patrones de diseño, estos son soluciones a problemas específicos, son considerados de esta manera siempre y cuando se halla comprobado su efectividad, flexibilidad y reusabilidad. Para realizar correctamente los diagramas de clases del diseño se consideran como más significativos los siguientes:

#### Modelo - Vista - Controlador en Symfony

Modelo Vista Controlador (en los adelante MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.

Symfony está basado en un patrón clásico del diseño web conocido como arquitectura MVC, que está formado por tres niveles:

- ✓ El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio. Incorpora la capa del dominio y persistencia, es la encargada de guardar los datos en un medio persistente (ya sea una base de datos, un archivo de texto, XML, registro, etc). En el modelo es donde se hace el levantamiento de todos los objetos que el sistema debe de utilizar, es el proveedor de los recursos
- ✓ La vista transforma el modelo en una página web que permite al usuario interactuar con ella. Se encarga de presentar la interfaz al usuario, en sistemas web, esto es típicamente HTML. En la vista solo se deben de hacer operaciones simples.
- ✓ El controlador se encarga de procesar las interacciones del usuario y realiza los cambios



apropiados en el modelo o en la vista. Es el que escucha los cambios en la vista y se los envía al modelo, el cual le regresa los datos a la vista, es un ciclo donde cada acción del usuario causa que se inicie de nuevo un nuevo ciclo.

Las principales ventajas de hacer uso del patrón MVC son:

- ✓ La separación del Modelo de la Vista, es decir, separar los datos de la representación visual de los mismos.
- ✓ Es mucho más sencillo agregar múltiples representaciones de los mismos datos o información.
- ✓ Facilita agregar nuevos tipos de datos según sea requerido por la aplicación ya que son independientes del funcionamiento de las otras capas.
- ✓ Crea independencia de funcionamiento.
- ✓ Facilita el mantenimiento en caso de errores.
- ✓ Ofrece maneras más sencillas para probar el correcto funcionamiento del sistema.

### **Acciones y controlador frontal**

Normalmente el patrón MVC se encargaba de realizar muchas tareas, para solucionar esto Symfony incluye un controlador frontal para aligerar el trabajo. Entre las tareas de este controlador se encuentran el manejo de las peticiones del usuario, el manejo de la seguridad, cargar la configuración de la aplicación y otras tareas similares. Por este motivo, el controlador normalmente se divide en un controlador frontal, que es único para cada aplicación, y las acciones, que incluyen el código específico del controlador de cada página.

Una de las principales ventajas de utilizar un controlador frontal es que ofrece un punto de entrada único para toda la aplicación. Así, en caso de que sea necesario impedir el acceso a la aplicación, solamente es necesario editar el script correspondiente al controlador frontal. Si la aplicación no dispone de controlador frontal, se debería modificar cada uno de los controladores.

### **Decorator**

Responde a la necesidad de añadir dinámicamente funcionalidad a un objeto. Esto nos permite no tener que crear sucesivas clases que hereden de la primera incorporando la nueva funcionalidad, sino otras que la implementan y se asocian a la primera. Aplicaciones:



- ✓ Añadir objetos individuales de forma dinámica y transparente.
- ✓ Responsabilidades de un objeto pueden ser retiradas.
- ✓ Cuando la extensión mediante la herencia no es viable.
- ✓ Hay una necesidad de extender la funcionalidad de una clase, pero no hay razones para extenderlo a través de la herencia.
- ✓ Hay la necesidad de extender dinámicamente la funcionalidad de un objeto y quizás quitar la funcionalidad extendida.

### Singleton

Está diseñado para restringir la creación de objetos pertenecientes a una clase, forzando a que solo se pueda crear una única instancia, de ahí su nombre *singleton* (instancia única).

### Composite

Sirve para construir objetos complejos a partir de otros más simples y similares entre sí, gracias a la composición recursiva y a una estructura en forma de árbol. Esto simplifica el tratamiento de los objetos creados, ya que al poseer todos ellos una interfaz común, se tratan todos de la misma manera.

#### 3.4.1.- Diagrama de clases del diseño con estereotipos Web.

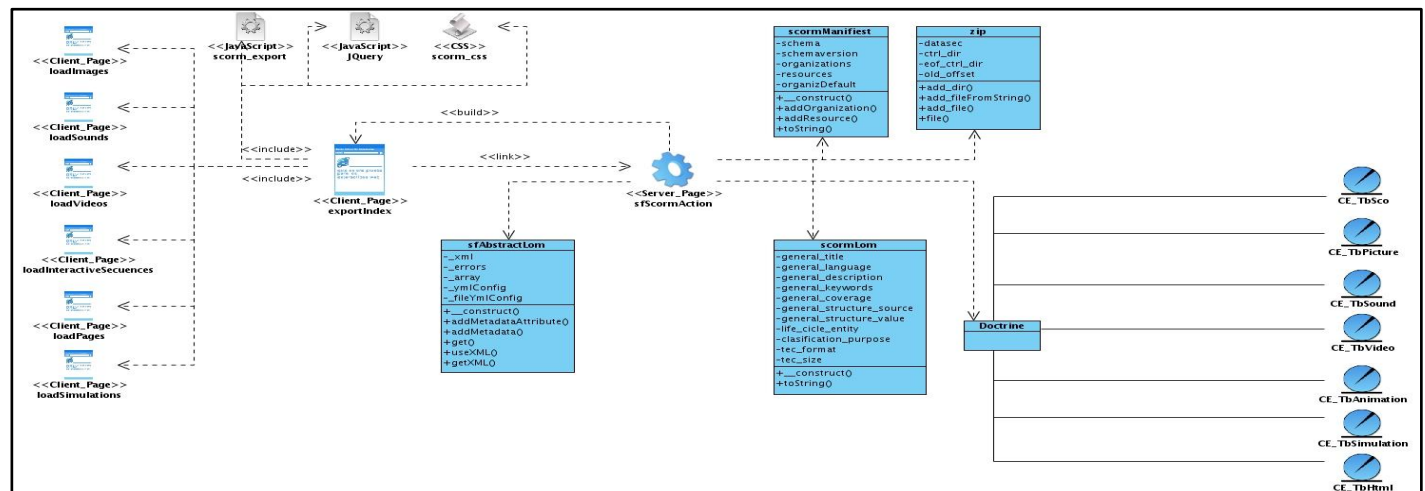


Figura 3.5 Diagrama de estereotipo web del CU Exportar paquete SCORM<sup>1</sup>.

### 3.4.2.- Diseño de las Interfaces de Usuario

A continuación se muestran los bocetos diseñados, elementos claves para la implementación del producto.

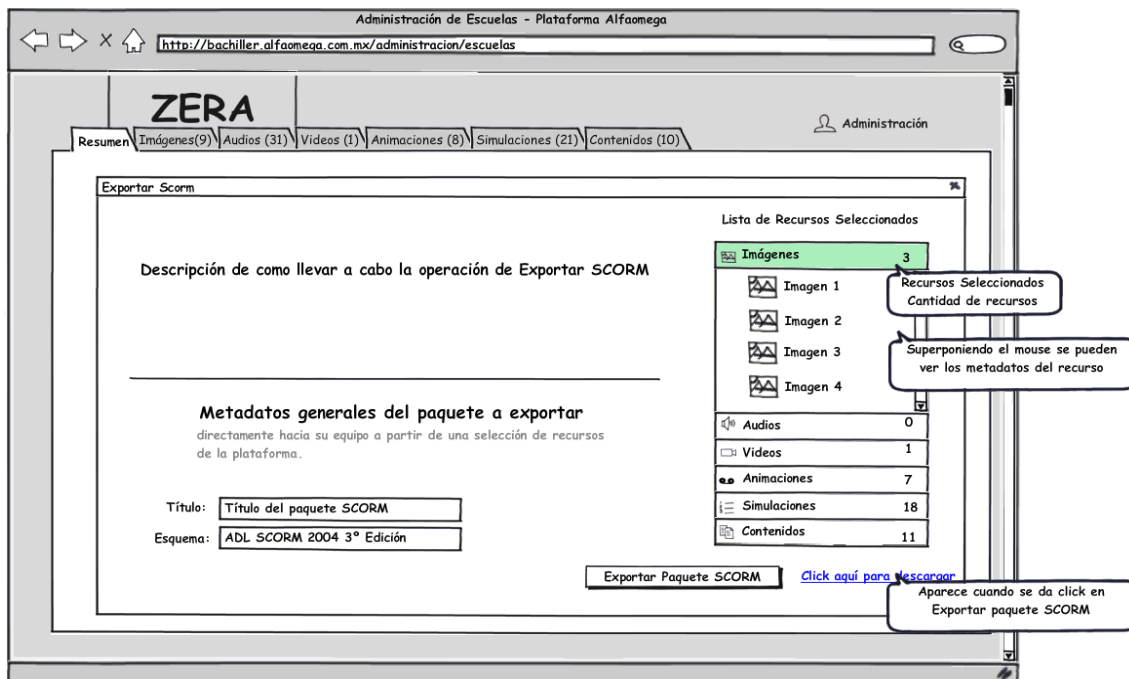


Figura 3.6 IU Resumen. (Exportar SCORM)<sup>2</sup>

### 3.5.- Diseño de la base de datos

Los modelos de datos aportan la base conceptual para diseñar aplicaciones que hacen un uso intensivo de datos. Un modelo de datos describe las representaciones lógicas y físicas de datos persistentes utilizados por la aplicación. Es usado para describir la representación lógica y física de la información persistente manejada por el sistema. A continuación se presentan las entidades y campos fundamentales relevantes del modelo de datos propuesto.

<sup>1</sup> Diagramas de Estereotipo Web, ver Anexo III

<sup>2</sup> Interfaces de Usuario, ver Anexo IV



### 3.5.1.- Descripción de las entidades relevantes.

tb_sco		
Descripción: Contiene los objetos de aprendizaje, ejemplo un sonido, video, etc.		
Atributo	Tipo	Descripción
metadata	TEXT	TEXT representa los campos descripción, contenido de página, contenido de palabra calientes y cualquier otro campo que contenga un texto con un máximo de 65535 caracteres.

Tabla 3.1 Descripción de la tabla tb\_sco.

tb_simulation		
Descripción: Agrupa la información existente de una media simulación.		
Atributo	Tipo	Descripción
simulation	IMAGE/LONG BINARY	IMAGE/LONG BINARY representa los campos imagen, sonido, video y cualquier otro campo que contenga un dato cuyo tamaño puede estar comprendido desde cero hasta un máximo de 2.14 gigabytes.

Tabla 3.2 Descripción de la tabla tb\_simulation.

tb_html		
Descripción: Almacena los datos de la evaluación del estudiante		
Atributo	Tipo	Descripción
html	TEXT	Representa los campos descripción, contenido de página, contenido de palabra calientes y cualquier otro campo que contenga un texto con un máximo de 65535 caracteres.

Tabla 3.3 Descripción de la tabla tb\_html.

tb_video		
Descripción: Agrupa la información existente de una media video.		
Atributo	Tipo	Descripción
video	VARCHAR(1000)	Atributo que permite guardar una media video.



url	TEXT	Atributo que permite guardar la url de donde fue tomada el video.
acces_date	DATE	Atributo que permite guardar la fecha en la que se guardó el video
reference_picture	IMAGE/LONG BINARY	IMAGE/LONG BINARY representa los campos imagen, sonido, video y cualquier otro campo que contenga un dato cuyo tamaño puede estar comprendido desde cero hasta un máximo de 2.14 gigabytes.

**Tabla 3.4** Descripción de la tabla tb\_video.

<b>tb_animation</b>		
<b>Descripción:</b> Agrupa la información existente de una media animación.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
url	TEXT	TEXT representa los campos descripción, contenido de página, contenido de palabra calientes y cualquier otro campo que contenga un texto con un máximo de 65535 caracteres.
acces_date	TIMESTAMP/DATE	TIMESTAMP representa los tipos de datos donde es necesario registrar además de la fecha la hora.
reference_picture	IMAGE/LONG BINARY	TEXT representa los campos descripción, contenido de página, contenido de palabra calientes y cualquier otro campo que contenga un texto con un máximo de 65535 caracteres.
animation	IMAGE/LONG BINARY	Atributo que permite guardar una media animación.

**Tabla 3.5** Descripción de la tabla tb\_animation.

<b>tb_sound</b>		
<b>Descripción:</b> Agrupa la información existente de una media audio.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
sound	VARCHAR(1000)	Atributo que permite guardar una media, sonido.
url	TEXT	Atributo que permite guardar la url de donde fue guardado el sonido.
acces_date	DATE	Atributo que permite guardar la fecha en la que se guardó el sonido

**Tabla 3.6** Descripción de la tabla tb\_sound.



<b>tb_applet</b>		
<b>Descripción:</b> Agrupa la información existente de una media applet.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
applet	IMAGE/LONG BINARY	Atributo que permite guardar una media applet.
html_tag	TEXT	TEXT representa los campos descripción, contenido de página, contenido de palabra calientes y cualquier otro campo que contenga un texto con un máximo de 65535 caracteres.
reference_picture	IMAGE/LONG BINARY	Representa los campos imagen, sonido, video y cualquier otro campo que contenga un dato cuyo tamaño puede estar comprendido desde cero hasta un máximo de 2.14 gigabytes.

**Tabla 3.7** Descripción de la tabla tb\_applet.

<b>tb_picture</b>		
<b>Descripción:</b> Agrupa la información existente de una media imagen.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
big_picture	IMAGE/LONG BINARY	Atributo que permite guardar una media imagen.
small_picture	TEXT	Representa los campos descripción, contenido de página, contenido de palabra calientes y cualquier otro campo que contenga un texto con un máximo de 65535 caracteres.
url	TEXT	Atributo que permite guardar la url de donde se guardó la imagen.
access_date	DATE	Atributo que permite guardar la fecha en la que se guardó la imagen.
width	VARCHAR(10)	VARCHAR (10) representa los campos tipo de ejercicio, grupo, respuesta y cualquier otro campo que contenga una cadena con longitud de caracteres de 10.
high	VARCHAR (10)	Representa los campos tipo de ejercicio, grupo, respuesta y cualquier otro campo que contenga una cadena con longitud de caracteres de 10.

**Tabla 3.8** Descripción de la tabla tb\_picture

<b>tb_general</b>
-------------------

Descripción: Almacena las palabras claves de cada OA.		
Atributo	Tipo	Descripción
title	VARCHAR (500)	Representa el campo enunciado y cualquier otro campo que contenga una cadena con longitud de caracteres de 500.
description	VARCHAR (1000)	Representa el campo ámbito y cualquier otro campo que contenga una cadena con longitud de caracteres de 1000.
coverage	VARCHAR (128)	Representa los campos username, password, nombre de un área de interés y cualquier otro campo que contenga una cadena con longitud de caracteres de 128.
structure	VARCHAR (50)	Representa los campos nombre del estado, nombre del rol, nombre de configuración y cualquier otro campo que contenga una cadena con longitud de caracteres de 50.

Tabla 3.9 Descripción de la tabla tb\_general.

A continuación se presenta el diagrama de clases que se determinaron como persistentes en el dominio de la solución propuesta:

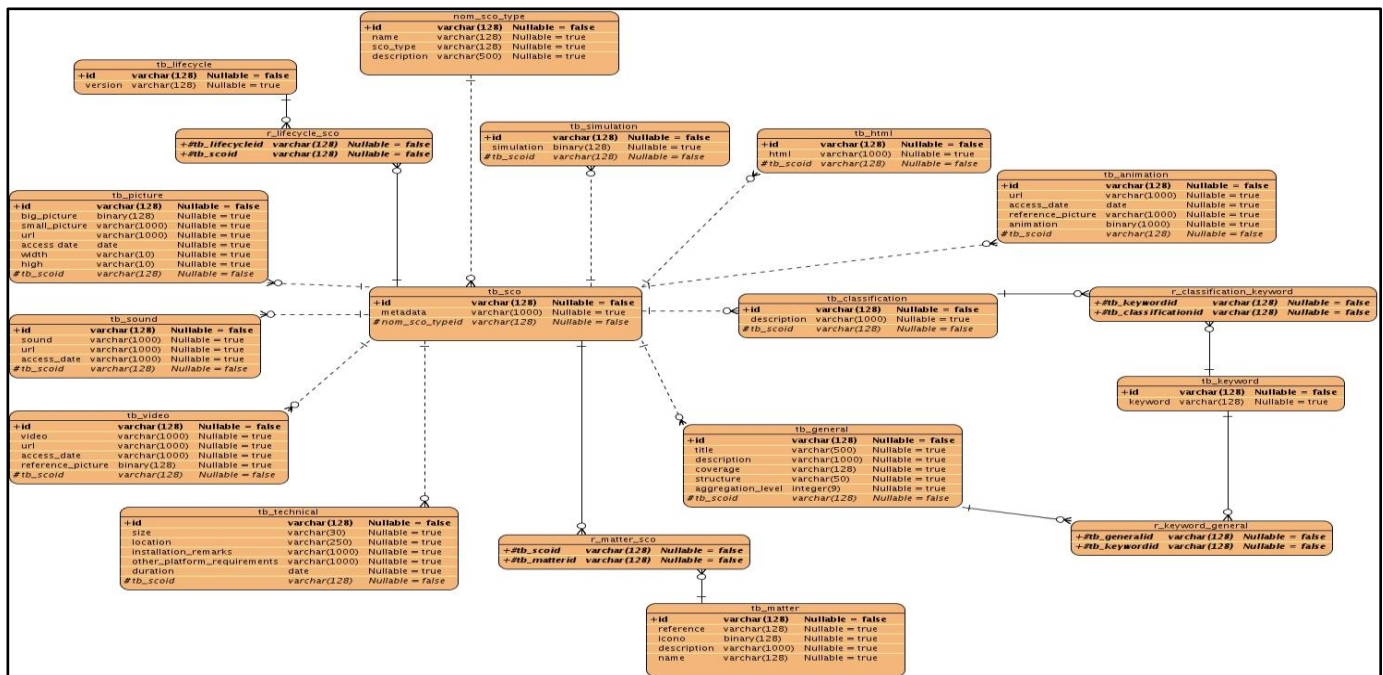


Figura 3.7 Modelo de datos de la solución propuesta.





### **3.6.- Conclusiones parciales**

En este capítulo se describe el diagrama de clases del sistema, las interfaces de usuarios, los diagramas de estereotipo web y diagrama de paquetes, que ayudarán al programador a desarrollar el modelo de implementación de la solución propuesta.



## CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBA

En el capítulo se lleva a cabo la implementación del sistema a través de los lenguajes de programación y las metodologías seleccionados. Se arriban a conclusiones sobre las verificaciones efectuadas de las pruebas realizadas, para comprobar que el sistema brinde las funcionalidades acordadas.

### 4.1.- Estándares de código

Los estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código.

#### 4.1.1.- Nomenclatura de las clases

Los nombres de las clases serán en idioma Inglés, comienzan con la primera palabra en minúscula, en caso de que sea un nombre compuesto se empleará notación “*lowerCamelCase*”, esta notación solo la incumplirán las siguientes clases: *XML*, *XML\_TAG*, *PclZip*, *Zip* debido a que son clases externas utilizadas para implementar la solución.

#### 4.1.2.- Nomenclatura de las funciones

El nombre a emplear para las funciones serán en idioma Inglés, se escribe con la primera palabra en minúscula, en caso de que sea un nombre compuesto se empleará notación “*lowerCamelCase*” y con sólo leerlo se reconoce el propósito de la misma, en el caso de las funciones *JavaScript* empezarán con el prefijo “*plug\_scom\_*” luego el nombre de la función utilizando la notación “*UpperCamelCase*” Ej: “*plug\_scom\_UpperCamelCase*”.

### 4.2.- Descripción de las principales clases utilizadas.

Las clases muestran un mayor nivel de detalle y permiten el rápido mantenimiento al código, se conciben para satisfacer los requisitos funcionales y no funcionales, teniendo en consideración la tecnología en la cual se implementará la solución.

**scormManifest:** Clase controladora que se encarga de generar todo el contenido del fichero *imsmanifest.xml* que va a ser exportado, para esto se auxilia de las clases *scormOrganization*, *scormResource* y *scormLom*.



**scormOrganization:** Clase encargada de generar el bloque *organizations* con todo su contenido siguiendo las normas el estándar SCORM para esto se auxilia de las clases *scormItem* y *scormLom*.

**scormItem:** Clase encargada de generar el bloque *ítem* del paquete SCORM.

**scormResource:** Clase encargada de generar el bloque *resources* el cual es de obligatoria inclusión dentro de la estructura de SCORM para esto se auxilia de las clases *scormFile* y *scormLom*.

**scormFile:** Clase encargada de generar el bloque *file* dentro de cada etiqueta *resource*.

**scormLom:** Clase que se encarga de generar los metadatos necesarios para cada objeto del paquete SCORM.

**readScorm:** Clase controladora que se encarga de leer el contenido del fichero *imsmanifest.xml*, así como validarlo y obtener todos los recursos, para esto se auxilia de la clase *XML*.

**readLom:** Clase controladora que se encarga de leer los metadatos de cada objeto y facilitar la obtención de estos para su posterior manipulación, para esto se auxilia de la clase *XML*.

**XML:** Clase auxiliar que se encarga de parsear un XML y convertirlo en un arreglo, para esto se auxilia de la clase *XML\_TAG*.

**XML\_TAG:** Clase auxiliar que representa un *tag* de un XML facilitando así la obtención de su valor.

**sfAbstractLom:** Clase encargada de leer el XML existente en la base de datos el cual representa los metadatos asociados a un OA y facilita la obtención de estos metadatos.

**PclZip:** Clase utilizada en el proceso de descompresión del paquete SCORM ya que presenta métodos muy útiles para descomprimir un fichero, incluso permite extraer un fichero específico sin descomprimir totalmente el paquete.

**Zip:** Clase utilizada en el proceso de crear / comprimir el paquete SCORM ya que presenta métodos muy útiles y fáciles de usar para añadir ficheros a un *.zip*.

**sfScormActions:** Clase controladora encargada de manejar todos los procesos de exportación e importación de los paquetes SCORM.

**imsScormActions:** Clase controladora encargada de visualizar, descargar y eliminar los paquetes importados o exportados a la plataforma.

#### 4.3.- Proceso de Implementación.

Para llevar a cabo la implementación de la solución se creó la extensión llamándola: *sfScormPlugin*; se comenzó por las interfaces de usuarios realizadas en el diseño, quedando estas de la siguiente forma:

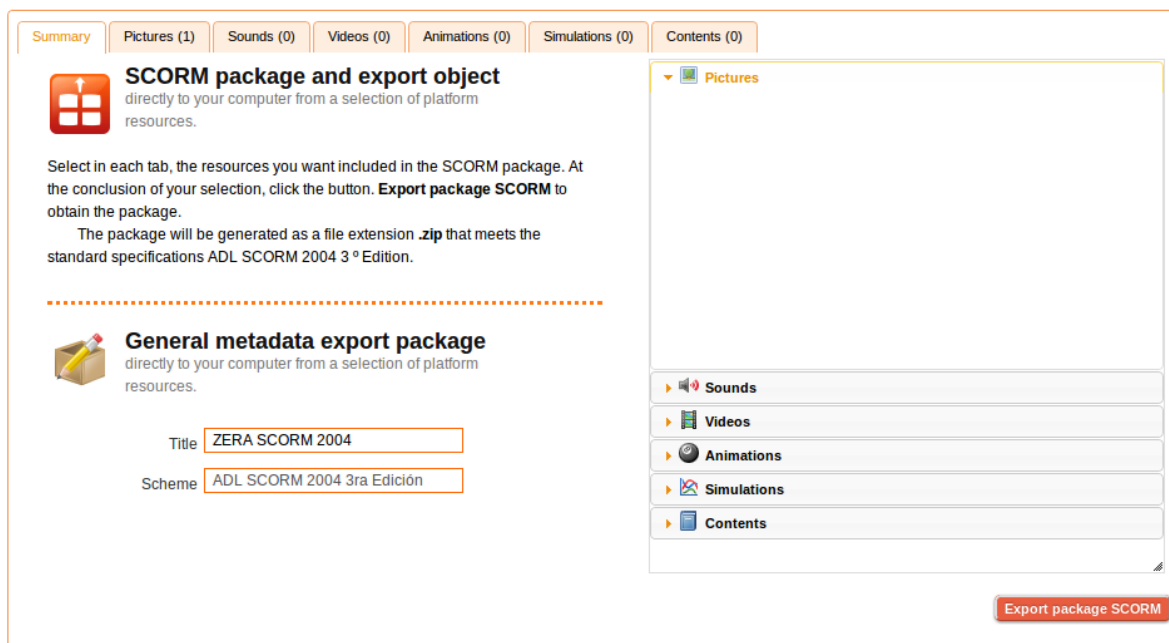


Figura 4.1 IU Resumen. (Sección Exportar SCORM)<sup>3</sup>

Al terminar con esta primera etapa se procedió a la implementación del negocio comenzando por la realización de las clases del diseño enfatizando en las más importantes: *scormManifest*, *readScorm*, *readLom*, *scormLom*; se continuó con la implementación realizando los métodos necesarios para obtener todos los recursos de la BD relacionados con el capítulo, tema o subtema seleccionado, a medida que fue avanzando en el desarrollo se fue construyendo el fichero *JavaScript* “*scorm\_js.js*” con las funciones necesarias para construir la solución, así como el fichero “*scorm\_css.css*” contenedor de todos los estilos utilizados.

Continuando con el proceso se construyó una nueva tabla en la BD “*TbScormList*” para tener registrado todos los paquetes que se exporten o importen a la Plataforma; se creó con la ayuda de la tarea de *Symfony* “*generate-admin*” un módulo en ZERA llamado “*imsScorm*” para mostrar los datos almacenados en la tabla *TbScormList*.

<sup>3</sup> Ver Anexo IV



Finalmente se integró la extensión con ZERA y se le hicieron las pruebas necesarias corrigiéndose los errores detectados.

#### **4.4.- Descripción de los algoritmos más importantes.**

##### **4.4.1.- Función executeUploadScorm (sfWebRequest \$request)**

Función utilizada en el Importar SCORM es la encargada de validar y verificar que el archivo subido sea un *.zip* luego de pasar esta etapa pasa a comprobar que dentro de este exista el fichero "*imsmanifest.xml*" (fichero que debe estar en la raíz del *.zip* de forma obligatoria de lo contrario no cumpliría con el estándar de empaquetamiento *IMS Content Package*) y que este fichero cumpla con la estructura del estándar SCORM 2004 3ra Edición, si cumple con todas las verificaciones necesarias permitirá al usuario continuara la siguiente etapa.

##### **4.4.2.- Función executeReadTotal (sfWebRequest \$request)**

Función utilizada en el Importar SCORM encargada de leer con la ayuda de la clase "*readScorm*" el fichero "*imsmanifest.xml*" y separar todos los recursos según su tipo, los tipos definidos son:

- ✓ Imágenes, solo las extensiones *jpg*, *png* y *gif*.
- ✓ Audios, solo las extensiones *mp3*.
- ✓ Videos, solo la extensión *flv*.
- ✓ Simulaciones, con extensión *jar*.
- ✓ Animaciones, solo la extensión *swf*.
- ✓ Contenidos o Páginas, con las extensiones *html* y *htm*.

##### **4.4.3.- Función executeBuildScorm (sfWebRequest \$request)**

Función utilizada en el Exportar SCORM para construir el link de descarga del paquete a exportar para esto se auxilia de la función "*buildZipInServer*".

##### **4.4.4.- Función buildZipInServer (sfWebRequest \$request, \$user, \$options)**

Función utilizada en el Exportar SCORM encargada de generar el paquete utilizando la lista de recursos seleccionados, para esto busca el recurso en la Base de Datos y es capaz de crear el "*imsmanifest.xml*", genera el archivo *.zip* y guarda dentro de este todos los recursos separados por carpetas según el tipo

incluyendo los metadatos relacionados a este, se auxilia de las clases "*scormManifest*", "*scormLom*" y "*sfAbstractLom*".

#### 4.5.- Modelo de Implementación

En la implementación se parte del resultado del diseño y se implementa el sistema en término de componentes, el propósito principal de la implementación es desarrollar la arquitectura y el sistema como un todo. Los artefactos que complementan la implementación son el *Diagrama de Despliegue* y *Componentes*.

##### 4.5.2.- Diagrama de Despliegue

Es un tipo de diagrama que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. Los elementos que lo componen son los siguientes:

- ✓ Procesadores: nodos que tienen capacidad de procesamiento (computadoras por lo general).
- ✓ Dispositivos: nodos que no tienen capacidad de procesamiento.
- ✓ Protocolos: estándares que deben estar implementados en la red, para efectuar la comunicación entre los diferentes elementos.

A continuación se muestra el Diagrama de Despliegue confeccionado:

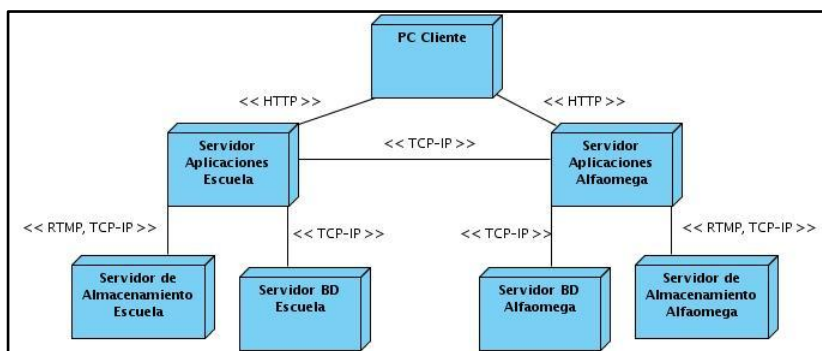


Figura 4.1 Diagrama de Despliegue del Sistema.

##### 4.5.3.- Diagrama de Componentes

Este diagrama describe las unidades físicas del sistema y sus relaciones. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser archivos simples, paquetes, librerías cargadas dinámicamente, etc. Para lograr una mejor

comprensión de este artefacto se decide recurrir a paquetes para agrupar los elementos que van a ser empleados en la implementación.

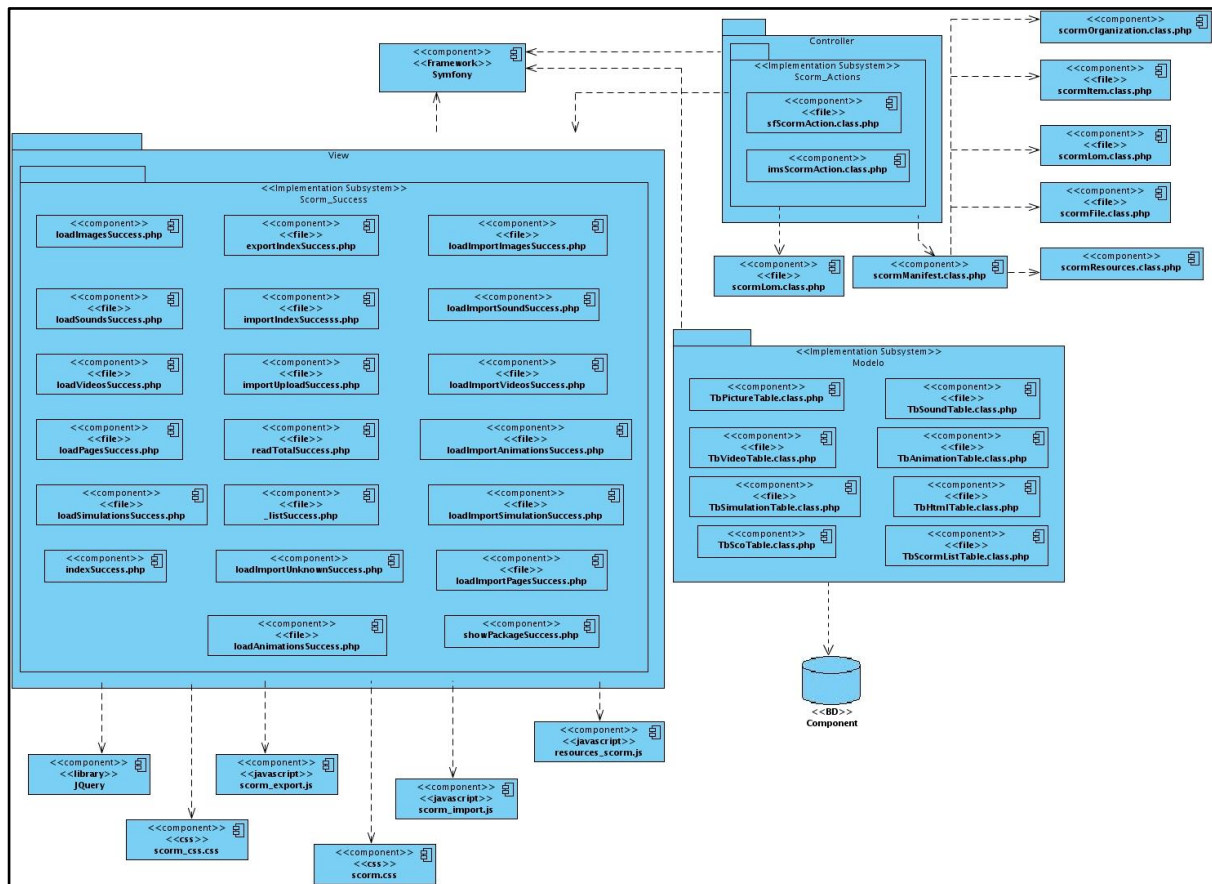


Figura 4.2 Diagrama de Componentes del Sistema.

#### 4.6.- Pruebas del Software

En todas las fases del desarrollo del proyecto hay que probar el software que se va construyendo, la etapa de prueba es tan o más importante que todas las realizadas hasta el momento puesto que en ella se refleja la calidad con que ha sido llevada a cabo la construcción del sistema. Las pruebas de software consisten en una serie de actividades en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados, realizando una evaluación del sistema o del componente probado. Su principal objetivo es evaluar la calidad del producto. Además de ser un elemento crítico para la garantía de la calidad del software.



La Prueba es aplicada para diferentes tipos de objetivos, en diferentes escenarios o niveles de trabajo.

Existen niveles de pruebas, las que a continuación se mencionan:

- ✓ Prueba de desarrollador.
- ✓ Prueba independiente.
- ✓ Prueba de Unidad.
- ✓ Prueba de Integración.
- ✓ Prueba de sistema.
- ✓ Prueba de aceptación.

#### 4.6.1.- Pruebas de Caja Negra.

Para la comprobación del producto implementado se realizarán las Pruebas de Integración, se dividen en Estructurales y Funcionales. Existen métodos de prueba como son los de Caja Blanca (en adelante CB) y Caja Negra (en adelante CN), este último se centran principalmente en los requisitos funcionales del *software*. Permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Los casos de prueba del método de CN pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa se mantiene. A continuación se muestran algunos de los Casos de Prueba, el resto se encuentran registrados en la documentación del proyecto Alfaomega<sup>4</sup>.

#### Caso de Prueba: Exportar paquete SCORM<sup>5</sup>.

##### Descripción General

El caso de uso se inicia cuando el actor selecciona la opción exportar SCORM, que le permite realizar una acción sobre distintos tópicos (Capítulo, Tema o Subtema). El actor puede incluir, ver, seleccionar de manera aleatoria, además de seleccionar y desmarcar todos los recursos, (imágenes, videos, audios, simulaciones, animaciones y contenidos), que desea exportar. La acción ver se activa cuando se accede

<sup>4</sup>Disponible en: [http://10.128.50.59/svn/repo/Documentacion/IV - Expediente de Proyecto/1\\_ingenieria/1.3\\_implementacion\\_y\\_prueba/Diseno\\_de\\_Caso\\_de\\_Pruebas/Subsistemas/RESOURCES/SCORM](http://10.128.50.59/svn/repo/Documentacion/IV - Expediente de Proyecto/1_ingenieria/1.3_implementacion_y_prueba/Diseno_de_Caso_de_Pruebas/Subsistemas/RESOURCES/SCORM)

<sup>5</sup> Casos de Prueba, Ver Anexo V





alguna de las categorías pertenecientes a alguno de los recursos existentes. Cuenta con un visor para cada tipo de recurso, se activa mediante la selección de un recurso en particular. Si el actor elige la opción de seleccionar todos, el sistema marcará por defecto todos los recursos pertenecientes a esa categoría, la respuesta del sistema en el caso de desmarcar todos, se desmarcarán los recursos previamente marcados. Una vez el actor realiza la selección de los recursos de una categoría, el sistema los adiciona a la lista de recursos seleccionados. Cuenta además con una acción que le permite crear y generar el paquete SCORM.

### Condiciones de Ejecución:

Debe haberse generado el escritorio de trabajo del usuario autenticado.

Debe de seleccionarse el o los recursos a exportar.

ID del escenario	Escenario	Título	Respuesta del sistema	Flujo central
EC1	EC 1.1 Selecciona la opción de Exportar Scorm para un Capítulo, Tema o Subtema.		El sistema realiza una búsqueda de los recursos pertenecientes a estos tópicos. Por lo que se activa el <u>CP Buscar los recursos del tópico seleccionado.</u>	Gestión de Macroíndice de Materia.  Clic derecho en capítulo tema o subtema.  Exportar SCORM.
			Se levanta una ventana modal donde se cargan con todos los recursos asociados a estos tópicos, además de una breve descripción de cómo se debe de llevar a cabo el proceso seleccionado.	
			Brinda la posibilidad de: <ul style="list-style-type: none"> <li>- Titular el paquete que se desea exportar.</li> <li>- Seleccionar los recursos de manera aleatoria.</li> <li>- Seleccionar todos los recursos mediante el botón Select All.</li> <li>- Desmarcar todos los recursos</li> </ul>	



			<p>mediante el botón Deslect All.</p> <ul style="list-style-type: none"> <li>- Mostrar los recursos por cada categoría existente.</li> <li>- Ver listado de los recursos seleccionados. Se activa el CP Listar recursos seleccionados.</li> </ul> <p>Y permite:</p> <ul style="list-style-type: none"> <li>- Generar un paquete SCORM.</li> <li>- Descargar el paquete SCORM generado.</li> <li>- Cerrar las operaciones en cualquier momento.</li> </ul>	
EC2	EC 2.1 Selecciona la pestaña Summary e introduce el título del paquete.	V		<p>Gestión de Macroíndice de Materia.</p> <p>Clic derecho en capítulo tema o subtema.</p> <p>Exportar SCORM.</p>
EC3	EC 3.1 Selecciona una de las pestañas perteneciente a los recursos existentes.		<p>Brinda la posibilidad de realizar las acciones:</p> <ul style="list-style-type: none"> <li>- Seleccionar todos los recursos mediante el botón Select All.</li> <li>- Desmarcar todos los recursos mediante el botón Deslect All.</li> <li>- Seleccionar aleatoriamente los recursos.</li> <li>- Pre-visualizar el o los recursos.</li> </ul>	<p>Gestión de Macroíndice de Materia.</p> <p>Clic derecho en capítulo tema o subtema.</p> <p>Exportar SCORM.</p>
EC4	EC 4.1 Selecciona la opción Select All.		<p>El sistema brinda la posibilidad de seleccionar de forma automática todos los recursos pertenecientes a la categoría seleccionada.</p>	<p>Gestión de Macroíndice de Materia.</p> <p>Clic derecho en capítulo tema o subtema.</p> <p>Exportar SCORM.</p>
EC5	EC 5.1 Selecciona la opción Deslect All.		<p>El sistema brinda la posibilidad de desmarcar todos los recursos de la categoría escogida.</p>	<p>Gestión de Macroíndice de Materia.</p> <p>Clic derecho en capítulo tema o subtema.</p>



				Exportar SCORM.
EC6	EC6.1 Opta por la pre-visualización de un recurso dando click uno de ellos.		El sistema muestra en un visor de recurso, el que ha sido seleccionado, esto es válido para todos los recursos.	Gestión de Macroíndice de Materia. Clic derecho en capítulo tema o subtema. Exportar SCORM.
EC7	EC 7.1 Selecciona la opción de Export Package SCORM.		Brinda la posibilidad de: <ul style="list-style-type: none"> <li>- Incluyen los recursos seleccionados al paquete.</li> <li>- Crear el paquete SCORM, en el servidor (el paquete posee una extensión .ZIP). Este contiene carpetas que contienen los elementos seleccionados, además de una carpeta llamada Metadata que contiene los metadatos asociados a cada recurso. Se crea también un XML llamado imsmanifest.xml el cual contiene la estructura definida para la creación de un paquete SCORM y otros ficheros que este paquete trae por defecto.</li> </ul> Se crea un vínculo de descarga llamado <u><a href="#">Click here to download.</a></u>	Gestión de Macroíndice de Materia. Clic derecho en capítulo tema o subtema. Exportar SCORM.
EC8	EC 8.1 Selecciona la opción <u><a href="#">Click here to download.</a></u>		Se levanta el cuadro de diálogo del navegador que le permite al usuario guardar el paquete creado en la dirección deseada.	Gestión de Macroíndice de Materia. Clic derecho en capítulo tema o subtema. Exportar SCORM.
EC9	EC 9.1 Selecciona la opción Cerrar.		Brinda la posibilidad de cerrar la ventana modal. El caso de uso termina	Gestión de Macroíndice de Materia. Clic derecho en capítulo tema o subtema. Exportar SCORM.



EC10	EC 10.1 Opta por marcar el checkbox de los recursos.		Separa los elementos seleccionados de los que no lo están. <i>Se activa el <u>CP Listar recursos seleccionados</u>.</i>	Gestión de Macroíndice de Materia. Clic derecho en capítulo tema o subtema. Exportar SCORM.
EC11	EC11.1 No selecciona el o los recursos a exportar.		El sistema muestra un mensaje de información.	Gestión de Macroíndice de Materia. Clic derecho en capítulo tema o subtema. Exportar SCORM.
EC12	EC12.1 Selecciona la flecha que se encuentra en una de las categorías.		El sistema oculta o muestra los elementos existentes.	Gestión de Macroíndice de Materia. Clic derecho en capítulo tema o subtema. Exportar SCORM.

Una vez realizado los Casos de Prueba, se puede pasar a verificar las No Conformidades (en adelante NC), detectadas en la 1ra iteración, ya sea de CU o del Sistema. Para evaluar las NC se llevará a cabo la elaboración de una tabla, que contará con: *el Requisito funcional, No conformidad detectada, Importancia de la NC (clasificadas en 3, 2 o 1), Descripción (clasificada en Alta, Media o Baja) y estado con respecto a la Solución (RA: Resuelta y Aprobada por el revisor, PD: Pendiente por solución del equipo de desarrollo, NP: No Procede, PR: Pendiente por revisión conjunta, PC: Pendiente por la parte del cliente, AV: Aplazada para resolver en próximas versiones, NR: Nuevo Requisito).*

#### NC detectadas en los Casos de Uso

No. NC	Requisitos funcionales	No Conformidad	Importancia de la NC	Descripción	Estado con respecto a la solución
1	CU_Importar paquete	Arreglar el CU Importar paquete SCORM flujo alternativo 1.a que no	2	Media	RA



	SCORM	coincide con el CP en el EC 11. Además especificar que es solo para esos tipos de recursos, aunque se pongan los tipos de recursos debajo, arriba se deja muy abierta la idea de para los tipos de recursos.			
--	-------	--	--	--	--

**NC detectadas en el Sistema**

No. NC	Requisitos funcionales	No Conformidad	Importancia de la NC	Descripción	Estado con respecto a la solución
1	CP_Listar paquetes SCORM.	Se ha encontrado una no conformidad en el escenario: "EC10.1 Selección la opción Delete". El sistema muestra un mensaje de error y no elimina.	3	Alta	RA
2	CP_Listar paquetes SCORM.	Se ha encontrado una no conformidad en el escenario: "EC8.1 Selecciona la opción Apply". La opción Apply situada debajo de la tabla no funciona y no permite	3	Alta	RA



		eliminar el campo seleccionado.			
3	CP_Importar paquete SCORM	El idioma de la aplicación está en español y aún así muestra mensajes en inglés.	1	Baja	RA
4	CP_Importar paquete SCORM	En el escenario que tiene como id EC11 dice que para todos los recursos al dar click sobre él debe abrirse un visor de recursos pero para las imágenes esto no ocurre.	-	-	NP
5	CP_Listar paquetes SCORM.	En la aplicación se ha encontrado una no conformidad, en la dirección Recursos Interactivos/Soporte SCORM/Listar Paquetes, al cambiar el idioma el párrafo principal se mantiene en francés.	1	Baja	RA
6	CP_Exportar paquete	El idioma de la aplicación está en inglés y aun así	1	Baja	RA

	SCORM	muestra mensajes en español.			
7	CP_Exportar paquete SCORM	El idioma de la aplicación está en español y muestra mensajes en inglés.	1	Baja	RA
8	CP_Exportar paquete SCORM	En el escenario que tiene como id EC6 dice que todos los recursos deben ser mostrados en un visor de recursos pero para las imágenes esto no pasa.	3	Alta	RA
9	CP_Listar recursos seleccionados	El mensaje tiene un error en la frase "Debe 'de' escribir"... el 'de' creemos que no debe estar. Lo mismo sucede en el "debe 'de' seleccionar"...	1	Baja	RA

#### 4.6.2.- Pruebas de Caja Blanca

Se denomina cajas blancas a un tipo de pruebas de software que se realiza sobre las funciones internas de un módulo. Entre las técnicas usadas se encuentran; la cobertura de caminos (pruebas que hagan que se recorran todos los posibles caminos de ejecución), pruebas sobre las expresiones lógico-aritméticas, pruebas de camino de datos (definición-uso de variables), comprobación de bucles (se verifican los bucles para 0,1 y n iteraciones).

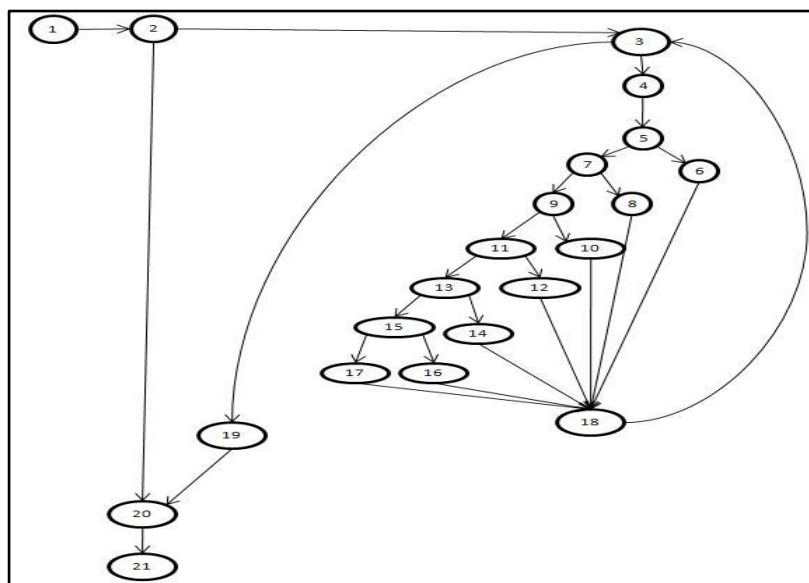
Las pruebas de caja blanca son un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivar casos de prueba que garanticen que:

- ✓ Se ejercitan todos los caminos independientes de cada módulo.
- ✓ Se ejercitan todas las decisiones lógicas.
- ✓ Se ejecutan todos los bucles.
- ✓ Se ejecutan las estructuras de datos internas.

Para la realización de estas pruebas de caja blanca se utilizará la técnica de camino básico o cobertura de caminos. El método del camino básico permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. En este tipo de pruebas se elaboran grafos de flujo, se determina su complejidad ciclomática para obtener el conjunto básico de caminos linealmente independientes y de esta forma los casos de prueba del conjunto básico con el objetivo de ejecutar al menos una vez cada sentencia del programa.

Dada la complejidad y extensión de las pruebas requeridas, en el presente capítulo se hará alusión a la prueba realizada al CU Importar SCORM a la función “*executeReadTotal*”. A continuación se muestra la misma.

#### 4.6.2.1.- Grafo de complejidad.







#### 4.6.2.2.- Complejidad ciclomática.

La complejidad ciclomática (VG) de un grafo de flujo está dada por la cantidad de aristas menos la cantidad de nodos más dos.

$$V(G) = \text{Aristas} - \text{Nodos} + 2$$

$$V(G) = (28 - 21) + 2 = 9$$

Dando como resultado que existen 9 caminos linealmente independientes:

**Camino 1:** 1 – 2 – 20 – 21

**Camino 2:** 1 – 2 – 3 – 19 – 20 – 21

**Camino 3:** 1 – 2 – 3 – 4 – 5 – 6 – 18 – 3 – 19 – 20 – 21

**Camino 4:** 1 – 2 – 3 – 4 – 5 – 7 – 8 – 18 – 3 – 19 – 20 – 21

**Camino 5:** 1 – 2 – 3 – 4 – 5 – 7 – 9 – 10 – 18 – 3 – 19 – 20 – 21

**Camino 6:** 1 – 2 – 3 – 4 – 5 – 7 – 9 – 11 – 12 – 18 – 3 – 19 – 20 – 21

**Camino 7:** 1 – 2 – 3 – 4 – 5 – 7 – 9 – 11 – 13 – 14 – 18 – 3 – 19 – 20 – 21

**Camino 8:** 1 – 2 – 3 – 4 – 5 – 7 – 9 – 11 – 13 – 15 – 16 – 18 – 3 – 19 – 20 – 21

**Camino 9:** 1 – 2 – 3 – 4 – 5 – 7 – 9 – 11 – 13 – 15 – 17 – 18 – 3 – 19 – 20 – 21

#### 4.6.2.3.- Casos de pruebas diseñados para recorrer cada camino.

No. del Camino	Caso de Prueba	Resultado Esperado
Camino 1	Se importa un paquete sin recursos.	Devuelve todos los arreglos vacíos.
Camino 2	Se importa un paquete con un recurso.	Devuelve un solo elemento.
Camino 3	Se importa un paquete con imágenes.	Solamente agrega valores al arreglo de imágenes.
Camino 4	Se importa un paquete con audios.	Solamente agrega valores al arreglo de audios.
Camino 5	Se importa un paquete con videos.	Solamente agrega valores al arreglo de videos.
Camino 6	Se importa un paquete con animaciones.	Solamente agrega valores al arreglo de animaciones.
Camino 7	Se importa un paquete con simulaciones.	Solamente agrega valores al arreglo



		de simulaciones.
Camino 8	Se importa un paquete con contenidos.	Solamente agrega valores al arreglo de contenidos.
Camino 9	Se importa un paquete con recursos desconocidos.	Solamente agrega valores al arreglo de desconocidos.

#### **NC detectadas en las pruebas de Caja Blanca**

Se identificó solamente 1 NC en las pruebas realizadas al fragmento de código analizado, la cual fue debidamente Solucionada y Resuelta.

#### **4.7.- Conclusiones parciales**

En este capítulo se definieron los casos de prueba basados en CN y CB, que sirvieron para guiar al cliente en cuanto a las funcionalidades del sistema para ser aceptado conforme a sus necesidades. Las pruebas efectuadas al producto permitieron corregir errores, que mejoran la calidad de la solución y posibilita la aceptación final del cliente.

La extensión sfScormPlugin, cumple las expectativas de los interesados al aportarles la posibilidad de intercambiar recursos y contenidos entre varios LMS.



## CONCLUSIONES GENERALES

El desarrollo de la investigación aportó conocimientos sobre las posibilidades que brinda el estándar SCORM. El estudio de plataformas que emplean este estándar, demostró lo importante que resultan para la reutilización de contenidos y recursos, además de la interoperabilidad entre varios LMS.

La selección de la Metodología de desarrollo RUP, permitió la guía del proceso y posibilitó la creación de artefactos fundamentales que posibilitaron la construcción de las funcionalidades de la extensión *sfScormPlugin*, con una alta calidad, cumpliendo con las expectativas esperadas por el cliente. Las herramientas empleadas ayudaron a realizar los modelos y diagramas útiles para la implementación de la propuesta de solución, además de los bocetos de interfaz de usuario, considerados como una guía para la implementación del producto.

La integración del producto *sfScormPlugin* a la plataforma ZERA, demostró su validez, así como el cumplimiento de las características requeridas y de los objetivos propuestos, avalada por las pruebas de calidad aplicadas para la detección y corrección de errores.

La documentación generada con este trabajo, posibilitó la implementación de la extensión *sfScormPlugin* y servirá de material de apoyo para otras investigaciones.



## RECOMENDACIONES

Toda vez concluida la investigación, se realizan las propuestas más abajo descritas, con el objetivo de su generalización y empleo. En próximas investigaciones que se realicen sobre el objeto de estudio descrito recomendamos:

- ✓ Separar totalmente la extensión *sfScormPlugin* de la plataforma ZERA, con el objetivo de que pueda ser incorporado en otros LMS.
- ✓ Incorporar la propiedad “*Secuenciación y Navegación*” que posee el estándar.
- ✓ Documentar todo el código para que pueda ser comprendido por futuros desarrolladores.



## BIBLIOGRAFÍA

### REFERENCIADA

1. *Software Educativos*. Ledo, María Vidal, Martínez, Freddy Gómez y Piedra, Alina M. Ruiz. La Habana : ems, 2009, Vol. vol\_24\_1\_10. ems12110.
2. **Universidad Autónoma de Barcelona, Pere Marqués**. El Software Educativo. *Biblioteca virtual de tecnología educativa*. [En línea] 4 de Marzo de 2010. [Citado el: 17 de Noviembre de 2010.] [http://www.lmi.ub.es/te/any96/marques\\_software/#index](http://www.lmi.ub.es/te/any96/marques_software/#index).
3. **Damián Pérez Valdés**. ¿Qué son las bases de datos? *Maestros del Web*. [En línea] 26 de Octubre de 2007. [Citado el: 17 de Noviembre de 2010.] <http://www.maestrosdelweb.com>.
4. **EcuRed**. Base de Datos. *EcuRed*. [En línea] [Citado el: 17 de Noviembre de 2010.] <http://www.ecured.cu>.
5. **Económicas On Line**. *Económicas On Line, Términos de Internet*. [En línea] [Citado el: 17 de Enero de 2011.] <http://www.economicas-online.com/glosarios/terminos.htm>.
6. **WordPress**. *Glorasio, WordPress*. [En línea] [Citado el: 17 de Enero de 2011.] <http://buxara.wordpress.com/glosarios>.
7. **datarec**. *datarec*. [En línea] [Citado el: 17 de Enero de 2011.] <http://www.datarecsa.com/Glosario/glosariop.htm>.
8. **EcuRed**. *EcuRed*. [En línea] [Citado el: 2 de Diciembre de 2010.] [http://www.ecured.cu/Plataformas\\_Educativas.html](http://www.ecured.cu/Plataformas_Educativas.html).
9. **IEEE**. *IEEE*. [En línea] [Citado el: 2 de Diciembre de 2010.] <http://www.ieee.org>.
10. *Connecting learning objects to instructional design theory: A definition, a metaphor and a taxonomy*. . (Ed), In D. Wiley. 2000.
11. *Use and Abuse of Reusable Learning Objects*. **Polsani, Pithamber R**. Learning Technology Center, University of Arizona, USA : Journal of Digital Information, 2003, Vol. Vol 3. No 4.
12. *Keeping the Learning in Learning Objects*. **REHAK., D., MASON, R. In Littlejohn**. Special Issue on Reusing Online Resources 1, s.l. : A.(ed): Journal of Interactive Media in Education, 2003.
13. **Navarro, M<sup>a</sup>. Ángeles Alonso**. *Como crear contenidos educativos accesibles para plataformas de e-learning*. 2009.
14. **EcuRed**. Framework. *EcuRed*. [En línea] [Citado el: 17 de Noviembre de 2010.]



<http://www.ecured.cu/framework>.

15. *Instituto Superior de Formación y Recursos en Red para el Profesorado*. [En línea] [Citado el: 2 de Diciembre de 2010.] <http://ares.cnice.mec.es/informes/16/contenido/3.htm>.
16. *Instituto Superior de Formación y Recursos en Red para el Profesorado*. [En línea] [Citado el: 2 de Diciembre de 2010.] <http://ares.cnice.mec.es/informes/16/contenido/8.htm#up>.
17. Le Cerdecam. [En línea] 15 de 12 de 2010. <http://www.cerdecam.be/>.
18. **Claroline**. *Claroline*. [En línea] [Citado el: 7 de Diciembre de 2010.] <http://www.claroline.net>.
19. **ATutor**. *ATutor Learning Management Tools*. [En línea] [Citado el: 4 de Diciembre de 2010.] <http://atutor.ca/>.
20. **EcuRed**. Moodle. *EcuRed*. [En línea] [Citado el: 25 de Noviembre de 2010.] <http://www.ecured.cu/index.php/Moodle>.
21. **SAdHEA**. Plataforma SAdHEA . [En línea] [Citado el: 7 de Diciembre de 2010.] <http://www.SAhHEA.com> .
22. **CRODA FORTES** Centro de Teconología de Tecnologías para la Formación. [En línea] UCI FORTES. [Citado el: 7 de Enero de 2011.] <http://portal.fortes.prod.uci.cu/projects/haa>.
23. **Fortes**. *Centro de Tecnologías para la Formación (FORTES)*. [En línea] [Citado el: 12 de Enero de 2011.] <http://portal.fortes.prod.uci.cu/projects/hacoa> .
24. **MOODLE, Personalizaciones de**. FORTES Centro de Tecnologías para la Formación. [En línea] [Citado el: 7 de Enero de 2011.] <http://portal.fortes.prod.uci.cu/projects/pms>.
25. **Kruchten, P.** *A Software Development Process for a Team of One*. 2002.
26. **Rational Software Corporation**. *Rational Rose Copyrigh 2003*. [En línea] [Citado el: 26 de Noviembre de 2010.] [http://www.slideshare.net/vivi\\_jocadi/rational-rose](http://www.slideshare.net/vivi_jocadi/rational-rose).
27. **UML tool for software application development**. *Visual Paradigm for UML*. [En línea] 1 de Febrero de 2010. [Citado el: 27 de Noviembre de 2010.] <http://www.visual-paradigm.com/product/vpuml>.
28. **Mockups, Balsamiq**. Balsamiq Mockups. [En línea] [Citado el: 10 de Enero de 2011.] <http://www.balsamiq.com..>
29. **Symfony Corporate**. *Symfony*. [En línea] [Citado el: 12 de Diciembre de 2010.] <http://www.symfony.es>.
30. **ANADIC SINALOA**. ANADIC SINALOA. [En línea] [Citado el: 5 de Diciembre de 2010.] <http://www.anadicsinaloa.com/index.php>.



## CONSULTADA

1. **ADL**. Advanced Distributed Learning, 2010. [Disponible en: <http://www.adlnet.org>]
2. **Claroline**. Plataforma Claroline. [Disponible en: <http://www.claroline.net/>]
3. Claude Ostyn. **In the Eye of the SCORM** An introduction to SCORM 2004 for Content Developers, USA, 2005, 2006.
4. Comité de Estandarización de Tecnologías Educativas del IEEE. Estándar para Metadatos de Objetos Educativos. New York, Institute of Electrical and Electronics Engineers, Inc. 2002
5. **desarrolloweb.com**. Zend Studio, 2003. [Disponible en: <http://www.desarrolloweb.com/> ]
6. Doug Rosenberg and Matt Stephnes. Use Case Driven Object Modeling with UML. Apress. Berkeley, 2007.
7. **Drupal**. Sitio Drupal. [Disponible en: <http://www.drupal.org/>]
8. **EcuRed**. CSS. [Disponible en: <http://www.ecured.cu/> ]
9. El servidor de web Apache: Introducción práctica. [Disponible en <http://acsblog.es/articulos/trunk/LinuxActual/Apache/html/index.html>]
10. **ITE**. Instituto de Tecnologías Educativas (ITE). [Disponible en: <http://www.ite.educacion.es>]
11. Kiskeya. CSS [Disponible en: <http://es.kioskea.net/contents/css/cssintro.php3>]
12. **MOODLE**. Plataforma MOODLE. [Disponible en: <http://www.moodle.com/>]
13. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP), 2010. [Disponible en: <http://www.willydev.net/> ]
14. PostgreSQL. PostgreSQL 7.1 Tutorial. [Disponible en: <http://www.postgresql.org/>]
15. PostgreSQL 7.3.2 Tutorial. The PostgreSQL Global Development Group, 1996-2002. [Disponible en: <http://www.postgresql.org/>]
16. **Symfony**. Plataforma Symfony. [Disponible en: <http://www.symfony.es/>]
17. Scribd. Manual Imprescindible de PHP5, 2004. [Disponible en: <http://www.scribd.com/doc/6779555/Anaya-PHP-5>]
18. WordPress. Desarrollo de Software – Metodología Tradicional o Ágil, 2010. [Disponible en: <http://pmqlinkedin.wordpress.com/>]
19. w3schools. CSS Tutorial [Disponible en: <http://www.w3schools.com/css/default.asp>]
20. XML. Extensible Markup Language, 2011 [Disponible en <http://www.w3c.org/XML/>]