



**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**FACULTAD 4**

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

***Bloque para la exportación de contenidos  
desde la plataforma de teleformación  
Moodle 2.1.x***

AUTORES:

**Linet Katuska Remón Salcedo**

**Boris Luis Correa Frías**

TUTORES:

**Ing. Yolanda Sardiñas Suárez**

**Ing. Jesús Hidalgo Guillén**

COTUTOR:

**Ing. Ana Delia González Ricardo**

La Habana, Cuba.

Junio, 2012

*"Dime y lo olvido, enséñame y lo recuerdo,  
involúcrame y lo aprendo."*

*Benjamin Franklin*



# Declaración de autoría

---

Declaramos que somos los únicos autores de este trabajo y autorizamos al Centro de Tecnologías para la Formación (FORTES) de la Universidad de las Ciencias Informáticas (UCI) a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los \_\_\_ días del mes de \_\_\_ de 2012.

Autores

\_\_\_\_\_  
Linet Katuska Remón Salcedo

\_\_\_\_\_  
Boris Luis Correa Frías

Tutores

\_\_\_\_\_  
Ing. Jesús Hidalgo Guillén

\_\_\_\_\_  
Ing. Yolanda Sardiñas Suárez

Cotutor

\_\_\_\_\_  
Ing. Ana Delia González Ricardo

Ing. Jesús Hidalgo Guillén

Correo electrónico: [jhidalgo@uci.cu](mailto:jhidalgo@uci.cu)

Ingeniero en Ciencias Informáticas, UCI, 2008. Profesor. Trabaja en el “Departamento de Producción de Herramientas Educativas” del centro FORTES de la facultad 4. Tiene 5 años de experiencia en el tema y 3 años de graduado. Ha participado en varios eventos nacionales e internacionales con trabajos relacionados con el tema de la teleformación.

Ing. Yolanda Sardiñas Suárez

Correo electrónico: [yssuarez@uci.cu](mailto:yssuarez@uci.cu)

Ingeniera en Ciencias Informáticas, UCI, 2008. Profesor. Trabaja en el “Departamento de Producción de Herramientas Educativas” del centro FORTES de la facultad 4. Tiene 5 años de experiencia en el tema y 3 años de graduada. Ha participado en varios eventos nacionales con trabajos relacionados con el tema de la teleformación.

Ing. Ana Delia González Ricardo

Correo electrónico: [adgonzalez@uci.cu](mailto:adgonzalez@uci.cu)

Ingeniera en Ciencias Informáticas, UCI, 2010. Trabaja en el Departamento de Producción de Herramientas Educativas. Se encuentra en su segundo año de adiestramiento. Un año de experiencia en la tutoría de tesis y de trabajo en la producción. Ha participado en eventos, nacionales e internaciones, relacionados con la temática: Soluciones para las herramientas educativas. Cuenta con publicaciones relacionadas con la Usabilidad y Accesibilidad en la plataforma de teleformación Moodle y Tecnologías para la Formación.

*A los tutores por su ayuda y al tribunal por su comprensión.*

***Boris Luis Correa Frias:***

*A mis padres Boris y Yolanda, y mi hermana Yola, por el apoyo y dedicación que me siempre me han brindado, por haber sido siempre ejemplo de entrega y sacrificio a seguir.*

*A mi compañera de tesis Linet, por su entrega, creatividad, paciencia.*

***Linet Katiuska Remón Salcedo:***

*A mis padres Emelina y Reunel, todo lo que soy es gracias a ustedes y para ustedes es todo lo que hago, a mis hermanas por ser mi fuente de inspiración y a mis amigos y compañeros por el apoyo.*

*A Boris, por ser un excelente compañero de tesis, por su paciencia, constancia y entrega.*

*A Yunion Orosa por su ayuda incondicional.*

*A la Revolución, gracias por la confianza y por esta oportunidad.*

*Boris Luis Correa Frias:*

*A mis padres y a mi hermana.*

*Linet Katiuska Remón Salcedo:*

*A mis padres y hermanas por su apoyo y confianza.*

---

## Resumen

El e-Learning ha revolucionado la enseñanza apoyándose en las Tecnologías de la Información y la Comunicación (TIC) para su extensión y desarrollo. Uno de los Sistemas Gestores del Aprendizaje (LMS) más potentes en este campo es la plataforma de teleformación Moodle, con la cual la Universidad de las Ciencias Informáticas (UCI) creó su Entorno Virtual de Aprendizaje (EVA). Para esta personalización se desarrolló el bloque C2SCORM que permite la exportación de cursos y/o parte de estos bajo el estándar SCORM 1.2. Posteriormente se le incorporó al que el estándar SCORM 2004 para Moodle 1.9.4. Las versiones 2.x de Moodle transformaron radicalmente su estructura, afectando a la mayoría de sus componentes lo que imposibilita la utilización del bloque C2SCORM 2.0 en versiones a partir de la 2.x. El objetivo del presente trabajo fue desarrollar una extensión para la plataforma Moodle 2.1.x que permitiera la exportación de los contenidos de sus cursos facilitando así la reutilización de estos en otros entornos educativos. Se emplearon los métodos de investigación científica histórico-lógico, analítico-sintético y modelación. Como resultado de la investigación se obtuvo C2SCORM 3.0, un bloque para la exportación de contenidos de los cursos de la plataforma Moodle 2.1.x en forma de paquete SCORM 2004, que facilitó su reutilización en otros entornos compatibles con el estándar.

**Palabras clave:** e-Learning, exportación, SCORM, Moodle, reutilización.

---

# Índice

<i>Introducción</i> .....	1
<i>Capítulo I: Fundamentación Teórica</i> .....	6
Introducción.....	6
1.1 e-Learning .....	6
1.2 Sistemas Gestores del Aprendizaje (LMS) y Sistemas Gestores de Contenidos del Aprendizaje (LCMS) .....	7
1.3 Plataforma de teleformación Moodle .....	9
1.4 Análisis del bloque C2SCORM 2.0 para la plataforma Moodle 1.9.4 .....	9
1.5 Análisis de los cambios significativos de la plataforma de teleformación Moodle 2.1.x respecto a su versión 1.9.4.....	11
1.6 Estándares de interoperabilidad .....	15
1.6.1 <i>Estándar para la descripción de contenidos</i> .....	16
1.6.2 <i>Estándar para el empaquetamiento de contenidos</i> . .....	17
1.7 Tecnologías a utilizar en el desarrollo del sistema .....	20
1.7.1 <i>Tecnologías del lado del servidor</i> .....	20
1.7.2 <i>Tecnologías del lado del cliente</i> .....	22
1.8 Herramientas a utilizar en el desarrollo del sistema .....	23
1.9 Metodología de desarrollo de software .....	24
1.10 Conclusiones parciales.....	27
<i>Capítulo II: Análisis y Diseño del sistema</i> .....	28
Introducción.....	28
2.1 Propuesta de solución .....	28
2.2 Modelo de Dominio.....	29



2.3 Especificación de Requisitos .....	30
2.3.1 Requisitos Funcionales .....	31
2.3.2 Requisitos No Funcionales .....	33
2.4 Modelo de Casos de Uso del Sistema .....	35
2.4.1 Descripción de Casos de Uso del Sistema.....	36
2.5 Modelo de Análisis .....	42
2.5.1 Diagramas de interacción .....	44
2.6 Modelo de Diseño .....	45
2.7 Estructura de la base de datos .....	48
2.8 Conclusiones parciales.....	50
Capítulo III: Implementación y Pruebas.....	51
Introducción.....	51
3.1 Modelo de Implementación.....	51
3.2 Diagrama de Componentes.....	52
3.3 Pruebas.....	57
3.4 Conclusiones parciales.....	60
Conclusiones .....	61
Recomendaciones .....	62
Referencias bibliográficas.....	63

---

## Introducción

El vertiginoso desarrollo de las Tecnologías de la Información y la Comunicación ha provocado el nacimiento de una cultura informática que se integra a cada una de las esferas de la vida social. Han sido incorporadas especialmente a las actividades laborales y, por sus potencialidades, al mundo de la educación. Sus efectos se evidencian con el surgimiento de nuevas formas de llevar a cabo los procesos de enseñanza-aprendizaje.

Uno de estos novedosos métodos es el e-Learning, definido por el Grupo de Trabajo de "e-Learning" 05 de la Red TTNNet España como: *“conjunto de tecnologías, aplicaciones y servicios orientados a facilitar la enseñanza y el aprendizaje a través de Internet / Intranet, que facilitan el acceso a la información y la comunicación con otros participantes.”* (1)

Para apoyar esta modalidad educativa surgen herramientas que evolucionan hasta los Sistemas Gestores del Aprendizaje (LMS por sus siglas en inglés) que permiten crear y administrar contenidos, así como controlar los procesos de enseñanza-aprendizaje. Los LMS pueden enfocarse en unidades atómicas como los objetos de aprendizaje (OA).

Un objeto de aprendizaje se define como cualquier recurso con una intención formativa, compuesto de uno o varios elementos digitales, descrito con metadatos, que pueda ser reutilizado dentro de un entorno e-Learning. (2)

Este término procede del paradigma de la Programación Orientada a Objetos (POO) y supone la creación de componentes (llamados objetos), con el fin de que puedan ser reutilizados en varios contextos. Esta filosofía les brinda a los OA la capacidad de ser autosuficientes, independientes y reutilizables, siendo esta última característica su principal objetivo para poder ser empleados en diferentes escenarios y facilitar su clasificación.

Entre los LMS más reconocidos se encuentra la plataforma de teleformación Moodle, aplicación para crear sitios web dinámicos de forma gratuita para el aprendizaje en línea. Sus características le permiten escalar a grandes despliegues con elevadas

cantidades de estudiantes y la creación de comunidades de aprendizaje entre los participantes.

La Universidad de las Ciencias Informáticas (UCI) a partir del año 2005 comenzó a utilizar la plataforma Moodle como parte de su programa de teleformación, que buscaba un aumento en la calidad de los procesos educativos.

Inicialmente la plataforma carecía de la capacidad de reutilizar sus contenidos, al no permitir la exportación de estos en forma de paquete SCORM para su uso en otros entornos, por lo que era necesario reelaborar la información de los cursos montados en la plataforma en forma de sitios web personalizados.

Para solventar esta deficiencia se crea en el año 2008 el bloque C2SCORM para la plataforma Moodle 1.8.6 (del inglés Course To SCORM) que permite la exportación de contenidos de los cursos de la plataforma utilizando el estándar SCORM 1.2. Los cursos exportados bajo este estándar permiten pequeñas variaciones con respecto a la secuenciación pero no dejan de tener una estructura demasiado lineal, poco propensa a modificaciones y sin dinamismo.

En el año 2011, el equipo de trabajo del proyecto Personalizaciones de Moodle del Centro de Tecnologías para la Formación (FORTES) le incorpora el estándar SCORM 2004 al bloque C2SCORM 2.0 para la plataforma Moodle 1.9.4. Este flexibiliza la secuenciación y aumenta el dinamismo de los contenidos.

Actualmente el Entorno Virtual de Aprendizaje (EVA) de la UCI está desplegado sobre Moodle 1.9.12 +, sin embargo, para mantener la sincronía con la evolución de la plataforma debe migrar dentro de un corto período de tiempo a una versión superior 2.x. A diferencia de las versiones anteriores, estas han sido reimplementadas totalmente por la comunidad de desarrolladores de Moodle proponiendo una arquitectura totalmente mejorada, con una mayor dinámica, acompañada de cambios que dotan a la plataforma de mayor flexibilidad, robustez, facilidad de uso, así como modificaciones en la forma de navegar y el diseño de las acciones formativas.

Cada uno de los recursos del módulo de actividades recursos ha sido reimplementado como un módulo independiente, las actividades wiki y taller fueron reescritas completamente, nuevos tipos de preguntas fueron adicionadas a los cuestionarios, así como cambios significativos en la sección de la base de datos referente a archivos XML y sintaxis de diferentes funciones. Al módulo SCORM se le realizaron mejoras en

la interfaz del reproductor de lecciones en aspectos como la navegación, desempeño y manejo del tamaño del estado.

El bloque C2SCORM 2.0 permite la exportación de contenidos en forma de paquete SCORM para la versión 1.9.4 de Moodle, sin embargo los cambios en la arquitectura de esta plataforma imposibilitan su reutilización en versiones a partir de la 2.x. Estas versiones, que comienzan a ser estables a partir de la 2.1.x, no cuentan con características que les permitan exportar contenidos en formato SCORM con el fin de reutilizarlos en otros entornos.

Por lo anteriormente expuesto se plantea como **problema de investigación**: ¿Cómo facilitar la reutilización de contenidos de los cursos de la plataforma de teleformación Moodle para la versión 2.1.x?

La presente investigación tiene como **objeto de estudio** la reutilización de contenidos educativos. Para dar solución al problema se define como **objetivo general** desarrollar una extensión para la plataforma Moodle 2.1.x que permita la exportación de contenidos de sus cursos facilitando así la reutilización de estos en otros entornos educativos. El **campo de acción** está enfocado a la reutilización de contenidos educativos de la plataforma Moodle 2.1.x.

Se plantea como **idea a defender** que a partir del desarrollo de una extensión para la plataforma Moodle 2.1.x que permita exportar los contenidos de sus cursos se facilitará la reutilización de estos en otros entornos educativos.

El objetivo general se desglosa en los siguientes **objetivos específicos**:

- o Analizar los elementos teóricos relacionados con las versiones del bloque C2SCORM y de la plataforma de teleformación Moodle.
- o Analizar y diseñar un bloque para la plataforma de teleformación Moodle 2.1.x que facilite la reutilización de los contenidos de los cursos.
- o Desarrollar y evaluar el bloque para la plataforma de teleformación Moodle 2.1.x que facilite la reutilización de los contenidos de los cursos.

Para dar cumplimiento a los objetivos específicos se plantean las siguientes **tareas**:

- o Análisis de los conceptos y tecnologías más utilizadas en el campo de la reutilización de contenidos para el e-Learning.

- o Análisis comparativo entre las arquitecturas de las versiones 1.9.4 y 2.1.x de la plataforma de teleformación Moodle.
- o Investigación del funcionamiento del bloque C2SCORM 2.0.
- o Identificación de las tecnologías y herramientas a utilizar en el desarrollo de un bloque para la plataforma Moodle 2.1.x que facilite la reutilización de los contenidos de los cursos.
- o Análisis y diseño de un bloque para la reutilización de contenidos de los cursos de la plataforma Moodle 2.1.x.
- o Desarrollo y validación de un bloque para la plataforma Moodle 2.1.x que facilite la reutilización de contenidos de los cursos.

Para realizar estas tareas se utilizarán como **métodos científicos de investigación** los histórico-lógicos, para analizar la estructura y funcionamiento de las versiones anteriores del bloque C2SCORM y la evolución de la plataforma Moodle 1.9.4 a las versiones 2.1.x. Se utilizará el método de modelación para reflejar la estructura, relaciones internas y características de la solución a través de diagramas, mediante la utilización del lenguaje de modelado UML. Los métodos analítico-sintéticos permitirán el procesamiento y análisis de la documentación referente a reutilización de contenidos educativos, extrayendo los elementos más importantes y necesarios para dar solución al problema existente.

El trabajo contará con una introducción, tres capítulos, conclusiones, recomendaciones, referencias bibliográficas, bibliografía y anexos.

En el **Capítulo I: Fundamentación Teórica** se abordan de forma general aspectos teóricos relacionados con el e-Learning y los diferentes estándares utilizados en la descripción y reutilización de contenidos. Se analiza de forma comparativa la arquitectura de la plataforma Moodle en sus versiones 1.9.4 y 2.1.x, así como el funcionamiento del bloque C2SCORM 2.0. Se especifican las tecnologías y herramientas a utilizar en el análisis, diseño e implementación de un bloque que permita la reutilización de contenidos de los cursos de la plataforma de teleformación Moodle 2.1.x, y la metodología a seguir en el proceso de desarrollo de software.

El **Capítulo II: Análisis y diseño del bloque** contiene el análisis y diseño de la solución propuesta. Se identifican los requerimientos funcionales y no funcionales, los casos de uso del sistema con sus respectivas descripciones textuales y su relación

con los actores del sistema. Se representan los diagramas de clases del análisis y el diseño, así como la estructura de la base de datos.

En el **Capítulo III: Implementación y Pruebas** se implementa el sistema para la plataforma Moodle 2.1.x y se describe este proceso desde el enfoque de la programación a través de los Diagramas de Componentes y de despliegue. Se realiza la validación de la solución propuesta.

---

# Capítulo I: Fundamentación Teórica

## ***Introducción***

La utilización de las TIC en cada una de las actividades humanas ha provocado que la información digital se convierta en uno de los activos más preciados. Los procesos educativos han sido beneficiados con estos avances obteniendo mecanismos más flexibles y novedosos. En este capítulo se abordan aspectos relacionados con el aprendizaje electrónico. Se describen herramientas, tecnologías y estándares relacionados con esta modalidad educativa y que serán empleados en la implementación de la aplicación, así como la metodología seleccionada para guiar su proceso de desarrollo. Se brinda especial atención, por ser la base de la solución, a la plataforma de teleformación Moodle, sus módulos, bloques y otras extensiones, así como a investigaciones precedentes que solucionaron problemas similares.

## ***1.1 e-Learning***

La integración de las TIC a los procesos educativos como parte de esta revolución tecnológica provocó el surgimiento del aprendizaje electrónico o e-Learning. Este nuevo método presenta numerosas ventajas:

- o Se facilita la retroalimentación de la formación, por lo que el profesor conoce la efectividad de los métodos y el grado de cumplimiento de los objetivos trazados.
- o Las herramientas de socialización de la información permiten que el conocimiento esté a disposición de todos los participantes, a diferencia de la enseñanza tradicional donde este estaba regulado por el profesor que ahora se convierte en un facilitador del aprendizaje.
- o Puede estar centrado en el aprendizaje del estudiante por lo que este último recibe una atención más personalizada.
- o Promueve la autonomía de los participantes y es menos intimidatorio que el método tradicional. (3)
- o Se facilita el acceso a información actual y el contacto con expertos.

- o Facilita la creación, actualización y distribución de contenidos, así como la adaptación al ritmo de aprendizaje del estudiante y la disponibilidad de las herramientas de aprendizaje independientemente de límites horarios o geográficos, por lo que se reducen los gastos en viajes. (4)
- o Sencillo seguimiento de actividades, pues se pueden implementar sistemas de alerta que avisen al profesor y a los estudiantes las fechas de entrega de trabajos, reuniones virtuales o ejercicios remitidos al docente. También se puede utilizar aplicaciones que permiten medir, entre otras cosas: tiempo en que el estudiante está conectado en cada módulo, resultados de los ejercicios, número de intentos del estudiante hasta realizar correctamente un ejercicio y ratios<sup>1</sup> como calificación obtenida / tiempo de lectura del módulo, entre otros. (3)
- o Cuenta con utilidades para la presentación de contenidos: textos, animaciones, gráficos, videos; y herramientas de comunicación entre estudiantes y tutores que facilitan el trabajo colaborativo: foros, chats, correo electrónico, blogs, wikis, entre otras; contribuyendo a una mayor posibilidad de elección. (5)

La demanda y rápida difusión del aprendizaje electrónico, potenció la creación de herramientas electrónicas que brindaran soporte al nuevo método. Entre estas aplicaciones se encuentran los Sistemas Gestores del Aprendizaje (LMS) y los Sistemas Gestores de Contenidos del Aprendizaje (LCMS).

## **1.2 Sistemas Gestores del Aprendizaje (LMS) y Sistemas Gestores de Contenidos del Aprendizaje (LCMS)**

La amplia aceptación del e-Learning condujo al desarrollo de herramientas especializadas que respondieran a las demandas académicas y tecnológicas, y perfeccionaran la administración de los contenidos educativos, así como el control de las actividades formativas que realizan los estudiantes.

Un LMS es un software, residente en un servidor, en el que se desarrollan acciones formativas. Entre las funciones de estos sistemas se encuentran la gestión de usuarios, recursos, materiales y actividades de formación, la administración del acceso, control y seguimiento del proceso de enseñanza-aprendizaje y el almacenamiento de datos de los usuarios. (6)

---

<sup>1</sup> Ratio: vocablo latino que se utiliza como sinónimo de razón, en el sentido del cociente de los números o de cantidades comparables. (21)



# Capítulo I: Fundamentación Teórica

---

Contienen además, herramientas de comunicación, servicios y áreas para los diferentes elementos de cada curso o materia, gestión académica y administrativa para el control de acciones de los usuarios, necesario para el análisis posterior de los resultados, y un sistema que permite gestionar evaluaciones cuantitativas y cualitativas. (6)

Aunque un LMS posee grandes ventajas carece de facilidades para la creación de contenidos educativos, por lo que se hace necesaria la integración con un Sistema de Gestión de Contenidos (CMS, del inglés Content Management System).

En consonancia, un LCMS (Learning Content Management System) engloba todas estas funcionalidades, facilitando la creación y gestión de los contenidos educativos a diferentes niveles.

Según algunos autores “*combina las capacidades de gestión de cursos de un LMS con las capacidades de almacenamiento y creación de contenidos de un CMS*” (7). Sin embargo, otros amplían su significado.

Los LCMS son plataformas multi-usuario donde se puede crear, reusar, almacenar, describir, administrar y distribuir contenido de aprendizaje electrónico desde un repositorio central, con el fin de ser utilizado posteriormente en cursos en línea. De acuerdo con esta definición, puede decirse que la diferencia entre un LMS y un LCMS es que el primero se encuentra enfocado a la publicación y monitoreo de contenido, mientras que el segundo a su proceso de elaboración. (8)

Esta herramienta especializada en contenidos de e-Learning permite reestructurar la información y los objetivos de los contenidos de manera dinámica para diseñar contenidos interactivos, así como crear y modificar los OA que atiendan a necesidades y estilos de aprendizaje específicos, sin conocimientos de programación o diseño web.

Estos contenidos, una vez dentro del sistema, pueden ser combinados, asignados a distintos cursos, descargados. Permiten la búsqueda y exploración de contenido de forma sencilla. Generalmente un LCMS promueve la portabilidad y la interoperabilidad de los recursos de aprendizaje almacenados a través de objetos de aprendizaje y estándares de intercambio. (9)

Los LCMS y LMS han impulsado la aparición de herramientas robustas que se caracterizan por contener gran cantidad de aplicaciones que facilitan el e-Learning.

Ejemplo de estas herramientas es la plataforma de teleformación Moodle sobre la cual se sustenta la presente investigación.

## **1.3 Plataforma de teleformación Moodle**

El Entorno de Aprendizaje Dinámico Orientado a Objetos y Modular (Moodle, del inglés Modular Object-Oriented Dynamic Learning Environment), diseñado por Martin Dougiamas en 2002, es un paquete de software para la creación y administración de cursos y sitios web en Internet. Se distribuye gratuitamente como software libre bajo la licencia pública GNU. Puede funcionar en cualquier ordenador con PHP instalado y soporta varios tipos de bases de datos, esencialmente MySQL y PostgreSQL.

Moodle cuenta con herramientas informáticas que facilitan la creación de actividades como: tareas, cuestionarios, encuestas, libros, glosarios, wikis entre otras; que enriquecen los procesos de enseñanza-aprendizaje. Para extender sus prestaciones, la plataforma permite la instalación de módulos, bloques y otras extensiones desarrolladas por la comunidad, así como importar y emplear objetos de aprendizaje provenientes de diversas fuentes que sigan los estándares SCORM e IMS Content Packaging. (10)

El aspecto del entorno puede personalizarse mediante los bloques de Moodle en función de las necesidades de los estudiantes y profesores. Los bloques tienen un carácter específico y permiten agregar funciones extra para fortalecer las capacidades del sistema. Se sitúan a los lados del sitio web y pueden ser configurados: ocultados, suprimidos o desplazados lateral o verticalmente, siempre que se tenga permisos de edición. Sus diversas funcionalidades pueden estar ligadas a otros bloques o trabajar de forma independiente.

## **1.4 Análisis del bloque C2SCORM 2.0 para la plataforma Moodle 1.9.4**

Con el objetivo de ser utilizados en otras plataformas compatibles, los cursos, son exportados en paquetes de contenido siguiendo las especificaciones del estándar SCORM 2004. Para lograrlo, se le incorpora a la plataforma Moodle el bloque C2SCORM 2.0.

C2SCORM 2.0 permite seleccionar el curso donde se encuentra el contenido a exportar, así como seleccionar las actividades: cuestionario, wiki, glosario, foro,

# Capítulo I: Fundamentación Teórica

---

lección, tarea; y recursos a empaquetar: páginas de texto, web, directorios, archivos. Permite editar los metadatos de los contenidos y propone una vista previa de las actividades y recursos a exportar. Facilita la incorporación de reglas de secuencia y navegación de los contenidos.

Como anteriormente se enuncia, para su implementación el bloque se guía por las especificaciones de secuencia que propone SCORM 2004. Otro de los estándares que emplea, y de los cuales más adelante se enuncian las principales características, es el LOM que agrupa los metadatos en nueve categorías donde se describen las principales características del objeto de aprendizaje. En el bloque solamente se emplean las siguientes (6):

- o General: información que describe el objeto de aprendizaje como un todo.
- o Life Cycle: caracteriza la historia y el estado presente del objeto de aprendizaje y de aquellos que han afectado a este objeto durante su evolución.
- o Meta-Metadata: posee la información sobre los mismos metadatos, no sobre el objeto de aprendizaje que se está describiendo.
- o Technical: constituye los requisitos y características técnicas del objeto de aprendizaje.
- o Educational: muestra las condiciones del uso educativo del recurso.
- o Rights: describe las condiciones de uso para la explotación del recurso.

A diferencia del estilo de la plataforma Moodle 2.1, la programación de C2SCORM 2.0 es estructurada en su mayoría. Razón por la cual las funciones antes existentes se encuentran despreciadas a partir de la versión 2.0.

Este bloque solo exporta cursos en formato semanal y por tema. No incluye las actividades chat, consulta, encuesta, base de datos, el recurso Paquete de contenido IMS y la variante de enlace web del recurso enlaces. Así como los tipos de pregunta del cuestionario: de descripción y de emparejamiento aleatorio de respuestas cortas. El código JavaScript no se encuentra en archivos independientes como exige Moodle 2.1, lo que compromete la accesibilidad del sitio.

C2SCORM 2.0 no incluye, dentro del modelo de definición de secuencia, Controles de Restricción de Opciones que se divide en restricción de opciones y prevención de

activación. Esta especificación evita problemas que se puedan generar con algunas estrategias de secuenciación.

## **1.5 Análisis de los cambios significativos de la plataforma de teleformación Moodle 2.1.x respecto a su versión 1.9.4**

La inclusión de las actuales tendencias del uso de la Web (AJAX, XHTML, CSS, capa de persistencia de datos, servicios web, entre otros) ha provocado cambios significativos en las más recientes versiones de Moodle.

Entre las características del entorno técnico sobre el que se asienta Moodle 1.9.4 se encuentra como requisito una versión de PHP 4.3.0 o superior para funcionar, sin llegar a 5.3 por diferencias potenciales en la sintaxis de los constructores, a diferencia de Moodle en sus versiones 2.1.x, que exigen una versión 5.3.2 o superior del lenguaje.

Para Moodle 1.9.4 los ficheros subidos por los usuarios al sistema se almacenan en la localización *moodledata*, en esta carpeta también se almacenan los datos de sesión de los usuarios que se han autenticado en el sistema. Moodle estructura los datos de esta carpeta por usuarios o por cursos. Cada curso tiene una carpeta, nombrada con un número entero que representa el identificador único asignado a dicho curso en la base de datos. La plataforma almacena datos de sus módulos en la subcarpeta *moddata* y cuando se trata de ficheros lo hace dentro de esta carpeta en un directorio con el mismo nombre que el módulo.

Por su parte Moodle en sus versiones 2.1.x guarda los archivos dentro de la plataforma como parte del contenido. Utiliza un flexible y consistente acercamiento para todo el manejo de archivos a través de la plataforma al brindar a los módulos control sobre el acceso de los usuarios a los archivos utilizando las habilidades y reglas locales, y al permitir el almacenamiento y uso de ficheros en repositorios externos como Alfresco, Box.net y Google Docs. (11)

Facilita determinar qué parte del LMS utiliza determinados archivos. Graba el origen de cada uno de los archivos, evita el almacenamiento redundante permitiendo que un archivo sea utilizado varias veces. Incorpora un soporte completo para los nombres de archivos *unicode*<sup>2</sup> y habilidades del sistema de archivo subyacente.

---

<sup>2</sup> Unicode: estándar internacional de codificación, concebido para evitar la confusión derivada de la coexistencia de diversos sistemas. Diseñado para facilitar el tratamiento informático, transmisión y

# Capítulo I: Fundamentación Teórica

---

Esta gestión de archivos se subdivide en las siguientes partes (12):

- o Almacenamiento de archivo: es de bajo nivel sin acceso a información de control. Almacena el contenido de archivos en disco con los metadatos en las tablas de base de datos asociadas.
- o Entrega de archivo: permite a los usuarios obtener los archivos a través de solicitudes y con el apropiado chequeo de seguridad.
- o Interfaces de usuario relacionadas a archivos: provee la interfaz de acceso a estos archivos. Pueden ser elementos de formularios que permitan a los usuarios seleccionar los archivos utilizando el API<sup>3</sup> de repositorios y que estos se almacenen dentro de Moodle, sustituyendo a la antigua interfaz para los archivos de curso.
- o API para la búsqueda de archivos: permite buscar y opcionalmente manipular las áreas de archivos para encontrar información acerca de los que estén disponibles en cada una de estas, así como mostrar vínculos a los archivos y opcionalmente gestionarlos.

La capa de acceso a la base de datos en Moodle ha sido creada utilizando la librería de PHP ADOdb<sup>4</sup>, que proporciona un método estándar de acceso a varios sistemas de bases de datos, utilizando una interfaz de programación consistente. Su utilización proporciona a Moodle soporte para varias bases de datos entre las que se incluyen: MySQL, PostgreSQL, Microsoft SQL y Oracle.

A diferencia de la versión 1.9.4, Moodle 2.1 utiliza el paradigma de Programación Orientada a Objetos (POO), garantiza un alto porcentaje de compatibilidad con las bases de datos existentes y reorganizan librerías fundamentales; ADOdb, PDO<sup>5</sup>,

---

visualización de textos de múltiples lenguajes y disciplinas técnicas además de textos clásicos de lenguas muertas. (53)

<sup>3</sup> API (Interfaz de Programación de Aplicaciones): es una interfaz para dar un acceso limitado a la base de datos de un servicio web, evitando que se conozca o acceda al propio código fuente de la aplicación original. (51)

<sup>4</sup> ADOdb: conjunto de librerías de bases de datos para PHP y Python, que permite a los programadores desarrollar aplicaciones web de una manera portable, rápida y fácil. La ventaja reside en que la base de datos puede cambiar sin necesidad de reescribir cada llamada a la base de datos realizada por la aplicación. (48)

<sup>5</sup> PDO (PHP Data Objects): extensión que permite acceder a distintas bases de datos utilizando las mismas funciones, lo que facilita la portabilidad. En PHP 5 existen drivers para acceder a las bases de

# Capítulo I: Fundamentación Teórica

---

simulando una capa de abstracción que evita la interacción directa con los desarrolladores. Los objetos de la base pueden convertirse en subclasses: al realizar las pruebas de unidad para comprobar el acceso a la base de datos, el entorno de trabajo puede sobrescribir los métodos necesarios. El soporte para parámetros de tipo “?” y “:param”, independientemente de los tipos de parámetros soportados por cada uno de los drivers<sup>6</sup>.

En Moodle 2.1.x aparecen cambios en la sintaxis de las funciones disponibles para el acceso a los datos y el manejo de las estructuras de la base de datos, en función de recuperar y modificar su contenido, y tener un grupo bien definido de funciones capaces de manejar todas las estructuras de la base de datos usando una descripción neutral respectivamente. Todas las llamadas a estas funciones son métodos públicos, ahora accesibles a través del objeto global \$DB.

Se remueven funciones innecesarias, como *addslashes()* utilizado para escapar caracteres, con el objetivo de privar a los datos de la necesidad de añadir o eliminar los *slashes* (“/”) en todo el código.

Moodle en sus versiones 2.1.x propone mejoras importantes a características básicas existentes y módulos de actividades.

**Bloques:** están implementados de forma consistente en cada página en Moodle. No hay límite para las regiones de los bloques (además de derecha e izquierda, ponerlos arriba, al centro o abajo de las páginas). Cualquier bloque puede ser hecho "pegajoso" (aparecer en todos los contextos debajo, por ejemplo en todos los cursos o en todas partes de un curso). Los bloques pueden ser acoplados o dejados fijos en un lado de la pantalla (si el tema lo soporta).

**Lecciones:** su código interno fue reescrito y sus formularios ahora son estándares de Moodle.

**Recursos:** todos han sido implementados en módulos de actividades de la plataforma. Archivo, para mostrar un archivo con formatos soportados (como un mini-sitio en *HTML*). Carpeta, para mostrar una colección de documentos. URL, para mostrar una

---

datos más populares (MySQL, Oracle, MS SQL Server, PostgreSQL, SQLite, Firebird, DB2, Informix, entre otras. (49)

<sup>6</sup> Driver: también conocido como controlador, es un programa que controla un dispositivo. Actúa como un traductor entre el dispositivo y los programas que este utiliza. (50)

# Capítulo I: Fundamentación Teórica

---

página con un URL<sup>7</sup> dado. Página, para una sola página, editada en línea usando el editor *HTML*. IMS, para mostrar un paquete de contenido en formato IMS. Se mejoró el soporte, que cumple el estándar *XHTML*, para marcos o *frames*, *iframes* y encapsulamiento en todos estos módulos.

**SCORM:** aparecen nuevos parámetros del módulo SCORM como mostrar estado de intentos y estructura del curso, forzar completado y forzar nuevo intento, bloquear luego de intento final; permitiendo cumplir el comportamiento dictado por los objetos SCORM del paquete original. Aparece una nueva interfaz de reportes incluyendo tabla ordenable y colapsable con caja de selección de grupos y la posibilidad de descargar en Excel, ODS y formato de texto. Mejoras en la interfaz del reproductor de lecciones que incluyen elementos como la navegación, el desempeño y el manejo del tamaño del estado.

**Wiki:** fue completamente reescrito y muestra una interfaz mejorada.

**Taller:** este módulo fue totalmente reimplementado y ofrece una interfaz mejorada ampliamente para manejar escenarios y usuarios.

**Módulo de cuestionarios y banco de preguntas:** se le realizaron mejoras en la navegación de cuestionarios para estudiantes, en los informes del cuestionario, especialmente recalificación y análisis de ítems, así como en el informe de estadísticas y en la interfaz de edición de preguntas. Aparece el marcado de preguntas durante un intento de resolver un cuestionario. Se activan preferencias (fechas de apertura / cierre, número de intentos, claves, límite de tiempo) para cada grupo o estudiante. Se realiza el etiquetado de preguntas y mejora la búsqueda en el banco de preguntas. Los cuestionarios con preguntas asignadas aleatoriamente de otros cuestionarios. Aparecen nuevos tipos de pregunta como calculada simple y calculada opción múltiple.

**Cambios de la API del tipo de preguntas:** anteriormente se proveía el soporte para los tipos de preguntas con cadenas de lenguaje en el archivo `quiz.php`. Ahora se obtiene a través del archivo `qtype_xxx.php`, de forma similar se realiza para los archivos de ayuda. El parámetro categoría es pasado a la función `save_cuestion()`. La mayoría de los tipos de preguntas no necesitan sobrescribir este método a diferencia

---

<sup>7</sup> URL (Uniform Resource Locator): significa localizador uniforme de recurso y se refiere a la secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet para su localización o identificación; la dirección única que identifica a una página web en Internet. (52)

de Moodle 1.9 donde era necesario hacer un procesamiento complejo entre las propiedades categoría y categoría a trasladar. Nueva página de administración para los tipos de preguntas y nuevos métodos para metadatos. (12)

A partir de lo anteriormente expuesto, se concluye que los cambios en la arquitectura de Moodle 2.1.x implican una nueva forma de diseño de las acciones formativas, en el desarrollo y manejo de las estructuras de la plataforma; y afectan el funcionamiento de la mayoría de sus componentes, impidiendo la reutilización o actualización de extensiones implementadas para Moodle 1.9.4 sobre la nueva arquitectura, como es el caso del bloque C2SCORM 2.0.

## ***1.6 Estándares de interoperabilidad***

Una creciente avalancha de plataformas e-Learning se desplaza aparejada a la extensión de Internet. Para lograr la compatibilidad y el intercambio de recursos entre estas surgen estándares que constituyen normas comunes que rigen el desarrollo de protocolos, servicios y contenidos.

Los estándares pueden ser aprobados por un organismo oficial de estandarización o por voluntad propia o conveniencia sin ser aprobados por un organismo. Al estandarizar los contenidos se asegura la accesibilidad, reusabilidad, extensibilidad, perdurabilidad y la interoperabilidad de los mismos. Estas características garantizan la disponibilidad y flexibilidad de los contenidos ante cambios o ampliaciones en la plataforma.

La interoperabilidad se refiere a la capacidad de diferentes sistemas informáticos, aplicaciones y servicios para comunicar, compartir e intercambiar datos, información y conocimiento de una forma precisa, efectiva y consistente; para integrarse y funcionar de forma correcta con otros sistemas, aplicaciones y servicios, y ofrecer nuevos productos electrónicos. (13)

El contenido debería ser independiente de la herramienta o plataforma, de tal manera que se puedan utilizar diferentes sistemas para acceder a un mismo contenido.



Algunos de los estándares más empleados en la reutilización de contenidos son IMS<sup>8</sup> Learning Design (IMS LD), Modelo de Referencia para Objetos de Contenido Compartido (SCORM), Dublin Core y Learning Object Metadata (LOM).

## 1.6.1 Estándar para la descripción de contenidos

Los metadatos 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 objetos de aprendizaje). Estas bibliotecas soportarán, entonces, consultas significativas que permitirán la recuperación de aquellos materiales almacenados que cubran una determinada necesidad pedagógica.

La utilidad de un esquema de metadatos radica en su aceptación por una comunidad suficientemente amplia de productores y consumidores de material educativo. Efectivamente, si dos comunidades utilizan esquemas de metadatos distintos, difícilmente los materiales producidos podrán coexistir en un mismo repositorio, a menos que se haya encontrado previamente un consenso que permita homogeneizar los metadatos utilizados por ambas comunidades (por ejemplo, transformándolos a un esquema común).

Desde la comunidad de e-Learning se han realizado distintos esfuerzos para estandarizar los esquemas de metadatos que deben ser utilizados en la producción de contenidos educativos. El esfuerzo más prometedor ha desembocado en el estándar IEEE LOM que también ha sido adoptado en una versión preliminar como una especificación de descripción de metadatos por IMS. (14)

### Learning Object Metadata

Metadatos de los objetos de aprendizaje (del inglés Learning Object Metadata, LOM) es un modelo de datos para la comunicación de los objetos de contenido (Data Model for Content Object Communication) desarrollado por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, por sus siglas en inglés).

---

<sup>8</sup> IMS (Instructional Management System Project): organización mundial sin ánimo de lucro que incluye más de 50 miembros y afiliados y cuya misión es desarrollar y promover especificaciones abiertas para la tecnología de aprendizaje. (17)

LOM establece un esquema de datos conceptual que define la estructura de un registro de metadatos para un objeto de aprendizaje. La utilización de este esquema permite a los autores de objetos de aprendizaje especificar qué elementos componen una instancia de metadatos, con el propósito de facilitar la búsqueda, evaluación, adquisición y uso de objetos de aprendizaje por parte de los estudiantes, profesores o sistemas automatizados, así como su intercambio y uso compartido.

Este estándar ha ganado prestigio a nivel internacional por su estabilidad, además, es el definido por SCORM para realizar el empaquetamiento de contenidos, características por las cuales es seleccionado para la realización del presente trabajo.

## **1.6.2 Estándar para el empaquetamiento de contenidos.**

Empaquetamiento, según Luis A. Álvarez en “Empaquetamiento y Visualización de Objetos de Aprendizaje SCORM en LMSs de Código Abierto”, consiste en etiquetar el contenido de modo que pueda ser reconocido como tal, por el LMS y permita su carga en el sistema. (15)

De lo anterior se puede derivar que un estándar de empaquetamiento es un conjunto de normas y reglas que determinan la forma de etiquetar un contenido. Un ejemplo de estos estándares es SCORM 2004 utilizado en la versión 2.0 del bloque C2SCORM.

## **Sharable Content Object Reference Model (SCORM)**

SCORM (Modelo de Referencia de Objetos de Contenido Compartido) es desarrollado por ADL (del inglés Advanced Distributed Learning - Aprendizaje Avanzado Distribuido) es una iniciativa del departamento de defensa de los Estados Unidos de América.

SCORM se puede definir según <http://www.adlnet.org>, como un conjunto de especificaciones técnicas interrelacionadas construidas teniendo en cuenta el trabajo de otras organizaciones de estándares como son AICC<sup>9</sup>, IMS e IEEE con la finalidad de crear un modelo de contenido unificado. Estas especificaciones facilitan el re-uso de contenidos de aprendizaje a través de múltiples plataformas y productos e-Learning. Establece las pautas para describir, empaquetar, entregar, ejecutar y auditar contenidos por los diferentes sistemas. (16)

---

<sup>9</sup>Autonomous Intelligent Cruise Control o Comité de Entrenamiento Basado en Ordenador de la Industria de la Aviación, desarrolla especificaciones técnicas, dentro de la comunidad de formación de la aviación, para el entrenamiento asistido por ordenador. (17)

Este modelo cuenta con guías que permiten el desarrollo e implementación de una enseñanza distribuida eficaz. SCORM permite además, la organización de contenidos soportando la descripción de itinerarios formativos, secuenciación de contenidos, el empaquetamiento de los contenidos para su cómoda distribución, y el seguimiento del proceso de aprendizaje. “Se basa en la especificación IMS de paquetes de contenido y proporciona requisitos explícitos y guías de implementación para empaquetar Assets<sup>10</sup>, Objetos de Contenido Compartibles<sup>11</sup> (SCOs) y las agregaciones de contenido.” (17)

El modelo pretende cumplir con los requisitos de accesibilidad, reusabilidad, perdurabilidad e interoperabilidad de los componentes de enseñanza. Las especificaciones se encuentran en “libros técnicos” que las agrupan en tres temas principales (10):

- o El manual CAM (Modelo de Agregación de Contenidos) de SCORM describe los componentes usados en una experiencia de aprendizaje<sup>12</sup>, cómo hacer paquetes de esos componentes para que se intercambien entre sistemas, cómo describir esos componentes para permitir que se busque dentro de ellos y que sean descubiertos y cómo definir la información de secuencia para los componentes. El Modelo de Agregación de Contenidos de SCORM promueve la consistencia en el almacenaje, etiquetado, empaquetado, intercambio y descubrimiento del contenido de aprendizaje. (18)
- o El manual del Ambiente de Desempeño (RTE) de SCORM describe los requisitos del LMS para administrar el Ambiente de Desempeño. Significa que el proceso de lanzamiento de contenido, la comunicación estandarizada entre el contenido y el LMS, y los elementos estandarizados del modelo de datos usados para transmitir información relacionada con la experiencia del estudiante con el contenido. El manual RTE también cubre los requisitos de los Objetos de Contenido Compartido (SCO) y cómo usan la Interfaz de Programación de Aplicaciones (API) y el modelo de datos del Ambiente de Desempeño de SCORM. (18)

---

<sup>10</sup> Assets: representación electrónica de medios, texto, imágenes, sonidos, páginas web, objetos de evaluación u otras piezas de datos que puedan ser procesadas en un cliente web y entregadas al estudiante. (18)

<sup>11</sup> Objetos de Contenido Compartibles: recopilación de uno o más assets que representan un recurso de Aprendizaje individual que puede ser lanzado y que usa el Ambiente de Desempeño de SCORM para comunicarse con el LMS.

<sup>12</sup> Experiencia de aprendizaje: forma de organizar el aprendizaje significativo del estudiante haciéndolo con acciones concretas, co-responsable de su propio aprendizaje (55). SCORM por su parte la define en sus manuales como la interacción del estudiante con los objetos de contenido.

# Capítulo I: Fundamentación Teórica

---

- o El manual de Secuencia y Navegación (SN) de SCORM describe cómo se puede secuenciar el contenido SCORM certificado por ADL a través de un conjunto de eventos de navegación iniciados por el estudiante o el sistema. La división y el flujo de ese contenido se pueden describir con un sistema predefinido de actividades. (19)

SCORM 2004 introduce cambios a las versiones anteriores entran en varias categorías: aclaración de conceptos, aclaración de requisitos, cambios debido a los esfuerzos de estandarización / especificación, las mejores prácticas de la Comunidad ADL, mejoras y solución de problemas.

Uno de los esfuerzos principales que están detrás de los cambios a SCORM ha sido la evolución de las especificaciones y estándares subyacentes en SCORM 2004 (18):

- o Modelo de Datos para la comunicación de Objetos de Contenido de IEEE.
- o Interfaz de Programación de la Aplicación ECMAScript de IEEE para Contenido a Comunicación de Servicios de Ejecución.
- o Metadatos del Objeto de Aprendizaje (LOM, por sus siglas en inglés) de IEEE.
- o Enlace de Esquema de Extensible Markup Language (XML) de IEEE para el Modelo de Datos de Metadatos del Objeto de Aprendizaje.
- o Paquetes de Contenido de IMS.
- o Secuencias Simples de IMS.

SCORM actualmente ofrece una Interfaz de Programación de Aplicaciones (API) para comunicar información sobre la interacción del aprendiz con los objetos de contenido, un modelo de datos definido que representa esta información, especificaciones para los paquetes de contenido que permitan la interoperabilidad del contenido de aprendizaje, un conjunto estándar de elementos de metadatos que se puedan utilizar para describir el contenido de aprendizaje y un sistema de reglas de secuencia estándar que se pueden aplicar a la organización del contenido de aprendizaje.

El SCORM también describe cómo interpretan los Sistemas de Administración de Aprendizaje SCORM certificados por ADL, las reglas de secuencia expresadas por un programador de contenido unidas al conjunto de eventos de navegación iniciados por el aprendiz o por el sistema y sus efectos sobre el ambiente de ejecución. Define un método para representar el comportamiento previsto de una experiencia de

aprendizaje de autor de modo que cualquier Sistema de Administración de Aprendizaje SCORM certificado por ADL secuenciará las actividades de aprendizaje descritas de una manera consistente. (19)

Permite desarrollar contenidos más dinámicos en los que la experiencia de aprendizaje depende del desempeño del estudiante sin que el profesor pierda la posibilidad de controlar la forma y el orden en que estos se apropian del contenido. Describe cómo se pueden originar y procesar eventos de navegación iniciados por el aprendiz o por el sistema, dando como resultado la identificación de las actividades de aprendizaje que se entregarán. Cada actividad de aprendizaje identificada para la entrega tendrá un objeto de contenido asociado. (18)

Para el desarrollo de la solución de la presente investigación se utilizarán los estándares LOM 1.0 y SCORM 2004 4ta Edición.

## **1.7 Tecnologías a utilizar en el desarrollo del sistema**

Diversas son las tecnologías a emplear en la creación de productos web. La plataforma de teleformación Moodle, sobre la cual se desarrolla la presente investigación, es una aplicación web escrita con el lenguaje script<sup>13</sup> PHP y que utiliza una base de datos SQL para almacenar información. Como sistema modular está estructurado por un núcleo y extensiones que proveen funcionalidades específicas, estas últimas incluyen archivos CCS y JavaScript. La información mostrada por Moodle debe acatar las restricciones del estándar XHTML Strict 1.0, lo que implica un XML bien formado. Las tecnologías anteriormente citadas, que a su vez fueron las empleadas en el desarrollo de la versión anterior del bloque, serán utilizadas para implementar la solución de la presente investigación.

### **1.7.1 Tecnologías del lado del servidor**

Estas tecnologías proporcionan un entorno rápido de creación de scripts y soporte para los estándares más importantes, posibilitando la integración con bases de datos. No tienen en cuenta el tipo de cliente; la aplicación se ejecuta en el servidor que es un ambiente controlado, una vez ejecutada la aplicación, el resultado que se envía al cliente puede estar en un formato estandarizado que cualquier cliente puede interpretar. (10)

---

<sup>13</sup> Script: son lenguajes que no necesitan ser compilados. Una máquina servidor es capaz de interpretar el código fuente y ejecutarlo, lo que hace de estos lenguajes la mejor solución para programación en entornos web. (54)

## **Preprocessed Hypertext Pages (PHP)**

PHP es un lenguaje script utilizado para el desarrollo web y puede ser embebido en páginas *HTML*. Con una sintaxis similar a C, Java y Perl, este lenguaje hace posible el desarrollo de páginas web que se generen dinámicamente y de forma rápida, y puede ejecutarse en distintos tipos de servidores web (20).

Por su naturaleza de código abierto se muestra como una alternativa de fácil acceso al software privativo. Es un lenguaje sencillo para el aprendizaje, a partir de las últimas versiones pasó a ser un lenguaje orientado a objetos y es capaz de conectarse con la mayoría de los gestores de bases de datos que se utilizan en la actualidad. (10)

## **eXtensible Markup Language (XML)**

XML (“Lenguaje de Marcas Extensible”), metalenguaje extensible de etiquetas que fue desarrollado por el World Wide Web Consortium (W3C)<sup>14</sup>. Es una adaptación del SGML<sup>15</sup> que permite la organización y el etiquetado de documentos. Por lo que XML no es un lenguaje en sí mismo, sino un sistema para definir lenguajes de acuerdo a las necesidades. El *XHTML*, el *MathML* y el *SVG* son algunos de los lenguajes que el XML está en condiciones de definir.

Las bases de datos, los documentos de texto, las hojas de cálculo y las páginas web son algunos de los campos de aplicación del XML. El metalenguaje aparece como un estándar que estructura el intercambio de información entre las diferentes plataformas. Al ser extensible y contar con un analizador estándar se pueden añadir nuevas etiquetas tras el diseño del documento, no requiere cambios para cada versión del metalenguaje y facilita el análisis y procesamiento de los documentos XML creados por terceros. (21)

---

<sup>14</sup>Consortio internacional que elabora recomendaciones para la World Wide Web: el sistema de documentos de hipertexto que se encuentran enlazados entre sí y que son accesibles a través de Internet. Mediante un software conocido como navegador, los usuarios pueden visualizar las páginas web (que contienen texto, imágenes, videos y otros contenidos multimedia) y navegar a través de ellas mediante los hipervínculos. (21)

<sup>15</sup> SGML o Standard Generalized Markup Language fue definido por la norma ISO 8879 en 1986 y desde entonces ha sido considerado el lenguaje estándar para mantener los depósitos centrales de la estructura documental. Se trata de un lenguaje para marcar y describir documentos con independencia total del hardware y software utilizados. (46)

## 1.7.2 Tecnologías del lado del cliente

Estas tecnologías son ejecutadas en el navegador del usuario el cual soporta la carga de procesamiento y funcionamiento. Las páginas clientes dependen, por tanto, de las características del sistema en el que se ejecutan.

### JavaScript

Es un lenguaje interpretado basado en guiones que son integrados directamente en el código *HTML*. Es utilizado en el desarrollo web: el código es transferido al cliente para que este lo interprete al cargar la página. Con este lenguaje no pueden crearse programas independientes (22). Permite agregar animaciones y efectos dotando de dinamismo a la página web al interactuar con el navegador.

### HyperText Markup Language (*HTML*)

*HTML* (Lenguaje de Marcado de Hipertexto) es la lengua franca para publicar hipertexto en la World Wide Web. Se trata de un formato no propietario basado en SGML, y se puede crear y procesar una amplia gama de herramientas, desde simples editores de texto a sofisticadas herramientas de creación en modo WYSIWYG<sup>16</sup>. Emplea etiquetas de texto en la estructura de encabezados, párrafos, listas, enlaces de hipertexto, entre otros. (23)

### Cascading Style Sheets (*CSS*)

Hojas de Estilo en Cascada (*CSS* por sus siglas en inglés), considerado lenguaje de especificación<sup>17</sup>, es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos. (24)

---

<sup>16</sup>What You See Is What You Get: Un editor o programa WYSIWYG permite a los desarrolladores ver el resultado final de determinado trabajo mientras la interfaz o el documento se está creando. WYSIWYG es un acrónimo de "lo que ves es lo que obtienes". Puede ser contrastado con editores más tradicionales que requieren que los desarrolladores entren los códigos descriptivos (o etiquetas) y no permiten, de una manera inmediata, ver los resultados de las etiquetas. (47)

<sup>17</sup>Lenguajes utilizados para especificar el sistema, conceptualizarlo o incluso validarlo, aunque también suelen ser legibles para un programa de computadora, que puede asistir en el proceso de validación.

CSS se utiliza para dar estilo a documentos *HTML* y *XML*, separando el contenido de la presentación. Permite a los desarrolladores web controlar el estilo y el formato de múltiples páginas web al mismo tiempo. Funciona a base de reglas, declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento *HTML* o *XML*. La regla tiene dos partes: un selector y la declaración. A su vez la declaración está compuesta por una propiedad y el valor que se le asigne (24)

Luego de analizar las diversas tecnologías y herramienta a emplear para la implementación de la presente investigación se determinó utilizar PHP 5.3.6, CSS 2.0, *XHTML* 1.0 Strict y JavaScript.

## **1.8 Herramientas a utilizar en el desarrollo del sistema**

### **Entorno de Desarrollo Integrado (IDE) NetBeans 6.9**

El IDE NetBeans es un entorno de desarrollo, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio 2000 y continúa siendo el patrocinador principal de los proyectos.

Está escrito en Java pero puede servir para cualquier otro lenguaje de programación. Existe además, un número importante de módulos para extender el NetBeans IDE. El cual es un producto libre y gratuito sin restricciones de uso.

Esta plataforma permite desarrollar aplicaciones a partir de componentes de software llamados módulos: archivos Java que contiene las clases para interactuar con las APIs de NetBeans y un archivo especial que lo identifica como módulo. Los desarrolladores pueden crear rápidamente sitios web, aplicaciones de escritorio y aplicaciones móviles utilizando la plataforma Java, así como JavaFX, PHP, JavaScript y Ajax, Ruby, Ruby on Rails, Groovy, Grails, y C / C + +.

El código fuente está disponible para su reutilización de acuerdo con la Common Development and Distribution License (CDDL) v1.0 y la GNU General Public License (GPL) v2. (25).

Este IDE aporta mejoras en cuanto a rendimiento con respecto a otras plataformas y cuenta con un completamiento de código superior a otros entornos, lo que economiza el tiempo de los desarrolladores. (26) Por las características antes expuestas y por ser



una herramienta libre, que ofrece una amplia documentación y formación de recursos, se selecciona NetBeans para implementar la solución de la presente investigación.

## **1.9 Metodología de desarrollo de software**

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos de software. Indican Quién debe hacer Qué, Cuándo y Cómo debe hacerse. Detallan la información que se debe producir como resultados de una actividad y la información necesaria para comenzarla.

Surgen por la necesidad de elevar la calidad del software y la eficacia y eficiencia de su proceso de desarrollo. No existe una metodología universal, sino una amplia gama de esta que nos permite seleccionar la que más se ajuste a los requerimientos del software a desarrollar. Razón por la cual es primordial determinar el alcance del proyecto para luego seleccionar la más apropiada.

Algunas de las metodologías orientadas a objetos existentes en la actualidad son Rational Unified Process (RUP), Extreme Programming (XP) y Microsoft Solution Framework (MSF). [ (27), (10)]

La presente investigación pertenece al proyecto “Personalizaciones de Moodle”, al no ser un producto aislado, se dificulta entonces la utilización de metodologías ágiles, que aunque más centradas en el desarrollo, descuidan un tanto la organización y documentación del trabajo realizadas. Estas características indican que el proceso de desarrollo del software debe ser guiado por una metodología robusta que permita una correcta integración y una documentación que asegure el mantenimiento, crecimiento de la presente y otras extensiones. Para el desarrollo de la solución se decide utilizar la metodología RUP por lo anteriormente expuesto, y porque es la metodología de desarrollo de software que seleccionó el proyecto al cual tributa esta investigación.

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

Proceso para el desarrollo de un proyecto de software dirigido por casos de uso: significa que orientan el proyecto a la importancia para el usuario y lo que este quiere. Está centrado en la arquitectura: relaciona la toma de decisiones que indican cómo tiene que ser construido el sistema y en qué orden. Es un proceso iterativo e incremental por lo que divide el proyecto en mini proyectos donde los casos de uso y la arquitectura cumplen sus objetivos de manera más depurada. (28)

Está basado en componentes e interfaces bien definidas, y junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Unifica los mejores elementos de metodologías anteriores. Está preparado para desarrollar grandes y complejos proyectos. (29)

Puede especializarse para una gran variedad de sistemas de software, en diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.

Esta metodología organiza el desarrollo de software y sistemas en cuatro fases, cada una compuesta de una o más iteraciones ejecutable del programa en esa etapa de desarrollo. (28):

- o Inicio: en esta etapa se realiza un plan de fases, se identifican los principales casos de uso, se identifican los riesgos y define el alcance del proyecto.
- o Elaboración: se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.
- o Construcción: está concentrada en la elaboración de un producto totalmente operativo y el manual de usuario.
- o Transición: consiste en la instalación del producto en el cliente y el entrenamiento a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados.

## **Herramienta CASE**

Se puede definir a las Herramientas CASE<sup>18</sup>(Ingeniería de Software Asistida por Computadora) como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software. Fueron desarrolladas para automatizar esos procesos y facilitar las tareas de coordinación de los eventos que necesitan ser mejorados en el ciclo de desarrollo de software, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación [ (30), (31)].

---

<sup>18</sup>Computer Aided Software Engineering.

Existen varias herramientas CASE que permiten llevar a cabo la automatización del desarrollo de un software, las que se utilizan en la UCI son: Rational Rose para el Sistema Operativo (SO) Windows y Visual Paradigm para Linux. La propuesta de solución de la presente investigación se desarrolla sobre el Sistema Operativo Linux, por lo cual no se procede al análisis de otras herramientas. A continuación, se detallan las características de la herramienta Visual Paradigm que complementan el por qué de su utilización para modelar la solución.

## **Visual Paradigm 8.0**

Visual Paradigm es una herramienta de modelado visual para todo tipo de diagramas UML y Diagramas de entidad-relación (ERD) para el diseño de bases de datos. Es compatible con la gestión extensiva de casos de uso, diagramas de requisitos y el diseño de bases de datos y entrega los esfuerzos más eficaces en el análisis y diseño de sistemas. Existen varios juegos de herramientas de diagramación que ayuda a aumentar la eficiencia del sistema de análisis y diseño de forma significativa. (32)

Esta herramienta multiplataforma presenta las siguientes características (33):

- o Soporte de UML versión 2.1.
- o Modelado colaborativo con CVS y Subversion (nueva característica).
- o Ingeniería inversa - Código a modelo, código a diagrama.
- o Ingeniería inversa Java, C++, Esquemas XML, XML, NET exe / dll, CORBA IDL.
- o Generación de código - Modelo a código, diagrama a código.
- o Editor de Detalles de Casos de Uso - Entorno todo-en-uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.
- o Diagramas de flujo de datos.
- o Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- o Ingeniería inversa de bases de datos - Desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación.
- o Generador de informes para generación de documentación.

- o Distribución automática de diagramas.
- o Reorganización de las figuras y conectores de los diagramas UML.
- o Importación y exportación de ficheros XML.

## ***1.10 Conclusiones parciales***

Luego de analizar las principales novedades de Moodle 2.1.x respecto a su versión 1.9.4 y el funcionamiento del bloque C2SCOM 2.0, se puede concluir que este último es incompatible con las más actuales y estables versiones de la plataforma Moodle. Además, las especificaciones del estándar SCORM 2004 garantizan la interoperabilidad de los contenidos. Por su parte, los procesos de desarrollo de software emplean metodologías que los orientan durante su ciclo de vida.

## Capítulo II: Análisis y Diseño del sistema

### ***Introducción***

La inevitable evolución de la plataforma de teleformación Moodle ha provocado la transformación de la mayoría de sus componentes, por lo cual se imposibilita la reutilización de productos desarrollados para versiones anteriores en Moodle 2.1.x. Una de esas aplicaciones es el bloque C2SCORM 2.0 que permite exportar los contenidos de los cursos en forma de paquete SCORM 2004 para la versión 1.9.4 de la plataforma. En el presente capítulo se propondrá una solución al problema de la exportación de contenidos en formato SCORM para Moodle 2.1.x. Se describirán las principales características del sistema propuesto a través de los requisitos funcionales y no funcionales, los cuales son modelados en casos de uso, acompañados de sus respectivas descripciones textuales para su mejor comprensión y relacionados con los actores del sistema. Se definen las clases del análisis, el modelo de diseño y la estructura de la base de datos de la aplicación.

### ***2.1 Propuesta de solución***

Como solución al problema planteado se propone desarrollar un bloque para la plataforma que permita exportar los cursos en forma de paquete SCORM. Este será la versión 3.0 de C2SCORM 2.0 que extiende sus funcionalidades. Su implementación seguirá las especificaciones del estándar SCORM 2004 para empaquetar los contenidos, que indica la descripción de contenidos a través cinco categorías de las nueve del estándar LOM.

Respetando las especificaciones del estándar SCORM 2004 se exportarán, de cada material, los metadatos: Identificador, Título, Descripción y Palabras clave de la categoría General, Tipo de Recurso Educativo y Nivel de Interactividad de la categoría Educativa, la Versión y el Estado correspondientes al Ciclo de Vida y Costo, Descripción y Derechos de Copia / Otras restricciones de Derechos. Así como Localización y Formato de la categoría Técnica y el Esquema de Metadatos correspondiente Meta-metadatos.

Se incluyen otros metadatos no obligatorios para SCORM que se consideran de importancia para la descripción y localización del contenido. Esta aplicación permitirá al usuario autorizado exportar cursos teniendo en cuenta su ubicación en la plataforma al realizar la solicitud, incluir reglas de secuencia y navegación, editar los metadatos y las preferencias del exportador al realizar el proceso. Si el usuario es administrador, podrá, además, instalar, configurar y desinstalar el bloque.

En el Anexo 3 (Ver [Anexo 3](#)) se muestra un conjunto de módulos existentes para la versión 2.1.x de Moodle con una breve descripción de la estructura de la información exportada, especificando con una “x” si son instalados por defecto y algunas observaciones respecto al grado de semejanza que tienen, una vez exportados, con respecto a su estructura y funcionalidad en Moodle.

### **2.2 Modelo de Dominio**

La problemática planteada está altamente centrada en las tecnologías informáticas por lo que durante su análisis de no fue posible identificar procesos del negocio. Siendo necesario representar la situación mediante un Modelo de Dominio.

Este modelo es un diagrama que representa los objetos o entidades del proyecto que intervienen en el proceso como clases del dominio y sus relaciones. Facilita la comprensión de los conceptos del dominio que maneja el usuario.

El Modelo de Dominio correspondiente al sistema propuesto (Fig. 2.1) cuenta con las siguientes clases:

**Moodle:** sistema de gestión de aprendizaje donde el usuario puede acceder para exportar el curso que desea a paquetes SCORM.

**Exportador:** usuario que tiene los permisos necesarios para exportar los cursos de la plataforma Moodle a paquetes SCORM.

**Administrador:** usuario que posee privilegios para exportar un curso a paquete SCORM y configurar el bloque C2SCORM 3.0.

**Curso:** conjunto de módulos contemplados en la estructura de la plataforma Moodle, y documentos requeridos para la planificación y organización de una asignatura. Los cursos son creados por los administradores, creadores del curso o gerentes, quienes pueden añadir el contenido y reorganizarlo de acuerdo a sus propias necesidades.

**Actividad:** acción diseñada con el uso de herramientas informáticas de comunicación como foro, wiki, chat, talleres, entre otras. Las actividades en su conjunto conforman un curso dentro la plataforma de teleformación Moodle.

**Recurso:** conjunto de fuentes tecnológicas seleccionados por su valor académico para ser consultados en el transcurso de la impartición de las temáticas de una asignatura.

**Bloque C2SCORM 3.0:** bloque que se le incorpora a Moodle para permitir la exportación de los cursos de la plataforma en paquetes SCORM.

**Paquete SCORM:** paquete que genera el bloque C2SCORM 3.0 a partir de un curso exportado en la plataforma Moodle.

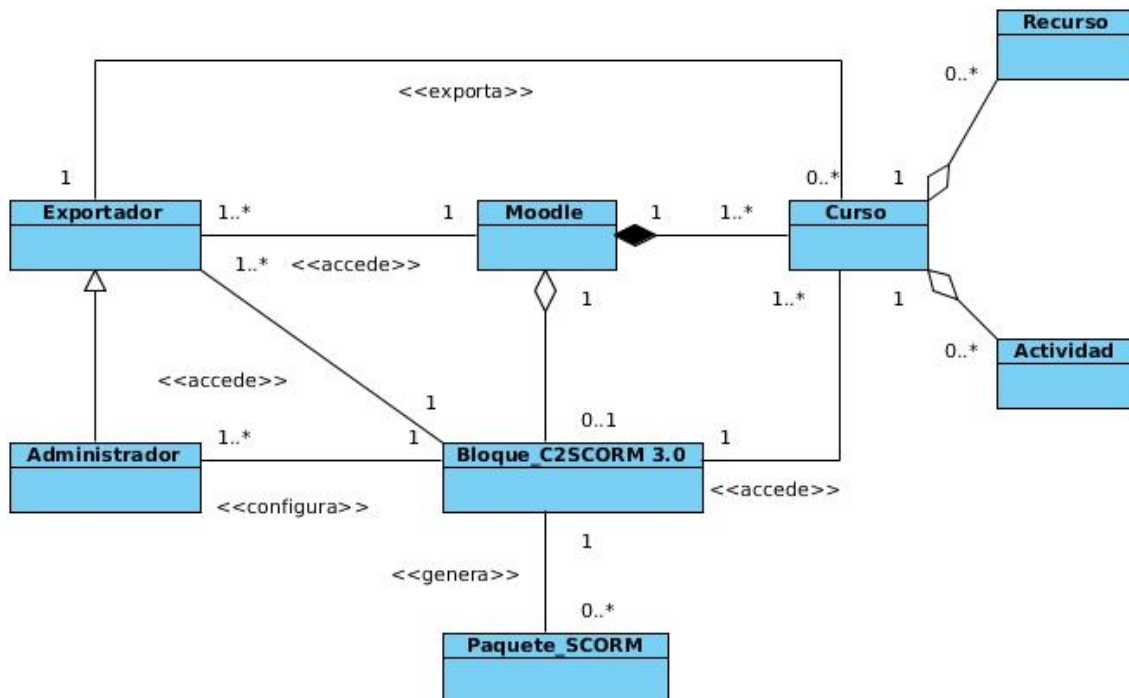


Fig. 2.1. Modelo de Dominio de C2SCORM 3. 0.

### 2.3 Especificación de Requisitos

Los requisitos son capacidades o condiciones que debe tener un sistema para satisfacer las necesidades de un usuario o resolver un problema, por lo cual se les debe prestar especial atención su especificación. Constituyen la base y la guía del proceso de desarrollo del software; de su cumplimiento depende la calidad del producto final.

Los requerimientos deben ser (34):

- o **Especificados por escrito:** como todo contrato o acuerdo entre dos partes.
- o **Posibles de probar o verificar:** si un requerimiento no se puede comprobar, entonces ¿cómo se sabe si se cumplió con él o no?
- o **Concisos:** un requerimiento es conciso si es fácil de leer y entender. Su redacción debe ser simple y clara para aquellos que vayan a consultarlo en un futuro.
- o **Completos:** un requerimiento está completo si no necesita ampliar detalles en su redacción, es decir, si se proporciona la información suficiente para su comprensión.
- o **Consistentes:** si no es contradictorio con otro requerimiento.
- o **No ambiguos:** cuando tiene una sola interpretación. El lenguaje usado en su definición, no debe causar confusiones al lector.

Para identificar los requerimientos del presente sistema se analizaron otros similares ya desarrollados. Esta técnica es conocida como Sistemas Existentes y permite extraer información al interactuar con las interfaces de usuario, observando el tipo de información que se maneja y cómo es manejada, así como las distintas salidas que los sistemas producen. (34)

### 2.3.1 Requisitos Funcionales

Los requerimientos funcionales son los que definen las funciones que el sistema será capaz de realizar, describen las transformaciones que el sistema realiza sobre las entradas para producir salidas. Es importante que se describa el ¿Qué? y no el ¿Cómo? se deben hacer esas transformaciones. Estos requerimientos al tiempo que avanza el proyecto de software se convierten en los algoritmos, la lógica y gran parte del código del sistema. (34)

A continuación se describen los requisitos funcionales que debe cumplir el presente sistema:

**RF-1. Exportar cursos:** el sistema debe exportar los cursos de la plataforma Moodle en formato por semanas o por temas en paquetes SCORM 2004.

**RF-1.1. Atender solicitud del usuario:** el sistema debe recopilar toda la información necesaria para exportar un curso personalizado por el usuario. Al recibir la solicitud de



## Capítulo II: Análisis y Diseño del sistema

---

exportar un curso, debe verificar a partir de qué contexto se está haciendo la solicitud, ya sea desde la página principal o desde el propio curso a exportar.

**RF-1.2. Mostrar actividades y recursos del curso a exportar:** el sistema debe permitir al usuario escoger las actividades y recursos que desea exportar dentro del curso seleccionado.

**RF-1.3. Visualizar las actividades y recursos:** el sistema debe ser capaz mostrar una vista previa de la estructura del paquete y de cada actividad o cada recurso.

**RF-1.4. Procesar preferencias escogidas:** el sistema utiliza las preferencias del usuario para crear la ruta y obtener las instancias de los módulos que serán exportados.

**R-1.5. Generar paquete SCORM:** a partir de la información reunida del curso, el sistema debe generar el paquete SCORM.

**RF-1.6. Descargar paquete SCORM:** el sistema debe permitir la descarga del paquete SCORM previamente exportado y brindar la posibilidad al usuario de escoger la ruta para descargar el paquete SCORM.

**RF-2. Modificar metadatos de las actividades y recursos:** el sistema debe permitir al usuario la modificación de los metadatos generados.

**RF-3. Incluir reglas de secuencia y navegación:** el sistema debe permitir al usuario incluir reglas de secuencia y navegación a las actividades y recursos del curso a exportar.

**RF-4. Configurar bloque:** el sistema debe permitir que el usuario pueda configurar el bloque C2SCORM 3.0. Si la configuración aún no ha sido personalizada por el usuario, el programa o sistema cargará la configuración por defecto (predeterminada). Una configuración predeterminada es aquella que no se ha redefinido, generalmente es la más recomendada. A continuación se muestran las configuraciones con que contará el bloque a desarrollar.

**RF-4.1. Configurar proceso de exportación para las actividades y secciones:** el sistema debe permitir al usuario elegir si las secciones que no tengan actividades disponibles para exportar, así como, que las secciones y actividades ocultas, entren en el proceso de exportación.

**RF-4.2. Configurar proceso de exportación para los Foros:** el sistema debe permitir al usuario decidir si se van a exportar las discusiones de los foros, las imágenes de los usuarios en los post y los ficheros adjuntos de los mensajes.

**RF-4.3. Configurar proceso de exportación para los Glosarios:** el sistema debe permitir al usuario decidir si se van a exportar los alias y ficheros adjuntos en las entradas del glosario, así como, las calificaciones de las entradas del glosario.

### 2.3.2 Requisitos No Funcionales

Los requerimientos no funcionales tienen que ver con características que de una u otra forma puedan limitar al sistema, como por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad y estándares. (34)

El sistema en cuestión debe respetar los siguientes requisitos no funcionales:

#### Usabilidad

- o Utilizar íconos sugerentes para lograr que el usuario encuentre lo que busca en el menor tiempo posible.
- o Mostrar textos explicativos al pasar el puntero por encima de los íconos.

#### Mantenimiento

- o Utilizar estándares para el desarrollo de aplicaciones web (*XHTML* 1.0 Strict, CSS 2.0).

#### Portabilidad

- o Al sistema se debe acceder desde cualquier sistema operativo.
- o Al sistema se debe acceder desde cualquier navegador.

#### Seguridad y privacidad

Confidencialidad:

- o La información manejada por el sistema debe estar protegida ante el acceso no autorizado y la divulgación. El acceso al sistema se restringirá a los usuarios de acuerdo al rol que desempeñen. El sistema debe garantizar que datos sensibles no viajen en texto plano por la red.

Integridad:

- o La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes, de la misma forma será considerada igual la fuente o autoridad de los datos.

Eficiencia:

- o La herramienta propuesta debe ser rápida y el tiempo de respuesta debe ser el mínimo posible (6 segundos máximo), adecuado a la rapidez con que el usuario requiere la respuesta a su acción.

### **Restricciones de diseño e implementación**

- o El sistema se implementará con tecnología PHP 5.3.6 utilizando como entorno de desarrollo al IDE NetBeans 6.9.

### **Requisitos para la documentación de usuarios en línea y ayuda del sistema**

Ayuda y documentación en línea:

- o El cliente podrá contar con un manual de usuario.
- o Ventanas emergentes.

### **Interfaces de usuario**

Apariencia o interfaz externa:

- o Cada página no debe exceder 100kb en las imágenes para no afectar la rapidez de respuesta de la misma.
- o Se deben tener en cuenta algunos elementos de diseño como gráficos de encabezamiento, estilos y formatos de texto establecidos para no afectar el diseño de Moodle.
- o No debe usarse tecnología de *frames* para evitar problemas de visualización en los navegadores, de indexación en buscadores y de usabilidad.

### **Interfaces de hardware**

Estación de trabajo Servidor

- o Microprocesador 2.0 GHz

- o 1 GB de RAM.
- o Espacio disponible en disco: 4 GB.

Estación de trabajo Cliente

- o Microprocesador 200 MHz.
- o 128 MB de RAM.

### Requisitos de Licencia

- o La plataforma escogida para el desarrollo de la aplicación está basada en GNU / GPL 3.0.

### ***2.4 Modelo de Casos de Uso del Sistema***

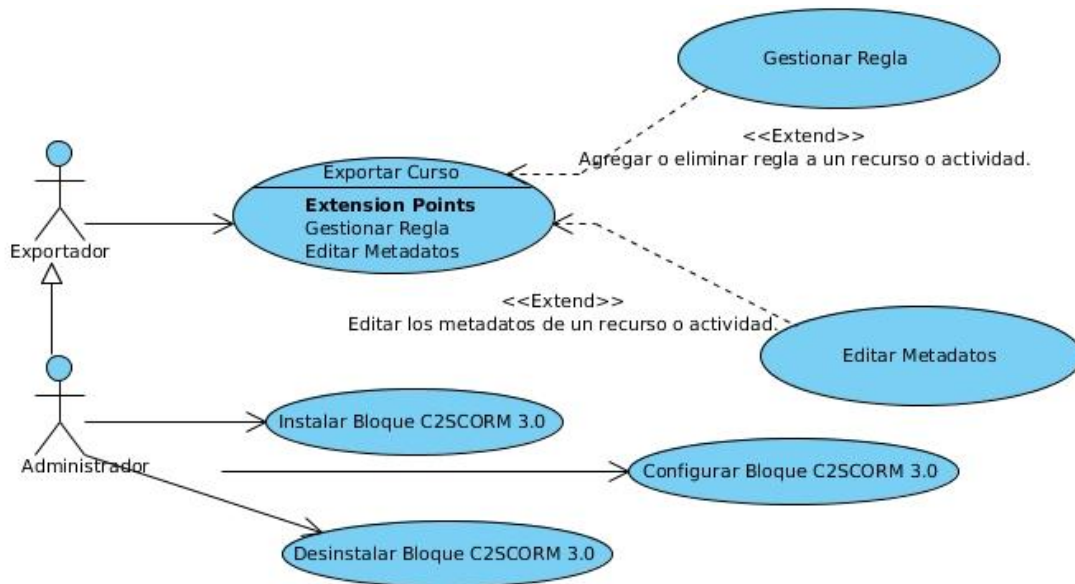
El Modelo de Casos de Uso del Sistema representa las funcionalidades y acciones de los usuarios sobre el sistema a desarrollar a través de los casos de uso y actores, que pueden ser tanto personas como máquinas cuyo rol o autoridad les permiten iniciarlos.

Los actores del presente sistema son los siguientes:

**Tabla 2.1.** Actores del sistema.

<b>Actores</b>	<b>Descripción</b>
Administrador.	Usuario con privilegios para instalar, configurar y desinstalar el bloque, así como de exportar un curso en forma de paquete SCORM.
Exportador.	Usuario con privilegios para exportar un curso en forma de paquete SCORM.

A continuación se muestran los Casos de Uso del Sistema (CUS) identificados a partir de los requisitos funcionales, así como los actores que los inicializan y las relaciones entre ellos.



**Fig. 2.2.** Modelo de Casos de uso de C2SCORM 3.0.

### 2.4.1 Descripción de Casos de Uso del Sistema

Las descripciones textuales de los casos de uso del sistema (CUS) facilitan la comprensión de estos. Muestra la secuencia de acciones que debe desarrollar el usuario para alcanzar un objetivo en la aplicación, así como la respuesta que esta dará a cada solicitud.

**Tabla 2.2.** Descripción del CUS Exportar Curso.

<b>Caso de uso:</b>	Exportar Curso.
<b>Actores:</b>	Exportador.
<b>Resumen:</b>	El caso de uso comienza cuando el exportador desea exportar un curso en la plataforma Moodle. Personaliza el proceso al seleccionar sus preferencias hasta que es generado el paquete SCORM y finaliza el caso de uso.
<b>Precondiciones:</b>	El usuario debe estar autenticado en la plataforma. Deben existir cursos en la plataforma Moodle. El exportador puede comenzar el proceso de exportación de un curso a partir de la página de

## Capítulo II: Análisis y Diseño del sistema

	inicio de la plataforma o encontrándose dentro del curso que desea exportar.
<b>Referencias:</b>	RF-1, RF-1.1, RF-1.2, RF-1.3, RF-1.4, RF-1.5, RF-1.6.
<b>Prioridad:</b>	Alta.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El exportador accede desde el curso que desea exportar al bloque C2SCORM 3.0.	2. Muestra, marcados por defecto, las actividades y recursos que contiene el curso. Brinda la posibilidad de organizarlos por secciones o por módulos para mejorar la selección de los mismos. Permite desmarcar aquellos que no se deseen exportar. Brinda las siguientes opciones: <ul style="list-style-type: none"> <li>o Continuar.</li> <li>o Cancelar.</li> </ul>
3. El exportador selecciona el tipo de organización por módulos.	4. Muestra recursos y actividades organizados por módulos.
5. Desmarca aquellos que no desee exportar. Luego selecciona la opción de continuar con el proceso de exportación.	6. Muestra los recursos y actividades organizados por secciones. Brinda las opciones: <ul style="list-style-type: none"> <li>o Editar metadatos (Ver CU Editar Metadatos).</li> <li>o Visualizar recursos y actividades.</li> <li>o Gestionar reglas de navegación y secuencia (Ver CU Gestionar Reglas).</li> </ul> Brinda las siguientes opciones: <ul style="list-style-type: none"> <li>o Continuar.</li> <li>o Cancelar.</li> </ul> Permite además continuar o cancelar el proceso.
7. Continúa con el proceso de exportación.	8. Construye el paquete SCORM que

## Capítulo II: Análisis y Diseño del sistema

	<p>contendrá las actividades y recursos del curso seleccionado, lo guarda en el sistema y le brinda la posibilidad al exportador de descargar el paquete hacia un directorio específico o de seguir navegando en la plataforma mediante las siguientes opciones:</p> <ul style="list-style-type: none"> <li>o Descargar.</li> <li>o Continuar.</li> </ul> <p>Muestra la información de los recursos y actividades que se están exportando.</p>
9. Desea descargar el paquete a un directorio específico.	10. En dependencia de la configuración del navegador del exportador, el paquete se guardará en un directorio específico o muestra un formulario para seleccionar la ruta del directorio en que se guardará el paquete SCORM creado.
11. El exportador selecciona la ruta deseada, para el caso en que el navegador no esté configurado, y selecciona la opción de descargar archivo.	12. Descarga el archivo en la ruta especificada y finaliza el caso de uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
1.a. El usuario selecciona Exportar desde un curso.	
1.a.1. El usuario selecciona, desde la página principal de Moodle, la opción Exportar curso.	1.a.2. Muestra las categorías y subcategorías existentes.
1.a.3. Selecciona la categoría deseada.	1.a.4. Muestra los cursos y subcategorías correspondientes a la categoría seleccionada.
1.a.5. Selecciona el curso que desea exportar.	1.a.6. Continúa en el punto 2 del flujo central de los eventos.

## Capítulo II: Análisis y Diseño del sistema

<b>3.a. El usuario organiza las actividades y recursos por secciones.</b>	
3.a.1. El exportador selecciona el tipo de organización secciones.	3.a.2. Muestra las actividades y recursos del curso organizados por secciones. Continúa en el paso 5 del flujo central de los eventos.
<b>3.b. El exportador selecciona la opción Cancelar.</b>	
3.b.1. El exportador cancela el proceso de exportación.	3.b.2. Muestra la página del curso seleccionado para exportar y finaliza el caso de uso.
<b>5.a. El exportador desmarca todos los campos.</b>	
5.a.1. El exportador desmarca todos los campos y selecciona Continuar.	5.a.2. Muestra un mensaje indicando que debe seleccionar al menos uno de los recursos o actividades.
<b>7.a. El exportador selecciona la opción visualizar un recurso o una actividad.</b>	
7.a.1. Desea visualizar un recurso o una actividad.	7.a.2. Muestra la vista del recurso o la actividad seleccionada en una página independiente.
7.a.3. El usuario navega y luego cierra la página de Vista Previa.	7.a.4. Muestra la Vista de Reglas y Metadatos. Continúa en el paso 6 del flujo central de los eventos.
<b>7.b. El usuario selecciona la opción Cancelar.</b>	
7.b.1. Selecciona la opción Cancelar.	7.b.2. Muestra la página del curso seleccionado para exportar y finaliza el caso de uso.
<b>9.a. El usuario selecciona la opción Continuar</b>	
7.a.1. Selecciona la opción Continuar.	7.a.2. Muestra la página del curso seleccionado para exportar y finaliza el caso de uso luego de eliminar el paquete generado.
<b>Poscondiciones:</b>	El paquete SCORM contiene el curso seleccionado exportado correctamente.



## Capítulo II: Análisis y Diseño del sistema

**Tabla 2.3.** Descripción del CUS Editar Metadatos.

<b>Caso de uso:</b>	Editar Metadatos (extendido).	
<b>Actores:</b>	Exportador.	
<b>Resumen:</b>	El caso de uso comienza cuando el exportador decide editar los metadatos de una actividad o recurso determinado. El sistema muestra los metadatos del recurso o la actividad. El exportador realiza los cambios necesarios y guarda los metadatos editados en el sistema, finalizando así el caso de uso.	
<b>Precondiciones:</b>	El caso de uso Exportar Curso debe haberse iniciado.	
<b>Referencias:</b>	R-2.	
<b>Prioridad:</b>	Alta.	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El exportador desea editar los metadatos de una actividad o recurso.	2. Muestra los metadatos de la actividad o el recurso seleccionado y permite editarlos: <ul style="list-style-type: none"> <li>o General.</li> <li>o Meta-metadatos.</li> <li>o Técnico.</li> <li>o Educativa.</li> <li>o Ciclo de vida.</li> <li>o Derechos.</li> </ul> Brinda la opción de continuar con el proceso de exportación conservando los cambios.	
<b>Sección General</b>		
3.a.1. Selecciona la opción General.	3.a.2 Muestra y permite editar metadatos correspondientes a la categoría General: <ul style="list-style-type: none"> <li>o Título.</li> <li>o Descripción.</li> <li>o Palabras clave.</li> </ul> Muestra, además, el Identificador correspondiente al recurso a actividad. Brinda la opción de continuar con el	

## Capítulo II: Análisis y Diseño del sistema

	proceso de exportación conservando los cambios.
<b>Sección Meta-metadatos</b>	
3.b.1. Selecciona la opción Meta-metadatos.	3.b.2. Muestra los metadatos correspondientes a la categoría Meta-metadatos: <ul style="list-style-type: none"><li>o Esquema de metadatos.</li><li>o Esquema de SCORM.</li></ul> Brinda la opción de continuar con el proceso de exportación conservando los cambios.
<b>Sección Técnico</b>	
3.c.1. Selecciona la opción Técnico.	3.c.2. Muestra los metadatos correspondientes a la categoría Técnico: <ul style="list-style-type: none"><li>o Esquema de metadatos.</li><li>o Esquema de SCORM.</li></ul> Brinda la opción de continuar con el proceso de exportación conservando los cambios.
<b>Sección Educativa</b>	
3.d.1. Selecciona la opción Educativa.	3.d.2. Muestra y permite editar los metadatos correspondientes a la categoría Educativa: <ul style="list-style-type: none"><li>o Tipo de recurso de aprendizaje.</li><li>o Nivel de interactividad.</li></ul> Brinda la opción de continuar con el proceso de exportación conservando los cambios.
<b>Sección Ciclo de vida</b>	
3.e.1. Selecciona la opción Educativa.	3.e.2. Muestra y permite editar los metadatos correspondientes a la categoría Ciclo de vida: <ul style="list-style-type: none"><li>o Versión.</li><li>o Estado.</li></ul> Brinda la opción de continuar con el

## Capítulo II: Análisis y Diseño del sistema

	proceso de exportación conservando los cambios.
<b>Sección Derechos</b>	
3.f.1. Selecciona la opción Derechos.	3.f.2. Muestra y permite editar los metadatos correspondientes a la categoría Derechos: <ul style="list-style-type: none"> <li>o Costo.</li> <li>o Derecho de copia / Otras restricciones.</li> <li>o Descripción.</li> </ul> Brinda la opción de continuar con el proceso de exportación conservando los cambios.
<b>Flujo Normal de Eventos</b>	
5. Modifica los datos deseados y selecciona la opción Continuar.	6. Notifica de la modificación del recurso o actividad mediante una metáfora intuitiva y concluye el caso de uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<b>5.a El exportador selecciona la opción Continuar.</b>	
5.a.1. El exportador selecciona la opción Continuar sin modificar los metadatos.	5.a.2. Muestra una metáfora intuitiva notificando que no se realizaron cambios en los metadatos de la actividad o al recurso y retorna a la página de Vista de Metadatos y Reglas. Finaliza el caso de uso.
<b>Poscondiciones:</b>	Se han actualizado correctamente los metadatos del recurso o actividad seleccionada.

### 2.5 Modelo de Análisis

El Modelo de Análisis es una abstracción del sistema que define una estructura para modelarlo y puede o no mantenerse durante todo el ciclo de vida del software. Por

## Capítulo II: Análisis y Diseño del sistema

---

esta razón sólo se centra en qué debe hacer el sistema: los requisitos funcionales que se presentan mediante los diagramas de clases del análisis de cada uno de los casos de uso.

Los diagramas de clases del análisis representan las relaciones entre los actores y el sistema, que se muestra seccionado en clases: interfaz, controladoras y entidades; que pueden ser la abstracción de una o varias clases o subsistemas del diseño del sistema.

**Clases interfaz:** se utilizan para modelar la interrelación entre el sistema y sus actores. Las clases interfaz modelan las partes del sistema que dependen de sus actores lo que implica que clarifiquen y reúnan los requisitos en los límites del sistema.

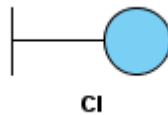


Fig. 2.3. Estereotipo de las clases interfaz.

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



Fig. 2.4. Estereotipo de las clases control.

**Clases entidad:** se utilizan para modelar información que posee una vida larga y que es a menudo persistente. En la mayoría de los casos se derivan directamente de una clase entidad del negocio. (35)

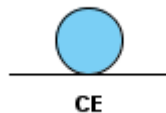


Fig. 2.5. Estereotipo de las clases entidad.

Los siguientes son ejemplos de los diagramas de clases del análisis correspondientes a los CUS.

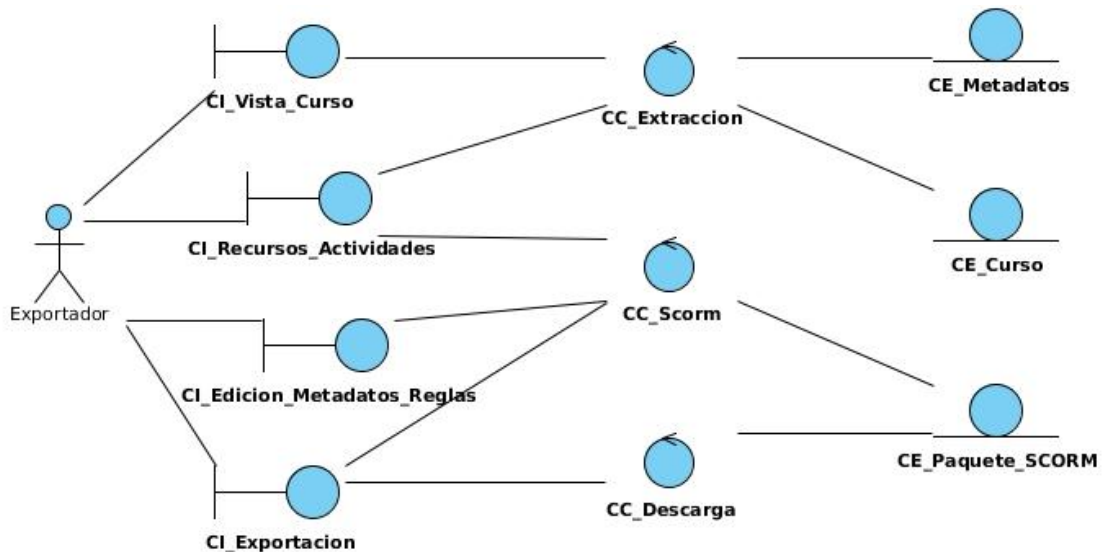


Fig. 2.6. Diagrama de clases del análisis del CUS Exportar Curso.

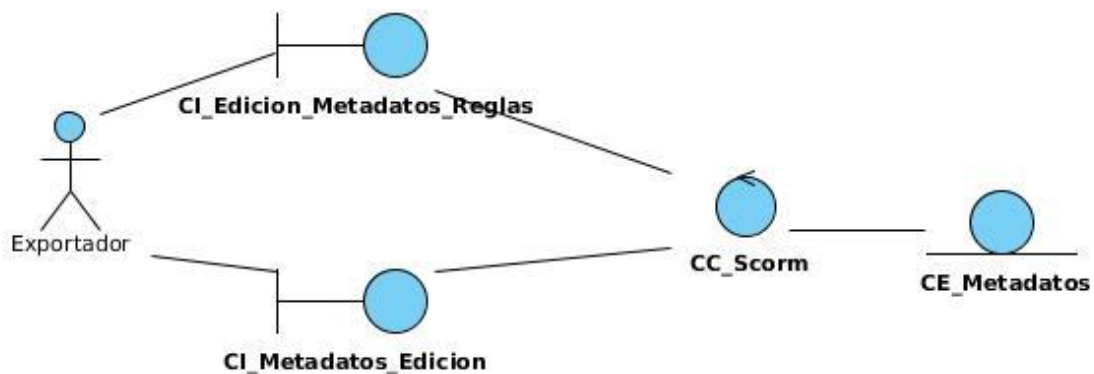


Fig. 2.7. Diagrama de clases del análisis del CUS Editar Metadatos.

### 2.5.1 Diagramas de interacción

Los diagramas de interacción representan la interacción entre los actores y el sistema para obtener determinado resultado. La metodología RUP propone su realización a través de los Diagramas de Colaboración o los Diagramas de Secuencia.

Los Diagramas de Colaboración permiten identificar los requisitos funcionales y las responsabilidades sobre los objetos. Muestran las relaciones entre objetos acompañadas de mensajes que indican el objetivo de la misma.

## Capítulo II: Análisis y Diseño del sistema

---

Por su parte los Diagramas de Secuencia, aunque con igual objetivo que el anterior, identifican la secuencia y orden cronológico de acciones. Estos diagramas contribuyen a la descripción de la dinámica del sistema en términos de la interacción entre sus objetos.

A continuación se muestran algunos de los Diagramas de Secuencia de los CUS.

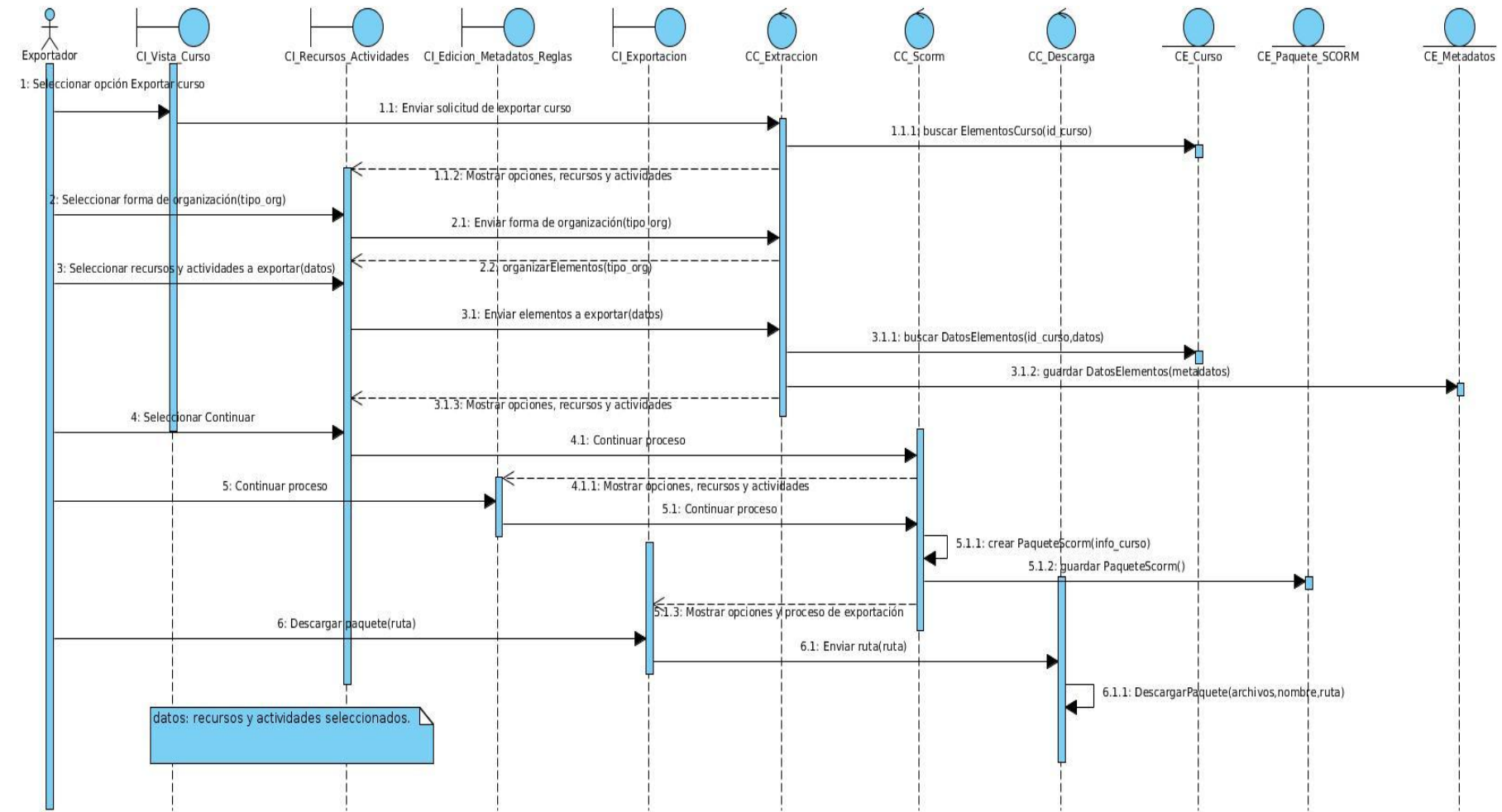


Fig. 2.12. Diagrama de Secuenciación del CUS Exportar Curso.

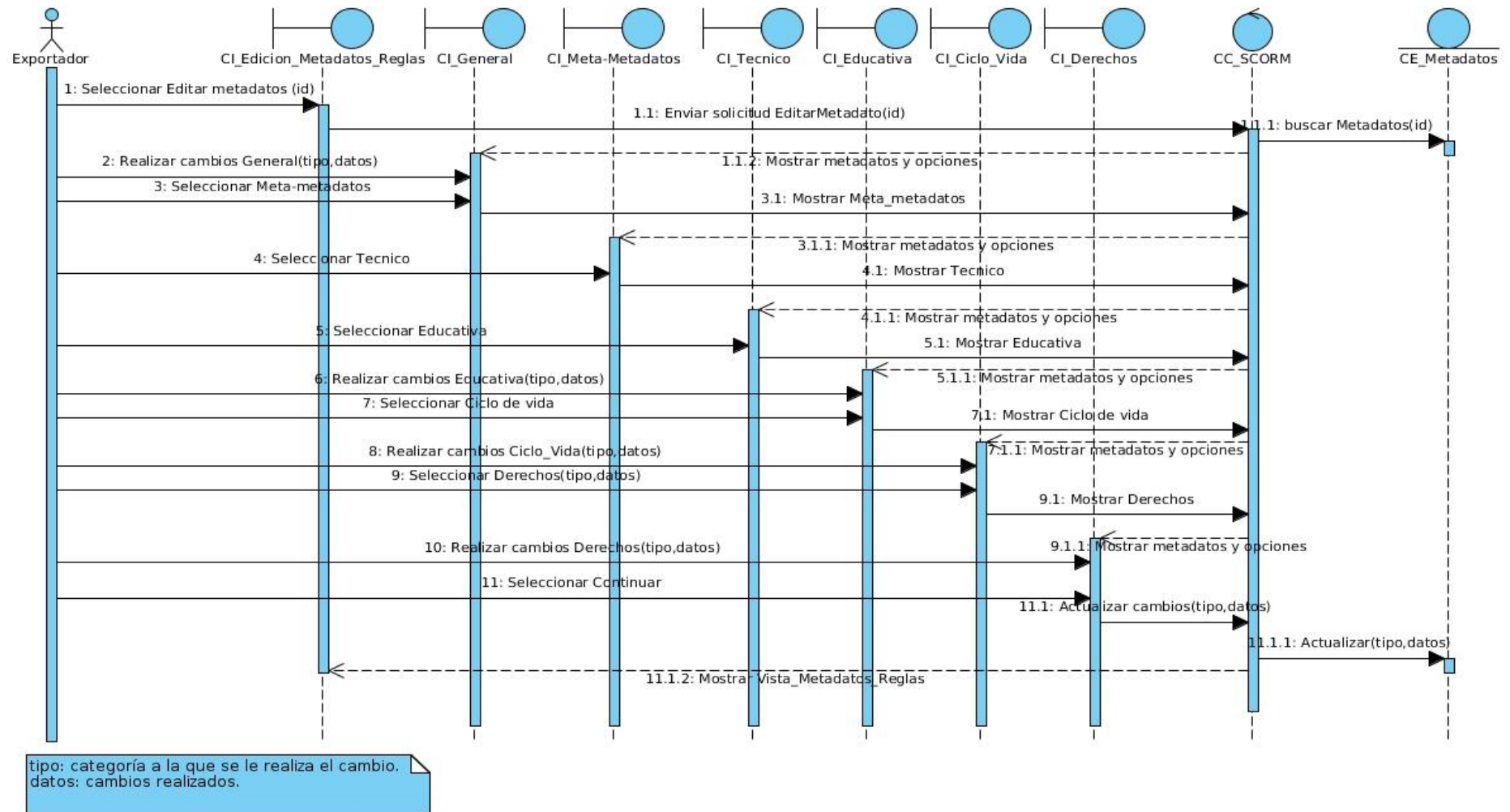


Fig. 2.13. Diagrama de Secuencia del CUS Editar Metadatos.



### **2.6 Modelo de Diseño**

Este modelo ofrece una vista del funcionamiento del sistema al que le da forma. Constituye una realización física de los casos de uso que tiene como base el resultado del análisis y contempla tanto los requisitos funcionales como no funcionales, por lo que tiene en cuenta las restricciones tecnológicas que estos establecen. Es un plano más cercano a la implementación la cual le sirve de antesala, por lo que constituye un modelo físico y concreto, y su vigencia se evidencia durante todo el ciclo de vida del software.

El Modelo de Diseño requiere de la realización de los casos de uso–diseño que es como define Jacobson: *“una colaboración en el Modelo de Diseño que describe cómo se realiza un caso de uso específico, y cómo se ejecuta, en términos de clases de diseño y sus objetos”* (36). En estos se modelan diagramas que incluyen las clases del diseño y sus relaciones.

Seguidamente se presentan los diagramas de clases del diseño con estereotipos web de cada uno de los casos de uso del sistema.



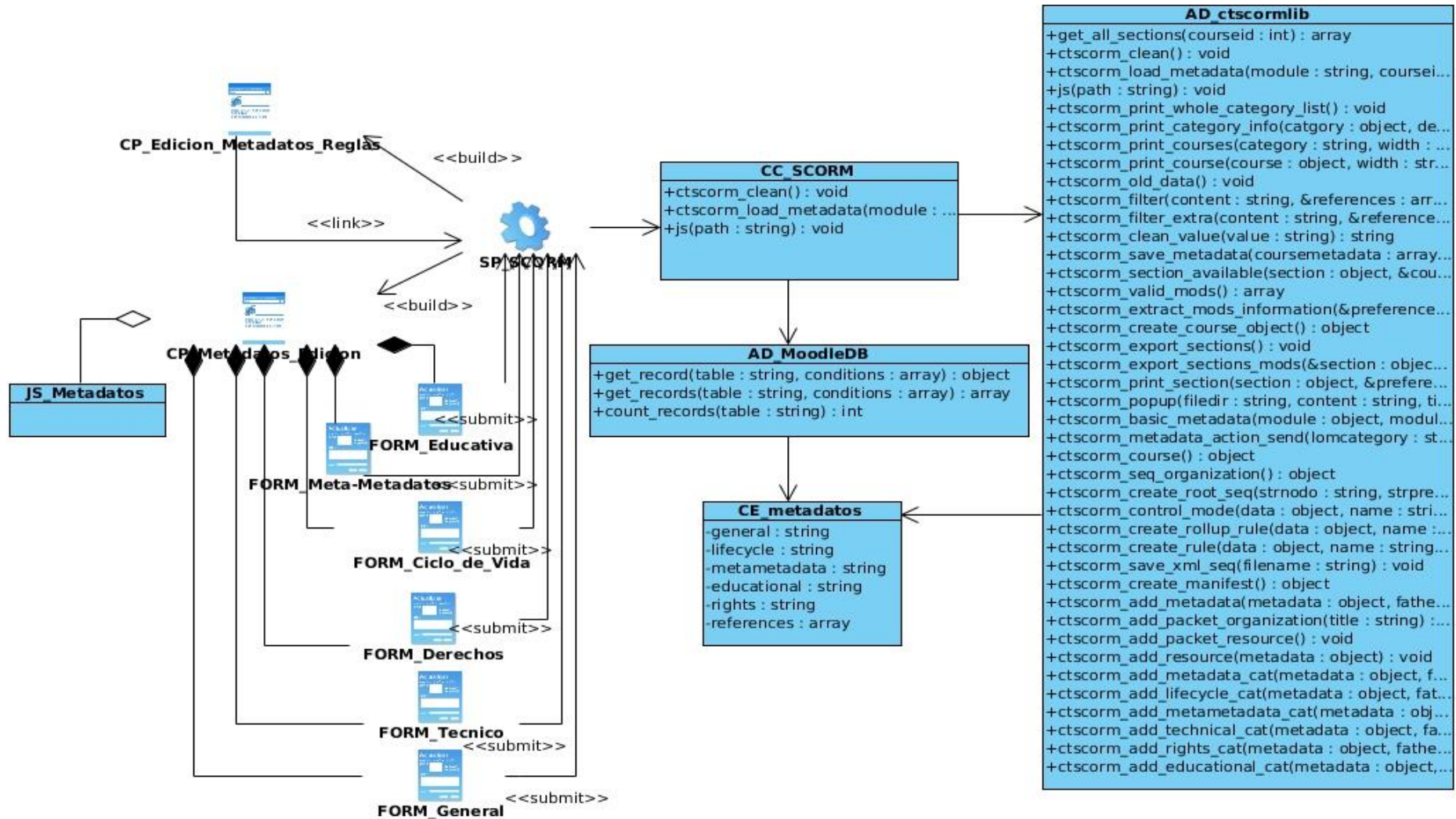


Fig. 2.19. Diagrama de clases del diseño del CUS Editar Metadatos.

### **2.7 Estructura de la base de datos**

La plataforma de teleformación Moodle como sistema modular guarda la información en diferentes lugares de la base de datos. Las versiones 2.1.x de la plataforma una vez instaladas generan una base de datos con 280 tablas y codificación UTF8. Cada tabla debe tener un campo numérico auto-incremental (id), como clave primaria.

Todo módulo tiene una tabla principal, con su mismo nombre, que contiene sus instancias y donde se encuentran: su identificador, el identificador del curso al que la instancia pertenece (*course*), el nombre completo de la instancia (*name*) y otros datos importantes para el funcionamiento de la plataforma. En su mayoría las tablas tienen un campo *timemodified*, que será actualizado con la fecha actual obtenida con la función *time ()* de PHP (20).

Los nombres de los campos deben ser sencillos y cortos, siguiendo las mismas reglas que los nombres de las variables. Los pertenecientes a las tablas relacionadas con un bloque deben estructurarse de la siguiente manera, *mdl\_block\_nombrebloque*.

Para representar la estructura de la base de datos del bloque, RUP propone la realización del Modelo de Datos, donde se muestran los elementos que la integran y las relaciones entre ellos. A continuación se presenta el modelo correspondiente al bloque C2SCORM 3.0.

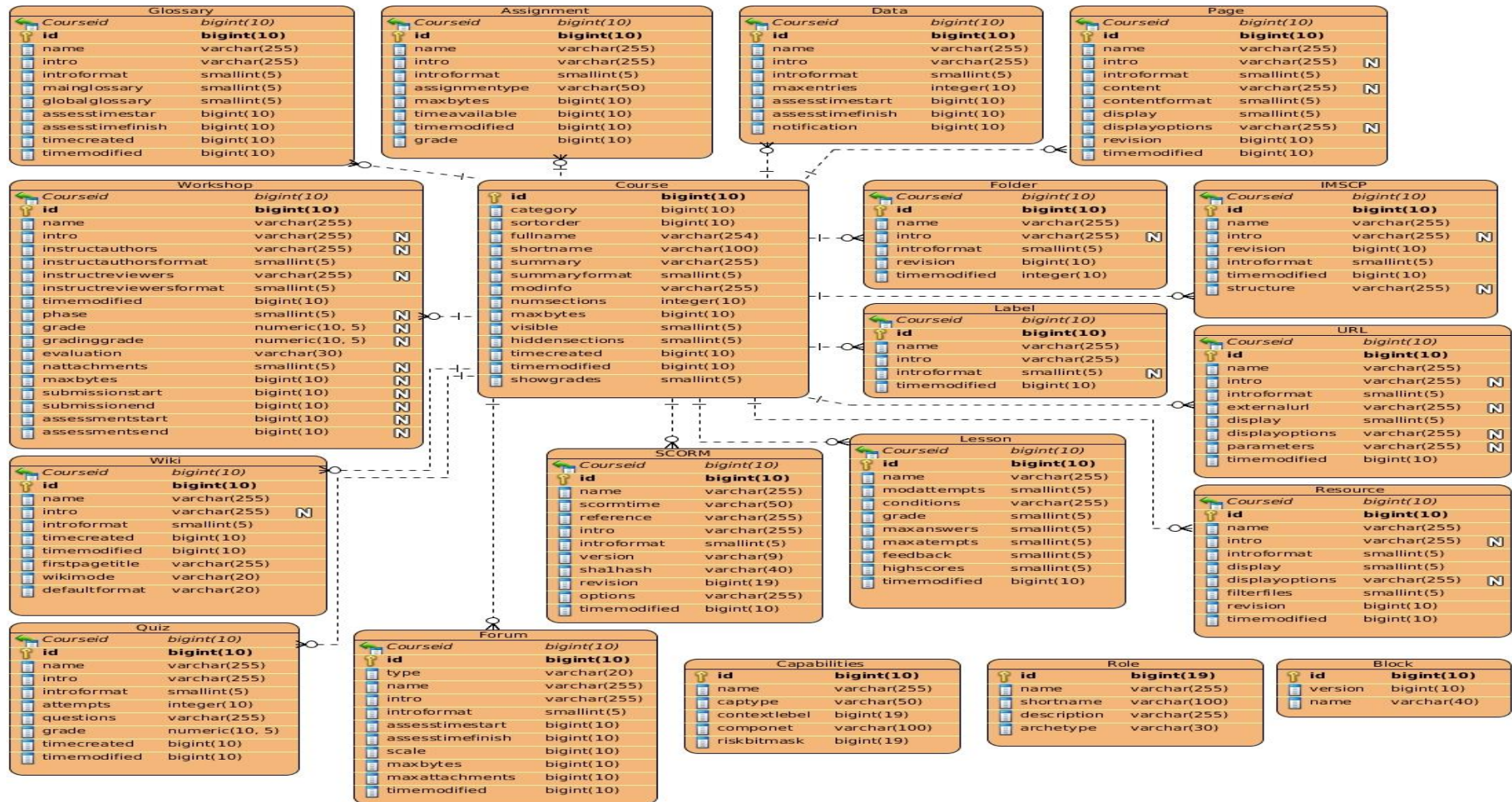


Fig. 2.24. Diagrama Entidad-Relación.

### ***2.8 Conclusiones parciales***

La definición de los principales conceptos relacionados con el dominio del problema permite la identificación de los requisitos funcionales que se pueden convertir posteriormente en funcionalidades del sistema. Los requerimientos funcionales se agrupan en casos de uso que se diagraman y describen textualmente para mejorar su comprensión por parte de los desarrolladores. Con la realización de los diagramas correspondientes al Modelo de Diseño, que contempla además de los artefactos de salida del Análisis y los requisitos no funcionales previamente identificados, se obtiene una vista del funcionamiento del sistema. Estos artefactos resultantes del Diseño constituyen la antesala de la implementación del sistema en cuestión.

---

## Capítulo III: Implementación y Pruebas

### ***Introducción***

Partiendo de los artefactos resultantes del diseño del bloque C2SCORM 3.0, en el presente capítulo se describirán los aspectos relacionados con su implementación en términos de componentes como es la distribución de su sistema de nodos, sus dependencias y la organización de sus componentes. Una vez implementada la solución será sometida a una serie de pruebas para verificar que satisfaga las necesidades de los usuarios y cumpla con los estándares de calidad.

### ***3.1 Modelo de Implementación***

El modelo de Implementación describe, en términos de componentes y subsistemas de implementación, los elementos del diseño, así como la organización de los componentes de acuerdo a los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados y la dependencia entre los componentes. (36)

El Diagrama de Despliegue ayuda a comprender la arquitectura de un sistema y según Miguel Humberto Cossio Taboada y otros en Diagrama de Despliegue, es: *“la forma de mostrar la configuración de nodos de procesamientos en tiempo de ejecución y los componentes que en ellos residen. Estos nodos forman la topología de hardware sobre el que se ejecuta el sistema. Este diagrama se preocupa principalmente de la distribución, entrega e instalación de las partes que constituyen el sistema físico”*. (37)

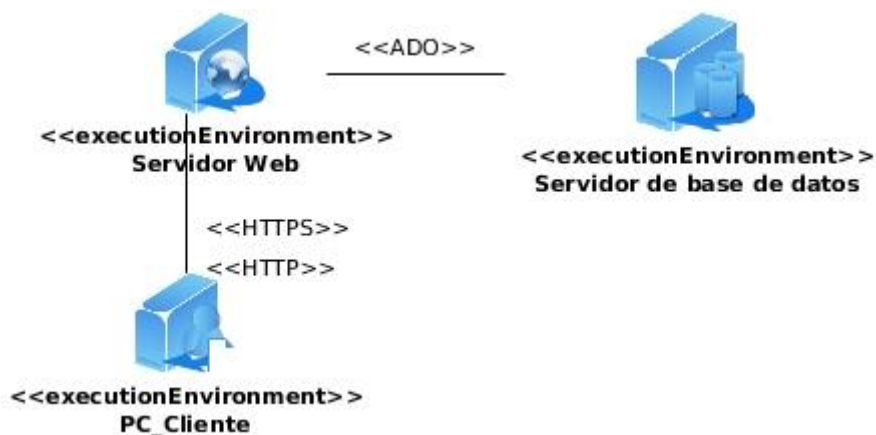


Fig. 3.1. Diagrama de Despliegue de C2SCORM 3.0.

El bloque C2SCORM 3.0 se integra a la plataforma Moodle, razón por la cual su Diagrama de Despliegue está compuesto por los nodos necesarios para conectarse a la plataforma. Para acceder al sistema los usuarios necesitan de una estación de trabajo Cliente que se conecte, mediante los protocolos HTTP<sup>19</sup> o HTTPS<sup>20</sup>, con un Servidor web donde residen las funcionalidades implementadas. Este servidor establecerá conexión con el servidor de base de datos PostgreSQL 8.4, a través del protocolo ADO<sup>21</sup>.

### 3.2 Diagrama de Componentes

Los Diagramas de Componentes muestran las dependencias lógicas entre componentes de software, sean éstos componentes fuentes, binarios o ejecutables, ilustran las piezas del software y controladores embebidos. En general, describen elementos físicos de un sistema así como sus relaciones. Prevalecen en el campo de la arquitectura de software pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema. (38)

A continuación se muestra una breve descripción de los componentes del sistema en cuestión cuyo diagrama puede observarse en la Fig. 3.2.

**ctscormdownload.php:** página que crea un archivo comprimido del paquete y brinda la posibilidad de descargarlo.

<sup>19</sup> Protocolo de transferencia de hipertexto.

<sup>20</sup> Protocolo seguro de transferencia de hipertexto.

<sup>21</sup> Protocolo de acceso a datos.



## Capítulo III: Implementación y Pruebas

---

**backuplib.php:** página que contiene funciones necesarias para crear y guardar los archivos *HTML* y de metadatos del paquete.

**ctscorm\_course.php:** página que lista las categorías existentes en el sitio, así como los cursos y sub-categorías pertenecientes a estas.

**index.php:** página principal del bloque C2SCORM 3.0.

**exportformview.js:** librería con las funciones necesarias para extraer la información referente a las actividades que sean seleccionadas para el proceso de exportación.

**settings.php:** página que muestra las variables que se definieron para la configuración global del bloque.

**moodleblock.class.php:** librería que contiene las clases base que se deben redefinir al implementar un bloque para la plataforma Moodle.

**block.php:** librería que permite a los administradores editar las variables de configuración del bloque.

**exportform.view.php:** página que muestra las actividades y recursos pertenecientes a un curso en dos variantes: vista por secciones del curso y vista por módulos.

**ctscormlib.php:** librería que contiene las funciones del bloque.

**execute.view.php:** página que añade los recursos y la organización al paquete, crea el *imsmanifest.xml* y muestra el estado de cada una de estas operaciones.

**scorm.php:** página que muestra las actividades y recursos que han sido seleccionados para el proceso de exportación, con opciones para acceder a la edición de los metadatos y la gestión de las reglas de secuencia de las mismas.

**editview.php:** página para editar los metadatos pertenecientes a una actividad o recurso en específico.

**exportcheckview.php:** librería para tomar las preferencias del curso cuyas actividades o recursos han sido seleccionados para el proceso de exportación. Crea los directorios necesarios para los diferentes archivos que conformarán el paquete, inicia las restantes librerías encargadas de extraer la información de las actividades y recursos a exportar, convertirla a un formato web similar al de la plataforma y almacenarla.

**filelib.php:** contiene las funciones para el manejo de archivos.

**preview.php:** página que muestra una vista previa de las secciones, actividades y recursos que han sido seleccionados para el proceso de exportación.

**simplelib.js:** librería con funciones necesarias para el trabajo con JavaScript asíncrono.

**ajaxrequest.php:** librería para acceder a los metadatos de las actividades y recursos que han sido seleccionadas para ser exportadas.

**sequencing.php:** librería para gestionar las reglas de secuencia de un recurso o actividad.

### ***rules***

Estos archivos permiten la creación y eliminación de las reglas que determinarán la secuencia y navegación de las actividades y recursos del paquete SCORM.

**mapinfo.php:** página que muestra opciones para gestionar lo referente al Mapa de Información de las reglas de secuencia.

**exitconditions.php:** página que muestra opciones para gestionar lo referente a las Condiciones de Salida de las reglas de secuencia.

**contromode.php:** página que muestra opciones para gestionar lo referente al Modo de Control de las reglas de secuencia.

**objective.php y objectives.php:** páginas que muestran opciones para gestionar el (los) Objetivo(s) de las reglas de secuencia.

**accumulation.php:** página que muestra opciones para gestionar lo referente a la Acumulación de las reglas de secuencia.

**potconditions.php:** página que muestra opciones para gestionar lo referente a las Poscondiciones de las reglas de secuencia.

**preconditions.php:** página que muestra opciones para gestionar lo referente a las Precondiciones de las reglas de secuencia.

### *mods*

Estas son librerías para la extracción de la información de cada módulo o recurso, su transformación, a un formato web similar al de la plataforma Moodle, y su almacenamiento.

**assignment.php:** librería para el módulo Tareas.

**quiz.php:** librería para el módulo Cuestionario.

**workshop.php:** librería para el módulo Taller.

**url.php:** librería para el módulo URL.

**page.php:** librería para el módulo Página web.

**data.php:** librería para el módulo Base de datos.

**label.php:** librería para el módulo Etiqueta.

**wiki.php:** librería para el módulo Wiki.

**scorm.php:** librería para el módulo Paquete SCORM.

**forum.php:** librería para el módulo Foro.

**folder.php:** librería para el módulo Carpeta.

**imscp.php:** librería para el módulo Paquete IMS.

**lesson.php:** librería para el módulo Lección.

**glossary.php:** librería para el módulo Glosario.

**book.php:** librería para el módulo (no estándar) Libro.

**resources.php:** librería para el módulo Recurso.

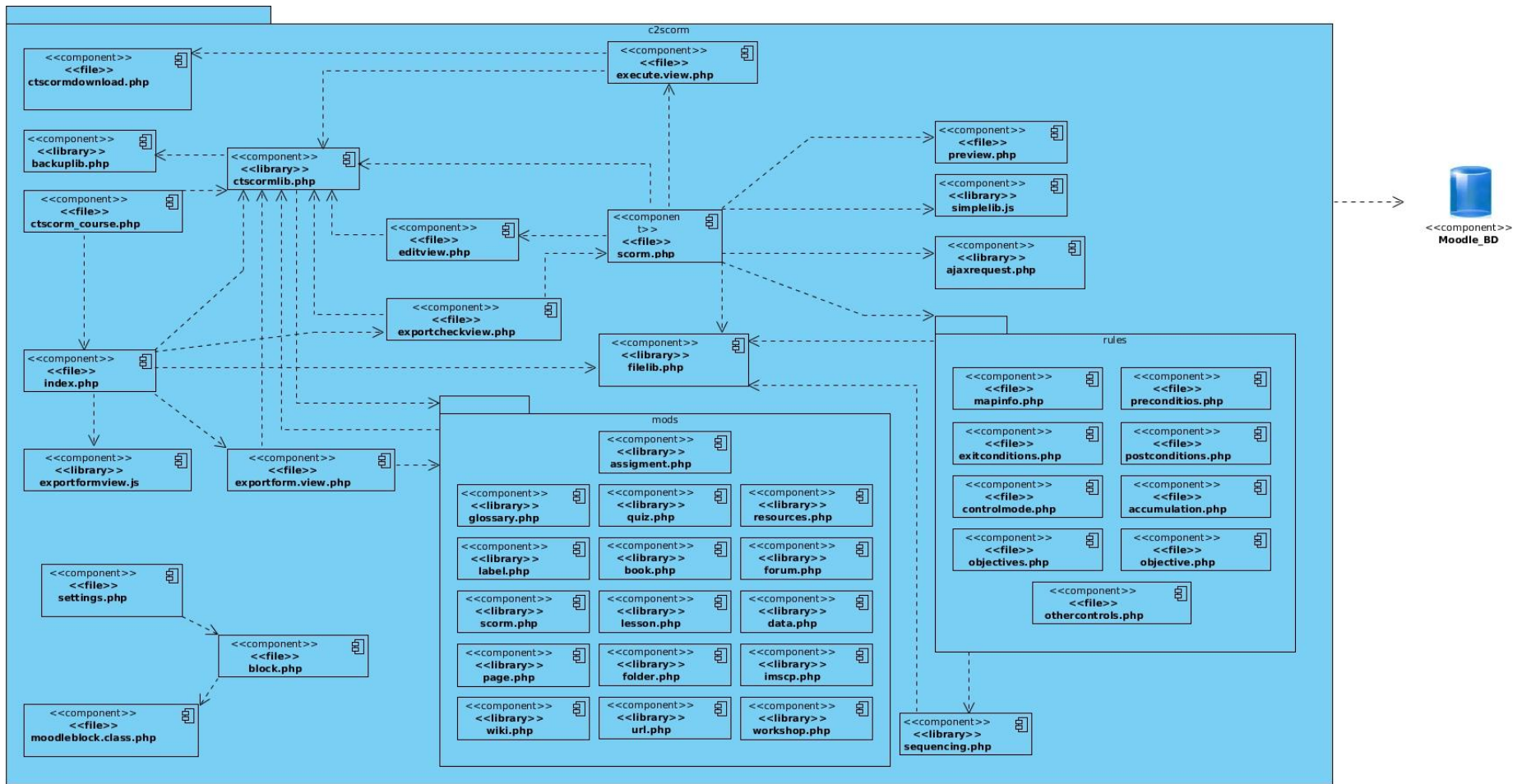


Fig. 3.2. Diagrama de Componentes de C2SCORM 3.0.

---

### 3.3 Pruebas

Luego de implementados e integrados los componentes del sistema se hace necesario verificar su funcionamiento y el cumplimiento de los requisitos necesarios para satisfacer las necesidades de los usuarios. Esta verificación se realiza a través de pruebas y permite detectar, documentar y rectificar errores en el sistema que comprometan su funcionalidad antes de ser liberado a los usuarios finales. Se verifica, además, el cumplimiento de las restricciones definidas como requisitos funcionales y no funcionales del sistema, así como su calidad. Las pruebas tienen un objetivo específico y se agrupan por niveles de acuerdo a la etapa de desarrollo en que se encuentra el software.

El modelo de pruebas describe principalmente cómo se prueban los componentes ejecutables en el Modelo de Implementación con pruebas de integración y de sistema, así como la forma en que han de ser probados los aspectos específicos del sistema. (36)

Al bloque C2SCORM 3.0 se le aplicará el nivel de prueba de sistema que se usa para verificar que los componentes funcionan correctamente como un todo. *“Es la actividad dirigida a verificar el programa final, después que todos los componentes del software y hardware han sido integrados, asegurando que el sistema realiza las debidas funciones”.* (39)

Se le realizarán al sistema pruebas de seguridad para identificar fallas en los mecanismos de protección establecidos que permitan acciones no autorizadas para los distintos roles de usuarios. Se realizarán, además, pruebas funcionales que *“tienen como objetivo asegurar el trabajo apropiado de los requisitos funcionales, entrada de datos, procesamiento y obtención de resultados. La principal intención de estas pruebas es medir la correspondencia entre la arquitectura de información propuesta y las funciones que realmente fueron implementadas.”* (39)

Este proceso exige la utilización de métodos de prueba diseñados cuyo objetivo es descubrir los fallos del sistema, por lo cual deben aplicarse a sus partes más sensibles. Uno de estos métodos es el de caja negra, con el cual se realizan pruebas, basadas principalmente en los requisitos funcionales, para comprobar la funcionalidad operativa de la interfaz del software. Este método será el empleado para verificar el cumplimiento de las especificaciones identificadas en las etapas anteriores del ciclo de vida del software.

## Capítulo III: Implementación y Pruebas

---

Para realizar las pruebas de caja negra se comprueba que la respuesta del sistema ante la entrada de datos o realización de determinada operación sea la esperada. La información tanto de entrada como de salida para cada una de estas operaciones está documentada en los casos de prueba que definen cuáles y cómo se deben probar las funcionalidades de la aplicación. Estos casos de prueba son derivados de la división del dominio del programa en clases de datos según la técnica de Partición Equivalente (40). Los del sistema propuesto podrán consultarse en el Anexo 1(Ver [Anexo 1](#)).

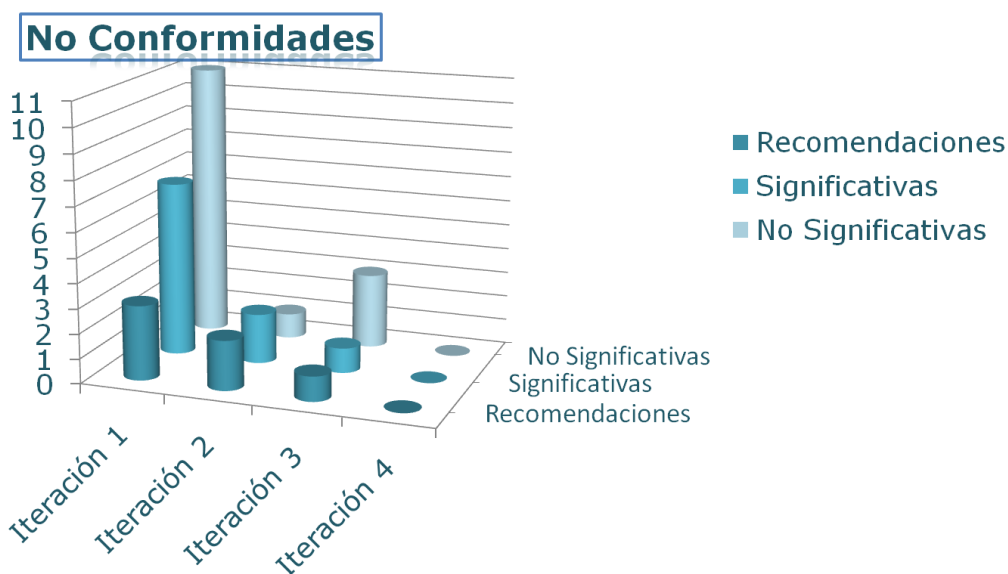
### Resultados de las pruebas

Para realizar las pruebas de seguridad se asignaron a los probadores los privilegios de Administrador y Exportador. Los usuarios intentaron acceder a funcionalidades para las cuales no contaban con los privilegios necesarios. Del proceso anterior se constató que el usuario con permisos de Exportador sólo puede llevar a cabo el proceso de exportación del un curso, en tanto con los de Administrador se puede instalar, desinstalar y configurar el bloque, así como exportar el curso.

Durante las pruebas funcionales se detectaron 31 No Conformidades, clasificadas según su importancia en Significativas, No Significativas y Recomendaciones (Ver Fig.3.3). Entre las Significativas se encontraban errores de validación de campos obligatorios, principalmente en los CUS Editar Metadatos y Gestionar Reglas, que se rectificaron para evitar complicaciones al generar y / o interpretar la información del paquete.

Otros de los fallos, No Significativos, se referían a la existencia de términos en inglés, aún cuando se trabajaba con la versión en español del bloque, los cuales se corrigieron para facilitar la comprensión por parte de los usuarios. Estos señalamientos se identificaron principalmente en el CUS Editar Metadatos.

Las Recomendaciones sugerían cambios en la redacción de las descripciones de las variables del CUS Configurar Bloque y en la presentación de los contenidos en las interfaces correspondientes a los CUS Gestionar Reglas y Configurar Bloque. Todas las recomendaciones fueron aceptadas y su incorporación benefició el diseño del sistema.



**Fig.3.3.** Gráfico de No Conformidades.

Se realizaron pruebas al paquete SCORM generado por el proceso de exportación, para las cuales se utilizaron las herramientas DeltaLearn Editor, Trident IDE y las plataformas de teleformación Moodle 2.1.2 y Zera, que permiten el trabajo con paquetes SCORM 2004.

Durante la primera iteración de las pruebas en la herramienta DeltaLearn se detectaron 18 errores y 4 advertencias que imposibilitaban la utilización del paquete en este entorno. Todos se correspondían con elementos de los metadatos que se manipulaban de forma incorrecta: (Ver [Anexo 2](#)), y que fueron corregidos posibilitando el empleo del paquete en DeltaLearn.

A diferencia de la anterior herramienta, las plataformas Moodle 2.1.2 y Zera, y la herramienta Trident IDE cargaron satisfactoriamente el paquete SCORM previamente generado por el bloque C2SCORM 3.0, permitiendo esta última su validación. Los resultados de las pruebas del paquete se pueden consultar en el Anexo 2 (Ver [Anexo 2](#)).

### ***3.4 Conclusiones parciales***

La realización de los Diagramas de Despliegue y de Componentes permitió obtener una vista física del sistema. Los componentes obtenidos como resultado de la implementación de los elementos del diseño se validaron mediante pruebas de caja negra y pruebas de seguridad. Las pruebas permitieron documentar los fallos detectados en el sistema durante su desarrollo.



---

## Conclusiones

En la presente investigación se arribó a las siguientes conclusiones:

- o La integración de la extensión C2SCORM 3.0 a la plataforma de teleformación Moodle 2.1.x permitió exportar sus contenidos siguiendo las especificaciones del estándar SCORM 2004, facilitando la reutilización de estos en otros entornos educativos.
- o Las pruebas funcionales y de seguridad realizadas al bloque C2SCORM 3.0 ratificaron el cumplimiento de los requerimientos de la aplicación previamente identificados.

---

## Recomendaciones

Se recomienda:

- La integración del bloque C2SCORM 3.0 al Entorno Virtual de Aprendizaje de la Universidad de las Ciencias Informáticas cuando este migre a una versión 2.1.x de Moodle.
- La realización de pruebas de aceptación para los nuevos usuarios.
- La ampliación del sistema partiendo de la identificación de nuevas necesidades de los usuarios de la plataforma de teleformación Moodle.
- La integración de los resultados de la presente investigación a la comunidad de Moodle.

---

## Referencias bibliográficas

1. **Grupo de Trabajo de "e-Learning" 05 de la Red TNet España.** *La formación sin distancia.* 2006.
2. *Toward an instructional design model based on learning objects.* **Chiappe Laverde, Andrés, Segovia Cifuentes, Yasbley y Rincón Rodríguez, Helda Yadira.** s.l. : Educational Technology Research and Development, 2007.
3. **Accogli, Juan Ignacio.** Ventajas del e-learning. [En línea] [Citado el: 07 de Abril de 2012.] [www.mental-gym.com/Docs/ARTICULO\\_80.pdf](http://www.mental-gym.com/Docs/ARTICULO_80.pdf).
4. **Machado, A.** Ambientes virtuales en la educación superior. [En línea] 2007. [Citado el: 17 de Enero de 2012.] <http://www.aves.edu.co>.
5. **Internet3D.** E-learning and internet 3D. [En línea] 2007. [Citado el: 17 de Enero de 2012.] <http://elearning3d.blogstop.com>.
6. **Díaz Tito, Benjamín, Hidalgo Guillén, Jesús y Gutiérrez Gómez, Yordanis.** *Herramientas para la Reutilización de Contenidos a partir de la plataforma Moodle.* Habana : s.n., 2008.
7. *Gestión de contenidos en el e-Learning.* **Lara Navarra, Pablo y Duart Montoliu, Josep Maria.** 2, 2005, Revista de Universidad y Sociedad del Conocimiento, Vol. 2, pág. 11.
8. *Hacia una taxionomía en la educación asistida por computador.* **Moreno, Julian, Ovalle, Demetrio Arturo y Vicari, Rosa M.** 9, 2010, Educación en Ingeniería, pág. 10.
9. *Sistemas de Gestión Integral de Objetos de Aprendizaje.* **Menéndez, Victor, Prieto, Manuel y Zapata, Alfredo.** 2, 2010, IEEE-RITA, Vol. 5, pág. 38.
10. **Negrín León, Julio Cesar y Rojas Fonseca, Naydeline.** *INCORPORACIÓN DEL ESTÁNDAR SCORM 2004 AL BLOQUE C2SCORM PARA LA EXPORTACIÓN DE CURSOS A PAQUETES.* La Habana : s.n., 2011.
11. **Moodle.** Moodle. [En línea] [Citado el: 22 de Enero de 2011.] <http://moodle.org>.
12. **García Alba, María Jesús.** *Análisis del desarrollo de extensiones para Moodle: Desarrollo de un módulo para la gestión de laboratorios docentes.* s.l. : Universidad de

- Alcalá, 2010.  
[http://webcache.googleusercontent.com/search?q=cache:HE\\_qZ9to9IYJ:www2.uah.es/libretics/files/GruposLab.pdf+optimizar+consultas+sql+postgres+moodle+2&cd=7&hl=es&ct=clnk&gl=cu](http://webcache.googleusercontent.com/search?q=cache:HE_qZ9to9IYJ:www2.uah.es/libretics/files/GruposLab.pdf+optimizar+consultas+sql+postgres+moodle+2&cd=7&hl=es&ct=clnk&gl=cu).
13. **Moodle.** Moodle. [En línea] [Citado el: 8 de Diciembre de 2011.] <http://moodle.org>.
14. *Interoperabilidad de los contenidos en las plataformas de e-learning*.. **Martínez Usero, José Ángel y Lara Navarra, Pablo.** 2007, Revista de Universidad y Sociedad del Conocimiento, pág. 8.
15. **Ministerio de Educación y Ciencia.** Educación. [En línea] [Citado el: 2 de febrero de 2012.] <http://ares.cnice.mec.es/informes/16/contenido/21.htm#up>.
16. **Álvarez G., Luis A., Espinoza P., Daniela P. y Bucarey A., Sandra G.** *Empaquetamiento y Visualización de Objetos de Aprendizaje SCORM en LMSs de Código Abierto.* Valdivia : s.n.
17. *Uso de estándares e-learning en espacios educativos.* **Álvarez Álvarez, José Valentín.** Sevilla : s.n., 2003, Vol. 5.
18. **Castro, M., y otros.** *ESTÁNDARES EN E-LEARNING: DE FUTURO INCIPIENTE A REALIDAD.* Madrid : s.n.
19. **Advanced Distributed Learning (ADL).** *Sharable Content Object Reference Model (SCORM®) 2004.* 2a Edición Introducción al SCORM. 2006. Disponible en [ADLNet.org \(http://www.adlnet.org/\)](http://www.adlnet.org).
20. **Advanced Distributed Learning.** *Manual de Secuencia y Navegación.* 2008. 1.2.3.
21. **PHP.** PHP: Hypertext Processor . [En línea] [Citado el: 14 de Enero de 2012.] [www.php.net](http://www.php.net).
22. **Definición.de.** Definición.de. [En línea] [Citado el: 15 de enero de 2012.] <http://definiciin.de>.
23. **Cobo, Ángel, Gómez, Patricia y Rocha, Rocio.** *PHP y MySQL Tecnologías para el desarrollo de aplicaciones web.*

24. **W3C**. W3C Interaction domain. [En línea] [Citado el: 15 de enero de 2012.] [www.w3.org](http://www.w3.org).
25. **Web, W3C world Wide**. W3C world Wide Web Consortuim. [En línea] Oficina Española. [Citado el: 15 de enero de 2012.] [www.w3c.es](http://www.w3c.es).
26. **Corporation, Oracle**. Netbeans. [En línea] [Citado el: 15 de enero de 2012.] [netbeans.org](http://netbeans.org).
27. **Florat Reyes, Daril y Campanioni Castillo, Gretell Lili**. *Secuencia y navegación en los Objetos de Aprendizaje de la Herramienta de Autor CRODA*. Ciudad de la Habana : s.n., 2011.
28. **Menéndez-Barzanallana Asensio, Rafael**. Informática Aplicada a la Gestión Pública. Facultad de Derecho UMU. [En línea] [Citado el: 16 de enero de 2012.] [www.um.es](http://www.um.es).
29. **Gómez Gallego, Juan Pablo**. *Fundamentos de la Metodología RUP*. s.l., 2007.
30. **IBM - Rational Unified Process (RUP)**. IBM - Rational Unified Process (RUP). [En línea] [Citado el: 16 de enero de 2012.] <http://www-01.ibm.com>.
31. **Amaya, Erwing**. Ingeniería de Software I. 2011.
32. **Pressman, Roger S**. *Ingeniería de Software, un enfoque práctico*. s.l. : McGraw-Hill Companies, 2002.
33. **Company Headquarters**. Visual Paradigm. [En línea] [Citado el: 16 de enero de 2012.] <http://www.visual-paradigm.com/>.
34. —. Visual Paradigm. [En línea] [Citado el: 15 de enero de 2012.] [www.visual-paradigm](http://www.visual-paradigm.com).
35. *LA INGENIERÍA DE REQUERIMIENTOS Y SU IMPORTANCIA EN EL DESARROLLO DE PROYECTOS DE SOFTWARE*. **Arias Chaves, Michael**. 10 y 11, Costa Rica : InterSedes: Revista de las Sedes Regionales, 2005, Vol. 6.
36. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James**. *El Proceso Unificado de Desarrollo de Software*. Madrid : Pearson Educación, S.A, 2000.

37. **Jacobson, Ivar , Booch , Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* Madrid : Addison-Wesley, 2000.
38. **Cossio Taboada, Miguel Humberto y Huanca Apaza, Floren Ovidio.** Diagrama de despliegue. [En línea] 2009. [Citado el: 30 de 03 de 2012.] [virtual.usalesiana.edu.bo/web/practica/archivo/ddespli2.doc](http://virtual.usalesiana.edu.bo/web/practica/archivo/ddespli2.doc).
39. **Mamani Nina, Dania, y otros.** Diagrama de componentes. [En línea] 2009. [Citado el: 30 de 03 de 2012.] [virtual.usalesiana.edu.bo/web/practica/archivo/compon.doc](http://virtual.usalesiana.edu.bo/web/practica/archivo/compon.doc).
40. **Pressman, Roger S.** *Ingeniería de software: Un enfoque práctico.* 5ta Edición. 2002. 8448132149.
41. —. *Ingeniería de software: Un enfoque práctico.* 6ta Edición. 2005. 9701054733.
42. **Lamarca Lapuente, María Jesús.** [En línea] [Citado el: 15 de enero de 2012.] [www.hipertexto.info](http://www.hipertexto.info).
43. **The Leading IT encyclopedia an learning center.** Watis?com. [En línea] [Citado el: 22 de enero de 2012.] [http://whatis.techtarget.com/definition/0,,sid9\\_gci213392,00.html](http://whatis.techtarget.com/definition/0,,sid9_gci213392,00.html).
44. ZdeS – Zona de Sistemas. [En línea] [Citado el: 10 de Mayo de 2012.] <http://zdes.wordpress.com/2009/09/17/que-es-adodb/>.
45. **Sintes Marco, Bartolomé.** mclibre. "Páginas web con PHP". [En línea] [Citado el: 10 de Mayo de 2012.] [http://www.mclibre.org/consultar/php/lecciones/php\\_db\\_pdo.html](http://www.mclibre.org/consultar/php/lecciones/php_db_pdo.html).
46. [www.masadelante.com](http://www.masadelante.com). [En línea] [Citado el: 10 de Mayo de 2012.] <http://www.masadelante.com/faqs/driver>.
47. Tecnología: Software, Aplicaciones y Servicios Web . [En línea] [Citado el: 10 de Mayo de 2012.] <http://www.desarrollodeweb.com.ar/blog/tecnologia-software-aplicaciones-y-servicios-web/331>.
48. [ilatina](http://www.ilatina.es). [En línea] [Citado el: 10 de Mayo de 2012.] <http://www.ilatina.es/urlque-url/2-12-7-12.htm> .

49. *PROPUESTA DE INTEGRACIÓN DEL SIGNARIO IBÉRICO EN EL PROYECTO UNICODE*. **Comes, Rosa y Moncunill, Noemí**. s.l. : Palaeohispanica X, 2009. 1578-5386.
50. **clic.com, Aula**. aulaClic.com. [En línea] [Citado el: 10 de Mayo de 2012.]
51. Moodle. *Moodle-Docs*. [En línea] [Citado el: 1 de Junio de 2012.] <http://docs.moodle.org>.
52. **González de Felipe, Ana Teresa**. *Guía de apoyo para el uso de Moodle 1.9.4. Usuario Desarrollador*. s.l. : EUITIO - Universidad de Oviedo, 2009.
53. Blogia. *Blogia*. [En línea] [Citado el: 1 de Junio de 2012.] <http://www.blogia.com/>.
54. **Bahamondes, Bárbara, Meri, Karina y Quintana, Johanna**. Scribd Home. [En línea] [Citado el: 4 de Junio de 2012.] <http://www.scribd.com/doc/3956635/Trabajo-colaborativo-21>.
55. **Ministerio de Educación y Ciencia de España**. Educación. [En línea] [Citado el: 23 de Enero de 2012.] [ares.cnice.mec.es/informes/16/index.html](http://ares.cnice.mec.es/informes/16/index.html).
56. *Experiencias de aprendizaje, en la organización del aprendizaje por competencias*. **Cantú Hinojosa, Irma Laura y García González, Sara Laura**. 3, s.l. : Revista Vasconcelos de Educación, 2006, Vol. II. 19-27.