

*UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS*

*FACULTAD # 10*

*“SOFTWARE LIBRE”*



**TÍTULO:**

*Herramientas para la Reutilización de Contenidos a partir de la plataforma Moodle.*

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

**AUTORES:**

Benjamín Díaz Tito

Jesús Hidalgo Guillén

Yordanis Gutiérrez Gómez

**TUTORES:**

MSc. Daymy Tamayo Avila

MSc. David Leyva Leyva

Ciudad de La Habana, Cuba.

2007 – 2008.

## DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

\_\_\_\_\_  
Benjamín Díaz Tito  
Autor

\_\_\_\_\_  
Yordanis Gutiérrez Gómez  
Autor

\_\_\_\_\_  
Jesús Hidalgo Guillén  
Autor

\_\_\_\_\_  
MSc. Daymy Tamayo Avila  
Tutor

\_\_\_\_\_  
MSc. David Leyva Leyva  
Tutor

*...“Si un hombre es perseverante, aunque sea duro de entendimiento se hará inteligente y aunque sea débil se transformara en fuerte”.*

***Leonardo Da Vinci***

## AGRADECIMIENTOS

*A la familia por brindarnos siempre apoyo y confianza.  
A Daymy y David por su ayuda, apoyo y tiempo dedicado.  
A todos los que de una forma u otra hicieron posible la realización de esta obra que se ha convertido en el fruto de años de esfuerzo y dedicación.*

**Yordanis:**

*A mis padres Irene y Orlando, por el apoyo y dedicación que me han brindado.  
A las personas a quienes acudí en busca de ayuda y que desinteresadamente brindaron un poco de su fe y apoyo.  
A mis compañeros de tesis Jesús y Benjamín, por haber estado siempre presentes y con deseos de trabajar.*

**Jesús:**

*A mis padres Jesús y Zenaida por tanto esfuerzo, firmeza y dedicación brindada, sin la ayuda de ustedes jamás lo hubiese logrado, los amo.  
A mis hermanos, en especial a Yudelsis por brindarme tantos momentos de felicidad.  
A mi niño Jesús Alejandro por soportar mi ausencia y egoísmo.  
A Aurelio (yeyo) por ser más que un hermano para mí.  
A Roxana mi compañera por haber estado a mi lado en los momentos más difíciles.  
A mis amigos por estar siempre, en especial a Alioscha, Yordanis, Benjamín, Pedro E, Raidel, Idalberto.*

**Benjamín:**

*Al Señor Jesucristo por haberme salvado y mantenido en sus caminos durante todos estos años.*

*A Ismael y Onelia, mis padres, por haberme guiado desde pequeño y por la confianza que depositaron en mí en todo momento; a mi hermano, Josué, por siempre estar a mi lado. Los quiero.*

*A Noemí, mi novia. Gracias por tu incondicional apoyo, paciencia, amor y comprensión.*

*A todos mis amigos por el aliento brindado en todo momento, en especial a Johnny, Ardenís, César, Henry, José R, Miguel.*

*A mis amigos y compañeros de trabajo durante todo este tiempo, Jesús y Yordanís.*

## DEDICATORIA

*A nuestros padres por entregarnos todo el amor y la confianza que necesitamos para alcanzar un sueño.*

## RESUMEN

Moodle constituye una herramienta con un alto grado de utilización en el mundo del e-Learning, sobre la cual está basado el Entorno Virtual de Aprendizaje de la Universidad de las Ciencias Informáticas. Ésta permite producir cursos basados en Internet y páginas web, siendo necesaria la conexión a este sistema para el uso de sus contenidos.

En este trabajo se realiza un análisis profundo del funcionamiento interno de la plataforma, priorizando la gestión de los bloques, el manejo de idiomas y el sistema de roles y permisos, proponiendo dos módulos que se integran a Moodle en forma de bloque con el objetivo de reutilizar los contenidos montados en la plataforma.

El primer módulo, C2Site, exporta los cursos como sitios web estáticos, para poder usarlos en circunstancias offline. Las versiones anteriores de este módulo se describen, explicando los factores que incidieron en su evolución. El segundo módulo, C2SCORM, exporta los contenidos en formato SCORM, dando la posibilidad de reutilizarlos en otras aplicaciones que soporten este estándar. Ambos módulos se describen utilizando Diagramas de Flujo de Datos.

### **Palabras clave:**

Reutilización, Exportación, Estándar, Moodle, SCORM.

## ÍNDICE DE FIGURAS

Figura. 2.1 <i>C2Site 1.1.1 instalado</i> .....	41
Figura. 2.2 <i>Actividades del curso a exportar</i> .....	41
Figura. 2.3. <i>Opciones desde donde generar el sitio web</i> .....	47
Figura. 2.4 <i>Clase base del bloque nombre_bloque</i> .....	49
Figura. 3.1 <i>Diagrama de Contexto</i> .....	70
Figura. 3.2 <i>Diagrama 0 o de Nivel Superior</i> .....	71
Figura. 3.3 <i>Diagrama 1 (Gestionar solicitud)</i> .....	72
Figura. 3.4 <i>Diagrama 2 (Gestionar preferencias)</i> .....	73
Figura. 3.5 <i>Diagrama 3 (Construir curso)</i> .....	73
Figura. 3.6 <i>Diagrama 3.1 (Construir módulos)</i> .....	74
Figura. 3.7 <i>Diagrama de Contexto</i> .....	85
Figura. 3.8 <i>Diagrama 0 o de Nivel Superior</i> .....	86
Figura. 3.9 <i>Diagrama 1 (Gestionar solicitud)</i> .....	87
Figura. 3.10 <i>Diagrama 2 (Gestionar Preferencias)</i> .....	87
Figura. 3.11 <i>Diagrama 3 (Generar componentes)</i> .....	88
Figura. 3.12 <i>Diagrama 3.1 (Generar módulos)</i> .....	89
Figura. 3.13 <i>Diagrama 4 (Construir paquete)</i> .....	90



## ÍNDICE DE TABLAS

Tabla 3.1: <i>Estructura de la información exportada</i> .....	66
Tabla 3.2: <i>Notación empleada en el diccionario de datos</i> .....	68

## TABLA DE CONTENIDOS

AGRADECIMIENTOS .....	I
DEDICATORIA .....	III
RESUMEN.....	IV
ÍNDICE DE FIGURAS .....	V
ÍNDICE DE TABLAS .....	VI
INTRODUCCIÓN .....	1
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA .....	6
Introducción del capítulo .....	6
1.1 ¿Qué es el e-Learning? .....	6
1.2 Sistema de administración de aprendizaje (LMS) .....	7
1.3 Sistema de administración de contenidos de aprendizaje (LCMS).....	8
1.4 Plataforma Moodle .....	9
1.4.1 Módulos de Moodle .....	10
1.5 Objetos de aprendizaje .....	13
1.6 Estándares de interoperabilidad .....	15
1.6.1 Estándar LOM.....	15
1.6.2 Modelo de Referencia para Objetos de Contenido Compartido.....	16
1.6.2.1 Modelo de agregación de contenidos (CAM) .....	18
1.6.3 ¿Por qué SCORM? .....	29
1.7 Tecnologías utilizadas.....	30
1.7.1 Tecnologías del lado del servidor .....	30
1.7.1.2 Metalenguaje de marcas extensible .....	31
1.7.1.3 Servidor Web Apache .....	32
1.7.2 Tecnologías del lado del cliente.....	33
1.7.2.1 JavaScript .....	33
1.7.2.2 Lenguaje de Marcado de Hipertexto .....	34
1.7.2.3 Hojas de estilo en cascada .....	34
1.8 Herramientas de desarrollo.....	35
1.8.1 Zend Studio .....	35
1.8.2 Dia.....	36

1.9 Metodología de desarrollo de software (Métrica v3) .....	36
Conclusiones .....	38
<b>CAPÍTULO 2: ANÁLISIS DE MOODLE Y LAS VERSIONES ANTERIORES DE C2SITE .....</b>	<b>39</b>
Introducción .....	39
2.1 Estructura general de los módulos .....	39
2.2 C2Site 1.1.1 .....	40
2.3 Estilo del código de Moodle .....	42
2.4 Aspectos fundamentales de Moodle 1.8.2 .....	43
2.4.1 Manejo de roles y capacidades .....	43
2.4.2 Manejo de idioma .....	45
2.5 Estructura de la base de datos .....	45
2.6 Copias de seguridad .....	46
2.7 C2Site 2.0.1 .....	46
2.8 Bloques de Moodle.....	48
2.8.1 Creación de bloques para Moodle.....	48
2.9 C2Site 3.0.1 .....	52
Conclusiones .....	53
<b>CAPÍTULO 3: SOLUCIÓN PROPUESTA .....</b>	<b>54</b>
Introducción .....	54
3.1 Aspectos comunes utilizados en la solución .....	54
3.1.1 Principales funciones utilizadas .....	54
3.1.1.1 Extracción de información.....	54
3.1.1.2 Verificaciones generales .....	57
3.1.1.3 Manejo de ficheros .....	58
3.1.1.4 Funciones generales .....	59
3.1.2 Configuraciones de los bloques desarrollados .....	60
3.1.3 Estructura de la información exportada.....	64
3.1.4 Diagramas de flujo de datos .....	66
3.2 C2Site .....	69
3.2.1 Diferencias principales de esta versión con las anteriores .....	69
3.2.2 Arquitectura del bloque .....	69

3.2.2.1 Gestionar solicitud del usuario.....	72
3.2.2.2 Gestionar preferencias escogidas .....	72
3.2.2.3 Construir sitio web.....	73
3.2.2.4 Panel de administración de los cursos exportados.....	74
3.2.2.5 Diccionario de Datos .....	75
3.2.2.6 Pseudocódigos más importantes.....	80
3.3 C2SCORM.....	84
3.3.1 Arquitectura del bloque .....	84
3.3.1.1 Gestionar solicitud del usuario.....	86
3.3.1.2 Gestionar Preferencias .....	87
3.3.1.3 Generar componentes .....	88
3.3.1.4 Construir paquete SCORM.....	89
3.3.1.5 Diccionario de Datos .....	90
3.3.1.6 Pseudocódigos más importantes.....	95
Representación en pseudocódigo de los principales procesos que forman parte de C2SCORM. ....	95
Conclusiones .....	101
CONCLUSIONES .....	102
RECOMENDACIONES .....	103
BIBLIOGRAFÍA .....	104
ANEXOS.....	106
Anexo 1. Implementación de estándares en las plataformas más utilizadas. ....	106
Anexo 2. Metadatos obligatorios en SCORM (ADL SCORM, 2004).....	107
Anexo 3. Multiplicidad de los elementos. ....	108
Anexo 4. Tipos de Datos. ....	109
Anexo 5. Pantalla de un sitio generado con C2Site 1.0 que contiene varios cursos. ....	112
Anexo 6. Contenidos a exportar (imagen C2Site 2.0.1). ....	113
Anexo 7. Opciones para formar el sitio web.....	113
GLOSARIO.....	114

## INTRODUCCIÓN

El uso generalizado de las Tecnologías de la Información y las Comunicaciones (TIC) conlleva cambios que alcanzan todos los ámbitos de la actividad humana, manifestándose de manera especial, y con el objetivo de flexibilizar los procesos de enseñanza-aprendizaje, en los métodos educativos.

Como resultado de la aplicación de estas tecnologías al ámbito de la educación y de la formación surge el *e-Learning* (electronic learning), definido por (Red TTnet, 2005) 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.”

Con el objetivo de apoyar esta modalidad surgen diferentes herramientas que han evolucionado hasta los actuales *Learning Management Systems (LMS)*. Estos sistemas permiten la creación y administración de contenidos, además de controlar y hacer seguimiento del proceso de enseñanza-aprendizaje.

Por otro lado existen sistemas que combinan la capacidad de los LMS para el control del proceso de aprendizaje con las capacidades de creación y almacenamiento de contenidos de los *Sistemas de Administración de Contenidos (CMS)*, llamados *Learning Content Management Systems (LCMS)*.

Uno de las grandes contrariedades de estos sistemas es que manejan grandes niveles de información aglutinada en cursos mientras que son prácticamente incapaces de enfocarse hacia unidades atómicas como los objetos de aprendizaje (OA).

No hay una definición exacta de Objeto de Aprendizaje debido a la diversidad de criterio en cuanto a las características de los mismos. En este trabajo se manejará como “*cualquier recurso con una intención formativa, compuesto de uno o varios elementos digitales, descrito*

*con metadatos, que pueda ser utilizado y reutilizado dentro de un entorno e-Learning”* (Guzmán, 2005).

El término OA procede del paradigma de la Programación Orientada a Objetos (POO), el cual 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 utilizados en diferentes escenarios de aprendizaje, con la finalidad de facilitar su clasificación y posterior reutilización.

Los materiales enriquecidos convenientemente con *metadatos* podrán almacenarse en bibliotecas digitales de contenidos educativos, por ejemplo, los *Repositorios de Objetos de Aprendizaje (ROA)* que proveen mecanismos que soportan consultas que permitirán la gestión de aquellos materiales almacenados basándose en estándares<sup>1</sup>.

Varios grupos y organizaciones como el Institute for Electrical and Electronic Engineers Learning Technology Standards Committee (IEEE LTSC); Advanced Distributed Learning (ADL); IMS Global Learning Consortium y Aviation Industry CBT Committee (AICC), han propuesto varias especificaciones y estándares que garantizan la interoperabilidad, reusabilidad, durabilidad y accesibilidad de los objetos de aprendizaje, los más difundidos son LOM<sup>2</sup> y SCORM<sup>3</sup> (de la IEEE y ADL, respectivamente).

LOM es un modelo de datos, usualmente codificado en XML<sup>4</sup>, usado para describir un objeto de aprendizaje y SCORM es un conjunto de estándares y especificaciones técnicas que permiten a los sistemas de aprendizaje basados en web, el encontrar, importar, compartir, reutilizar y exportar contenidos de formación de manera normalizada.

---

<sup>1</sup> Protocolo o norma regularizada, de uso recomendado u obligatorio.

<sup>2</sup> Este concepto se abordará más adelante.

<sup>3</sup> Ídem.

<sup>4</sup> Este concepto se abordará más adelante.

En Cuba, la aplicación de las Tecnologías de la Información y las Comunicaciones (TIC) en los Centros de Educación está propiciando una serie de transformaciones entre las que se encuentran la creación de nuevos modelos para la formación, el uso de herramientas informáticas y telemáticas dentro de nuevas concepciones.

En consonancia, la Universidad de Ciencias Informáticas (UCI) a partir del año 2005 comenzó a utilizar el LMS MOODLE (del inglés, *Modular Object Oriented Dynamic Learning Environment*) como parte de su programa de teleformación, integrando concepciones pedagógicas y tecnológicas que generen nuevas modalidades y aumenten la calidad de los procesos de enseñanza aprendizaje (PEA) en la universalización de la enseñanza.

Aunque esta plataforma brinda grandes beneficios para la teleformación, carece de la capacidad de reutilizar sus contenidos, debido a que no permite su exportación para posterior uso en otros entornos. Esto, unido a la inestabilidad en la infraestructura de las telecomunicaciones, impide poder explotar todas sus facilidades.

La UCI, no exenta de estos problemas tecnológicos, comenzó a tener dificultades con la superación del personal que debía salir de la universidad, surgiendo la necesidad de crear sitios web personalizados con la información de los cursos montados en la plataforma, repitiendo la elaboración de éstos. Este proceso constituía una pérdida de tiempo y esfuerzo, por lo que se debía buscar una alternativa que lo automatizara.

Debido a esto, en el Polo Productivo Teleformación, se implementaron diferentes versiones de un módulo para Moodle que permite exportar cursos en forma de sitio web estático. Estas versiones evolucionaron a partir del desarrollo de la plataforma y las deficiencias detectadas durante el proceso de explotación de las mismas.

Por otro lado, como se planteó anteriormente, Moodle carece de la capacidad de reutilizar sus contenidos, y con el objetivo de estandarizar esta información exportada para utilizarla en otros sistemas como ROAs y LMSs que cumplan con dichos estándares, se plantea como **problema de investigación** ¿Cómo favorecer la disponibilidad y reutilización de los

contenidos de la plataforma Moodle en la Universidad de las Ciencias Informáticas?

**Objeto de investigación:** Gestión de contenidos educativos, **Campo de Acción:** Reutilización de contenidos educativos de Moodle.

**Objetivo:** Desarrollar módulos que permitan la exportación de cursos o partes de estos, para poder ser distribuidos de forma independiente a la plataforma y reutilizados utilizando los estándares correspondientes.

**Idea a defender:** La reutilización de los contenidos educativos a partir de Moodle se puede alcanzar mediante una herramienta que permita exportarlos.

Para dar cumplimiento a estas necesidades se plantean las siguientes **tareas**:

- Analizar conceptos, tecnologías y tendencias más utilizadas en el campo del e-Learning.
- Analizar la arquitectura de Moodle.
- Analizar la base de datos de Moodle.
- Analizar los módulos de actividades, bloques y el manejo de roles de Moodle.
- Analizar los componentes de la especificación SCORM.
- Analizar el estándar LOM.
- Modelar la solución propuesta utilizando diagramas de flujo de datos.
- Implementar un bloque de Moodle para exportar cursos en forma de sitio web estático.
- Implementar un bloque de Moodle para permitir la reutilización de sus contenidos en forma de paquete SCORM.

Para el cumplimiento de estas tareas se utilizarán los **métodos teóricos** de *Análisis - Síntesis* para procesar la información y elaborar conclusiones, el *Histórico-Lógico* para el estudio del marco teórico de la investigación.

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



En el Capítulo 1, **Fundamentación Teórica**, se abordan de forma general los aspectos teóricos más importantes relacionados con el e-Learning, así como los diferentes estándares utilizados, como es el caso de SCORM, LOM y las herramientas de programación actuales necesarias para dar cumplimiento a los objetivos trazados.

En el Capítulo 2, **Análisis de Moodle y las versiones anteriores de C2Site**, se caracterizan las versiones anteriores de la herramienta web que precede este trabajo y las características principales de la plataforma que soporta la solución actual.

En el Capítulo 3, **Solución propuesta**, se realiza un análisis detallado de la solución propuesta, apoyado en la notación de los DFD como herramienta fundamental.

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

### Introducción del capítulo

En este capítulo se describen las principales características del e-Learning, los conceptos fundamentales relacionados con el PEA basado en esta modalidad educativa, así como las principales aplicaciones, tecnologías y estándares que han propiciado su desarrollo. Además, se analizan las herramientas utilizadas en el desarrollo de este trabajo y la metodología usada para modelar la solución.

### 1.1 ¿Qué es el e-Learning?

Las TIC han obtenido un gran avance en los últimos años, proveyendo mecanismos más flexibles y modernos para los procesos de enseñanza, donde aparece como actor principal el e-Learning, que como se definió anteriormente, “es un conjunto de tecnologías, aplicaciones y servicios orientados a facilitar la enseñanza y el aprendizaje a través de Internet/Intranet, que facilitan el acceso a la información y la comunicación con otros participantes”. (Red TTnet, 2005)

El e-Learning constituye una modalidad de enseñanza que presenta numerosas ventajas:

- Facilita la creación, adopción y distribución de contenidos, así como la adaptación al ritmo de aprendizaje y la disponibilidad de las herramientas de aprendizaje independientemente de límites horarios o geográficos. (Machado, 2007)
- Cuenta con utilidades para la presentación de contenidos como textos, animaciones, gráficos, videos, y herramientas de comunicación entre alumnos y tutores como foros, chats, correo electrónico, blogs, wikis, etc., contribuyendo a una mayor posibilidad de elección. (Internet 3D, 2007)
- Permite una mayor efectividad en los costos debido a la reducción de los gastos en viajes y la flexibilidad de incluir estudiantes al curso cuando esto sea necesario, sin ningún gasto asociado.
- Los accesos a información más actual y expertos remotos facilitan un proceso más rápido de formación, eficaz, y de mayor calidad.

- El alumno es el centro de una formación independiente y flexible, al tener que gestionar su propio aprendizaje, generalmente con ayuda de tutores externos. (Angel, 2000)

Entre las principales herramientas para la gestión de actividades en e-Learning se encuentran entornos de software como los LMS y LCMS, que son imprescindibles para dar soporte al proceso de enseñanza-aprendizaje.

### **1.2 Sistema de administración de aprendizaje (LMS)**

Una solución e-Learning se puede conformar principalmente por un entorno de software diseñado para automatizar y gestionar el desarrollo de actividades formativas, también llamado Sistema de Administración de Aprendizaje o LMS, ya definido anteriormente.

Un LMS, entre otras funciones, gestiona usuarios y recursos, administra el acceso, organiza catálogos de cursos, almacena datos de los usuarios, gestiona servicios de comunicación y provee informes para la gestión.

Esta aplicación reside en un servidor web, donde alumnos, tutores, profesores o coordinadores, se conectan a través de un navegador web y pueden acceder a los contenidos existentes, ver el programa de asignaturas, debatir en un foro, participar en una tutoría, configurar los cursos, habilitar servicios, etc. teniendo en cuenta sus respectivos roles o accesos.

Un LMS contiene además herramientas de comunicación, servicios y áreas para los diferentes elementos de cada curso, alumno o materia, gestión académica y administrativa para un control de acciones de los usuarios, necesario para el análisis posterior de los resultados, y un sistema que permita gestionar evaluaciones cuantitativas y cualitativas tanto de conocimiento como de asistencia.

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

engloba todas estas funcionalidades, facilitando la creación de los contenidos educativos y la gestión de éstos.

### **1.3 Sistema de administración de contenidos de aprendizaje (LCMS)**

Los sistemas de administración de contenidos educativos o LCMS, a diferencia de los LMS, están enfocados a la creación y administración de contenidos, a diferentes niveles, permitiendo de esta manera reestructurar la información y los objetivos de los contenidos de manera dinámica, para crear y modificar los OA que atiendan a necesidades y estilos de aprendizaje específicos. Estos contenidos, una vez dentro del sistema pueden ser combinados, asignados a distintos cursos, descargados, etc.

Además, un LCMS permite el diseño de contenidos interactivos sin necesidad de conocimientos de programación o diseño web, es una herramienta especializada en contenidos de e-Learning que gestiona recursos educativos, autoevaluaciones, archivos multimedia, itinerarios formativos, etc.

Un LCMS brinda colecciones de recursos digitales que contienen, a manera de base de datos, contenidos digitales y objetos de información y aprendizaje que conforman las lecciones, unidades didácticas y cursos generados. Estas colecciones son llamadas *repositorios de objetos de aprendizaje (ROA)*, que están dispuestos de tal manera que permiten consultar y reutilizar distintos contenidos, sin dañar la integridad de la información.

Otras facilidades brindadas por un LCMS son:

- Herramientas de auditoría y publicación, para crear OA que puedan ser revisados por otros usuarios, considerando los estándares *XML* y *SCORM*.
- Herramientas de colaboración que promueven la educación compartida; una interfaz dinámica para entregar la información, evaluaciones, actividades, etc.
- Una aplicación administrativa que permite llevar un seguimiento del desempeño de los usuarios, almacenar sus perfiles y características personales.

El e-Learning está basado principalmente en los LMS y LCMS. Existen varios ejemplos de plataformas con este fin, dentro de las que se encuentra Moodle, constituyendo un LMS con mucho auge en la actualidad y la base para el desarrollo de este trabajo.

#### **1.4 Plataforma Moodle**

Entorno de Aprendizaje Dinámico Orientado a Objetos y Modular (*Moodle*, del inglés Modular Object-Oriented Dynamic Learning Environment) es un paquete de software para la creación de cursos y sitios web, que permite el diseño de cursos estructurados incluyendo *recursos* y *actividades*, y llevar un seguimiento detallado de las acciones del educando y sus avances, cuestiones necesarias para dirigir un buen proceso de aprendizaje en línea.

Moodle favorece una alta disponibilidad para satisfacer las necesidades de miles de estudiantes, administradores, creadores de contenidos y profesores simultáneamente, permitiéndoles permanentemente iniciar sesión y ejecutar sus tareas diarias.

Éste sistema se distribuye gratuitamente como software libre bajo la licencia pública GNU, puede funcionar en cualquier computador en el que pueda correr PHP, soporta varios tipos de bases de datos, esencialmente MySQL y PostgreSQL, y asegura un gran rendimiento, combinando una buena elección del hardware, el sistema operativo y el sistema de bases de datos.

En Moodle existe una gran variedad de módulos para la creación de cursos, incluyendo recursos y actividades, que facilitan un proceso de aprendizaje en línea. Aquí se pueden encontrar lecciones, tareas, cuestionarios, encuestas, libros, glosarios, wikis, foros, chats, enlaces, etiquetas, etc. con estructuras diferentes según sus objetivos específicos. Además se pueden importar y utilizar OA de acuerdo a los estándares *IMS Content Packaging*<sup>5</sup> y *SCORM*, para facilitar la utilización de contenidos de diferentes fuentes.

Es una de las características de Moodle el poder instalarle fácilmente bloques, plugins, filtros y módulos no estándares, que debido a su condición de código abierto, éstos pueden ser

---

5 Formato digital estándar para representar paquetes de contenidos.

creados o modificados según las necesidades específicas de los usuarios vinculados directamente a la plataforma.

Aunque este sistema ofrece grandes beneficios, como los anteriormente mencionados, se hace necesario tener una conexión permanente que posibilite el acceso a esta plataforma para satisfacer las necesidades de los participantes en cualquier curso disponible, constituyendo así una limitación para quienes por diferentes razones no poseen acceso a la red.

### **1.4.1 Módulos de Moodle**

Los módulos son componentes auto-controlados que extienden las funcionalidades de una aplicación. Éstos deben ser fáciles de instalar y mantener, asegurando aislar el impacto que pueda tener un fallo sobre el resto del programa. La lógica del funcionamiento de Moodle se encierra en tres grandes grupos de módulos: los de comunicación, de recursos y de actividades.

*Módulos de comunicación:* Estos módulos permiten que los alumnos puedan comunicarse con el profesor (hacer preguntas, plantear dudas, etc.) y entre ellos construir su propia comunidad de aprendizaje.

*Módulos de recursos:* Estos módulos representan los contenidos y materiales del curso. Son todo tipo de textos, libros, apuntes, presentaciones de diapositivas, enlaces a páginas web externas etc., pensados para que los estudiantes los lean y estudien sobre ellos.

*Módulos de actividades:* Estos módulos son la parte activa y colaborativa donde el estudiante tiene que hacer algo más que leer un texto. Debates y discusiones, resolución de problemas propuestos, redacción de trabajos, talleres, cuestionarios en línea, etc.

Existe una gran variedad de módulos para Moodle que constituyen el eje principal de este sistema de gestión de aprendizaje. Estos módulos dan la posibilidad al usuario de enriquecer el diseño de cursos, permitiendo calificar de forma automática o manual, trabajar de forma

colaborativa, además de brindar la posibilidad de debates y consultas. Por defecto al terminar la instalación de la aplicación existen diecisiete (17) módulos estándares, y otros, no estándares, pueden ser añadidos al sistema aprovechando la flexibilidad de su arquitectura. Estos últimos pueden ser descargados desde la base de datos “Modules and Plugins” del sitio oficial de Moodle<sup>6</sup> e instalados descomprimiendo el paquete dentro del directorio */mod* en la raíz del sistema.

El objetivo de las actividades en Moodle se enfoca a la gestión del aprendizaje del estudiante, están pensadas para que el alumno trabaje y ejecute acciones, dando la posibilidad al profesor de seguir su progreso en el proceso de estudio y aprendizaje.

A continuación se explican brevemente los módulos más comunes:

### **Libro**

Este módulo facilita la elaboración de materiales sencillos de estudio compuestos por múltiples páginas, a la manera de un libro tradicional. Este módulo no es interactivo aunque existe la posibilidad de enlazar consultas, foros etc., así como de incluir objetos multimedia.

### **Tarea**

Este módulo permite asignar trabajos con objetivos individuales y se pueden especificar la fecha final de entrega y la calificación máxima. Las observaciones del profesor se adjuntan a la página de la tarea de cada estudiante y se le envía un mensaje de notificación.

### **Foro**

Son la principal herramienta de comunicación entre usuarios, ideal para publicar mensajes y entablar discusiones.

### **Chat**

Permite mantener conversaciones en tiempo real con otros usuarios.

---

<sup>6</sup> <http://moodle.org/mod/data/view.php?id=6009>

**Consulta**

Es una actividad que permite realizar encuestas rápidas y simples entre los miembros del curso.

**Hotpot**

Este módulo permite al profesorado administrar los ejercicios elaborados con “Hot Potatoes” a través de Moodle.

**Etiqueta**

Este módulo permite añadir texto HTML y pequeños gráficos en los bloques de contenido. Puede servir para identificar las partes del curso y dar una estructura lógica a la página.

**Encuesta**

Este módulo proporciona encuestas ya preparadas y contrastadas como instrumentos para el análisis de las clases en línea.

**Taller**

Es una actividad para el trabajo en grupo con un vasto número de opciones. Permite a los participantes diversas formas de evaluar los proyectos de los demás, así como proyectos-prototipo. También coordina la recopilación y distribución de esas evaluaciones de varias formas.

**Cuestionario**

Es un módulo que permite definir una base de datos de preguntas que podrán ser reutilizadas en diferentes cuestionarios. Las preguntas y las respuestas de los cuestionarios pueden ser mezcladas aleatoriamente. Son capaces de calificarse automáticamente.

**Recurso**

Este módulo admite la presentación de cualquier contenido digital, Word, Power Point, Flash, vídeo, sonido, etc. Los archivos pueden subirse y manejarse en el servidor, o ser creados en



la plataforma usando formularios web (de texto o HTML). Además es posible enlazar aplicaciones web para transferir datos.

### **Glosario**

Permite estructurar la información en conceptos y explicaciones como un diccionario o enciclopedia.

### **Wiki**

Este módulo permite a los participantes trabajar juntos en páginas web para añadir, expandir o modificar su contenido.

### **Lecciones**

Este módulo proporciona contenidos de forma flexible y normalmente termina con una pregunta y un número de respuesta posible.

### **SCORM/AICC**

Este módulo permite cargar y visualizar un paquete SCORM estándar.

## **1.5 Objetos de aprendizaje**

Como se planteó anteriormente, un objeto de aprendizaje es “un recurso con una intención formativa, compuesto de uno o varios elementos digitales, descritos con metadatos, que le posibilitan ser utilizados y reutilizados dentro de un entorno e-Learning.” (Guzmán, 2005)

Los OA brindan grandes beneficios, ya que el mismo recurso puede utilizarse en distintos contextos, es fácil su control, debido a la descripción dada por los metadatos, y son adaptables, facilitando al diseñador poder seleccionar y componer recursos según la aplicación. (Guzmán, 2005)

Con el uso de los OA se evita redescubrir soluciones, aumentando su alcance y beneficios, se pueden construir materiales de primera calidad con menos esfuerzo; éstos además, son

fáciles de acceder y compartir, incrementando así la disponibilidad de contenidos, con un soporte más amplio y distribuido.

Para cumplir el objetivo de que los OA sean unidades que faciliten el desarrollo y la expansión global del e-Learning, estos tienen atributos como: interoperabilidad, que posibilita intercambiar y mezclar contenidos de múltiples fuentes sobre distintos sistemas; reutilización, que posibilita la utilización de OA en contextos diferentes a los que fueron diseñados; accesibilidad, que posibilita al usuario acceder al contenido apropiado en el momento justo y el dispositivo correcto; y durabilidad, que es la no necesidad de una inversión significativa para lograr la reutilización o la interoperabilidad. (CNICE, 2007)

Estas características están relacionadas a la utilización de metadatos que conforman la estructura externa de los OA. Estos son un conjunto de atributos o elementos necesarios para describir un recurso, permitiendo conocer rápidamente sus principales características.

Los metadatos pueden ser: descriptivos, que principalmente tienen propósito de descubrimiento (cómo se encuentra un recurso); identificativos (cómo un recurso puede distinguirse de otro); de selección (cómo determinar que un recurso cubre una necesidad particular); administrativos, que incluyen información sobre cuándo y cómo fue creado el recurso, quién es el responsable del acceso o de la actualización del contenido y también se incluye información técnica, como la versión de software o el hardware necesario para ejecutar dicho recurso; y estructurales, que sirven para identificar cada una de las partes que componen al recurso (definen la estructura que le da forma). (Guzmán, 2005)

Estos metadatos describen las características principales de los OA, ayudando en su reutilización y clasificación, facilitando posteriormente su mantenimiento y localización en los ROAs; este proceso de reutilización está basado en el uso de estándares.

## 1.6 Estándares de interoperabilidad

Un estándar está constituido por normas, que facilitan la cooperación entre sistemas y desarrolladores de los mismos, aumentando la capacidad de elección entre diferentes contenidos compatibles que cumplan con éstas.

Existen dos tipos de estándares, los “de jure” y los “de facto”. Los estándares de jure son aquellos que han sido aprobados por un organismo oficial de estandarización, y los de facto son aquellos que se usan por voluntad propia o conveniencia y tienen una amplia aceptación, aunque no hayan sido aprobados por un organismo de estandarización. (CNICE, 2007)

Para reutilizar los contenidos educativos se hace necesario que sean diseñados de tal forma que puedan ser organizados en ROAs y utilizados en nuevos modelos de enseñanza para construir conocimiento, con este fin nacen los estándares LOM y SCORM.

### 1.6.1 Estándar LOM

*Learning Object Metadata (LOM)* es un modelo de datos que constituye un estándar oficial de metadatos para OA, especifica la semántica y la sintáctica de un conjunto mínimo de metadatos necesario para completar adecuadamente, identificar, administrar, localizar y evaluar los OA.

Debido a su extensión, los metadatos de LOM se organizan en forma jerárquica, y para poder comprender su estructura y las condiciones de llenado debe ser estudiado previamente, a fin de tener consistencia y contar con registros apegados a lo que el estándar recomienda.

LOM agrupa los metadatos en nueve categorías:

- *General*, donde se encuentra la información que describe el objeto de aprendizaje como un todo.
- *Life Cycle*, que caracteriza la historia y el estado presente del objeto de aprendizaje y de aquéllos que han afectado a este objeto durante su evolución.
- *Meta-Metadata*, que posee la información sobre los mismos metadatos, no sobre el objeto de aprendizaje que se está describiendo.

- *Technical*, que constituye los requisitos y características técnicas del objeto de aprendizaje.
- *Educational*, en la que se encuentran las condiciones del uso educativo del recurso.
- *Rights*, donde se describen las condiciones de uso para la explotación del recurso.
- *Relation*, donde se describe la relación del recurso descrito con otros OA.
- *Annotation*, donde se encuentran comentarios sobre el uso educativo del objeto de aprendizaje.
- *Classification*, para una descripción temática del recurso en algún sistema de clasificación.

Es importante destacar el hecho de que este estándar no especifica cómo implementar el esquema de los metadatos de los OA, sino que su alcance se limita a servir como referencia a otros estándares que se ocupen de esta tarea. (Surós, y otros, 2006)

Se plantea que “LOM se ha posicionado como un esquema de metadatos estable y con reconocimiento internacional, características que lo proyectan para ser implementado en aplicaciones de larga escala dentro del e-Learning”. (Guzmán, 2005)

La estructura de LOM, permite la búsqueda y recuperación del contenido de un objeto descrito por este modelo, ayudado por el estándar SCORM que permite el empaquetamiento y distribución de contenidos educativos.

### **1.6.2 Modelo de Referencia para Objetos de Contenido Compartido**

*Shareable Content Object Reference Model (SCORM)* es un estándar que presenta especificaciones para compartir, reutilizar, importar y exportar OA, permitiendo su utilización en cualquier plataforma compatible. Se concreta sobre las siguientes especificaciones:(ADL SCORM, 2004)

- IEEE Data Model For Content Object Communication, empleado en el Modelo de Agregación de Contenido para definir los Metadatos de los objetos de contenido.

- IEEE ECMAScript Application Programming Interface for Content to Runtime Services Communication, API, empleado por el Entorno de Tiempo de Ejecución para definir el mecanismo de comunicación entre el contenido y el LMS.
- IEEE Learning Object Metadata (LOM), explicado anteriormente.
- IEEE Extensible Markup Language (XML) Schema Binding for Learning Object Metadata Data Model.
- IMS Content Packaging, que provee la funcionalidad para describir y empaquetar material de aprendizaje, ya sea un curso individual o una colección de cursos, en paquetes portables e interoperables.
- IMS Simple Sequencing, que define reglas que describen el flujo de instrucciones a través del contenido según el resultado de las interacciones de un alumno con el contenido.

Un paquete SCORM está conformado por OA, descritos por metadatos contenidos en un documento estandarizado (*manifiesto*), utilizando lenguaje XML, y hojas de estilo que interpretan estos metadatos y validan estructuralmente el manifiesto. Estas características permiten compartir los OA sin perder la estructura original, haciendo uso de los ROAs.

SCORM trabaja con contenidos de formación, constituidos por unidades independientes, que representan elementos de aprendizaje que pueden estar compuestos por texto, imágenes, sonido, páginas web, video, etc., o un conjunto de todos ellos.

El estándar SCORM se centra en tres aspectos fundamentales, y para cada uno de ellos se publica un documento técnico distinto, donde se pueden encontrar: *Modelo de Agregación de Contenidos (CAM)*<sup>7</sup>, que asegura métodos coherentes en materia de almacenamiento, identificación, condicionamiento de intercambios y recuperación de contenidos; *Entorno de ejecución (RTE)*<sup>8</sup>, que describe las exigencias sobre el sistema de gestión del aprendizaje que éste debe implementar para que pueda gestionar el entorno de ejecución con el contenido

---

<sup>7</sup> Content Aggregation Model.

<sup>8</sup> Run-Time Environment.

SCORM; y *Secuencia y Navegación (SN)*<sup>9</sup>, que permite una presentación dinámica del contenido, describe cómo el sistema interpreta las reglas de secuenciamiento introducidas por un desarrollador de contenidos, así como los eventos de navegación lanzados por el estudiante o por el sistema. (Rouyet, y otros, 2006)

Para la creación de paquetes bajo el estándar SCORM, en este trabajo es de vital importancia al CAM, por lo que se profundizará en la siguiente sección.

### **1.6.2.1 Modelo de agregación de contenidos (CAM)**

El Modelo de Agregación de Contenidos (CAM), constituye un manual que detalla los mecanismos que se deben seguir para el empaquetamiento de los OA, descubrimiento en repositorios y su distribución e interoperabilidad entre distintos LMS.

Entre las principales características de CAM se pueden encontrar, primeramente la definición de los metadatos, segundo, la unión de estos al archivo XML, definiendo cómo codificarlo a fin de que sean entendibles por la máquina, y tercero, define cómo empaquetar una colección de OA con sus metadatos, y la información sobre la manera en que el contenido debe ser leído por el usuario.

Las reglas que establecen este comportamiento se dividen en cuatro partes: (ADL SCORM, 2004)

- Modelo de Contenidos, que proporciona una terminología común para ser usada en todo el CAM.
- Modelo de Empaquetado, que ofrece la descripción y los requerimientos para componer objetos de aprendizaje y formar unidades más complejas.
- Metadatos para la descripción de los propios componentes de SCORM.
- Normas de secuenciación y presentación que luego serán utilizadas en las normas del libro SN.

---

<sup>9</sup> Sequencing and navigating.

## Modelo de contenidos

SCORM recomienda seguir una versión reducida de LOM, aunque continúa con la división de nueve categorías, elige unos pocos componentes como obligatorios, para trabajar con los elementos de aprendizaje según la complejidad de estos, y los divide en los siguientes niveles: (Rouyet, y otros, 2006)

- Elemento básico (*Asset*), que representa los recursos de aprendizaje más básicos (texto, imagen, fichero MP3, página web, función *JavaScript*, etc.).
- *SCO (Sharable Content Object)*: que es un elemento básico, o un conjunto de ellos, dotado de la información necesaria para poder ser gestionados por un LMS vía SCORM RTE.
- Actividad, que es una instrucción, es decir, una acción a realizar sobre un SCO o sobre un elemento básico.
- Organización de contenido, que representa el mapa o árbol de actividades, su secuencia de ejecución.
- Agregación de contenido, que representa el conjunto completo de todos los elementos anteriores.

## Modelo de empaquetado

Con el objetivo de poner los OA a disposición de los alumnos, herramientas de autor, repositorios o plataformas de contenidos, SCORM utiliza de forma estricta la especificación IMS Content Packaging Specification, garantizando su estructura estándar. (Rebollo, 2004)

Un paquete SCORM es un archivo comprimido en formato “.zip” constituido por los ficheros físicos o sus apuntadores, y el manifiesto (*imsmanifest.xml*), que contiene la información necesaria para describir el contenido del paquete. Este paquete en SCORM se denomina *Package Interchange File (PIF)*. Se requiere que el PIF cumpla con RFC 1951, a este requerimiento SCORM añade que el formato de estos archivos sea PKZip v2.04g (.zip) (Surós, y otros, 2006). Los ficheros con este formato son los que se intercambian entre plataformas de formación. Representa una unidad de aprendizaje que tiene relevancia instruccional y puede repartirse independientemente. Queda a criterio del diseñador decidir qué es una unidad relevante.

## Componentes del manifiesto

El manifiesto está formado por cuatro secciones que se abordan a continuación: (Rebollo, 2004)

*Metadatos*: información que describe el paquete como un todo. Además describe características como el estándar que se utiliza y su versión.

*Organizaciones*: representa la organización de contenidos y su descomposición en actividades (item). Cada actividad está enlazada con los recursos que utiliza, que se encuentran en la siguiente sección, a través de su identificador (identifier). Este elemento incorpora también las instrucciones de secuenciación y navegación.

*Recursos*: describen los recursos externos y locales (en el mismo PIF) que utiliza el paquete. Un recurso está compuesto por archivos que son listados dentro de él. Todos los archivos incluidos dentro del paquete deben ser declarados y referenciados en el manifiesto. No cumplir con esta práctica puede provocar serios problemas cuando el paquete es importado, exportado o lanzado (entregado al usuario final, como por ejemplo un estudiante) por el sistema (ejemplo: plataforma de teleformación).

*SubManifiestos*: Los recursos complejos suelen estar formados por una jerarquía de entidades, cada una de las cuales tiene su propio manifiesto (cursos, lecciones, módulos) En ese caso, al construir el objeto agregado, es necesario indicar la dependencia existente entre los distintos componentes del recurso de aprendizaje.

Además se hace necesario cumplir con detalles técnicos que estandarizan el manifiesto, relevantes para el desarrollo de este trabajo, y que se describen a continuación.

### **Elemento <manifest>**

Constituye la raíz del manifiesto (*imsmanifest.xml*). Las ocurrencias de este elemento dentro de la raíz serán referenciadas como submanifiestos, aunque ADL no recomienda su uso. Las declaraciones de los espacios de nombres deben ser hechas en este elemento. Aunque esto



no es un requerimiento, según las especificaciones de XML, ADL lo recomienda como una buena práctica. (Surós, y otros, 2006)

Atributos:

- `identifier` (identificador, obligatorio): Este atributo identifica el manifiesto y es único dentro de él.
- `version` (versión, opcional): Identifica la versión del manifiesto.
- `xml:base` (opcional): Proporciona un camino relativo para la ubicación de los archivos contenidos en el manifiesto.

Hijos:

<metadata>

<organizations>

<resources>

#### Elemento <metadata>

Contiene metadatos que describen información relevante acerca del manifiesto y del paquete de contenido como un todo.

Hijos:

<schema>

<schemaversion>

#### Elemento <schema>

Describe el esquema que controla los requisitos del manifiesto. SCORM requiere que el contenido de este elemento sea ADL SCORM indicando que el paquete que representa el manifiesto ha sido construido de acuerdo con los requerimientos definidos por él.

#### Elemento <schemaversion>

Describe la versión del esquema, definido por el elemento <schema>, que se utiliza.

### Elemento <organizations>

Describe una o más estructuras u organizaciones para el paquete de contenido.

Atributos:

Default (obligatorio): Identifica la organización predefinida para usar. Su valor tiene que referenciar el atributo *identifier* de uno de los elementos <organization>, hijos del elemento <organizations>.

Hijos:

<organization>

### Elemento <organization>

Describe una organización jerárquica particular. Puede ser una lección, un módulo, un curso, un capítulo, etc.

Atributos:

Identifier (obligatorio): Un identificador para la organización que es único dentro del archivo manifiesto. Típicamente este valor es proporcionado por el autor o la herramienta de autor.

Hijos:

<title>

<item>

<metadata>

### Elemento <title>

Describe el título de la organización. Puede ayudar a elegir qué organización utilizar en dependencia de a quién vaya dirigido el contenido.

### Elemento <item>

Este nodo describe la estructura jerárquica de la organización. El elemento <item> puede anidarse dentro de otros elementos <item> sin límites en el número de niveles. Esta estructura

de los elementos <item>, da forma a la organización del contenido y describe las relaciones entre sus partes. De estos elementos, solo aquellos que sean hojas tienen que referenciar un elemento <resource>.

Atributos:

- **identifier** (obligatorio): Identifica al elemento <item> y es único dentro del manifiesto.
- **identifierref** (opcional): referencia el identificador de un recurso (*identifier*) en la sección recursos (*resources*) o en un submanifiesto.

### Elemento <resources>

Es una colección de referencias a los recursos (<resource>). No expresa orden o jerarquía de los elementos <resource> que contiene.

Atributos:

**xml:base** (optional): Proporciona un camino relativo para la ubicación de los archivos contenidos en el manifiesto. Tiene un valor SPM de 2000.

Hijos:

<resource>

### Elemento <resource>

Este elemento es una referencia a un recurso. Hay dos tipos de recursos definidos dentro de SCORM: Assets y SCOs.

Atributos:

- **Identifier** (obligatorio): Representa un identificador del recurso, único dentro del alcance de su archivo manifiesto.
- **type** (obligatorio): Indica el tipo del recurso. Si un elemento <item> referencia al elemento <resource> entonces su valor tiene que ser *webcontent*.
- **href** (optional): Es una URL que representa el “punto de entrada” o “punto de lanzamiento” del recurso. Su valor puede ser afectado por los valores que toma el

atributo `xml:base` en sus ancestros y en sí mismo. Deja de ser opcional para convertirse en obligatorio cuando un elemento `<item>` referencia al recurso `<resource>`.

- `xml:base` (opcional): Proporciona un camino relativo para la ubicación de los archivos contenidos en el manifiesto.
- `adlcp:scormType` (obligatorio): Define el tipo del recurso SCORM.

Hijos:

`<metadata>`

`<file>`

`<dependency>`

#### *Elemento <file>*

Este elemento se utiliza para listar los archivos de los cuales un recurso depende, se repite tantas veces como sea necesario por cada fichero que contiene el recurso. Cada uno de los ficheros físicos incluidos en el paquete de contenidos debe ser referenciado por un elemento `<file>`. Dejar estas referencias fuera del manifiesto puede causar una amplia gama de problemas.

Atributos:

- `href` (obligatorio): Identifica la localización del fichero.

Hijos:

`<metadata>`

#### *Elemento <dependency>*

Este elemento referencia un recurso de cuyos ficheros depende el recurso dentro del cual ha sido declarado el elemento `<dependency>`.

Atributos:

`identifierref` (obligatorio): Referencia el atributo `identifier` del elemento `<resource>` del cual depende el recurso donde el elemento `<dependency>` ha sido declarado. Se usa para obtener

la localización final de los ficheros de los cuales depende. No se permite la referencia a un recurso definido en un submanifiesto.

### **Metadatos en SCORM**

Los metadatos de LOM que propone SCORM con obligatoriedad de acuerdo a los niveles del modelo de contenidos se pueden encontrar en el Anexo 2.

Algunos elementos que conforman los metadatos usan el término SPM (el máximo permitido más pequeño) para describir la multiplicidad de estos elementos, ver Anexo 3. El SPM indica a las aplicaciones que procesan metadatos que tienen que procesar al menos ese número de elementos o caracteres, pero son libres de excederlo. (Surós, y otros, 2006)

Todos los metadatos se encuentran dentro del elemento <metadata> encontrado en el fichero imsmanifest.xml. El elemento <lom> contiene importantes elementos que SCORM requiere para describir todos los componentes del Modelo de Contenidos. Este elemento debe aparecer sólo una vez dentro de su nodo padre.

Hijos:

<general>

<lifeCycle>

<metametadata>

<technical>

<educational>

<rights>

<relation>

<annotation>

<clasification>

Los metadatos que representan interés para el desarrollo de este trabajo, se describen a continuación, teniendo en cuenta la multiplicidad y el tipo de dato (Anexo 4).

**Elemento <general>**

Hijos:

&lt;title&gt;

&lt;description&gt;

&lt;keyword&gt;

**Elemento <title>**

Representa el nombre del componente.

Multiplicidad	SPM	Tipo de Dato	SPM
1 y solo 1	-	LangString	1000

**Elemento <description>**

Representa una descripción textual del Componente del Modelo de Contenido.

Multiplicidad	SPM	Tipo de Dato	SPM
1 o más	10	LangString	2000

**Elemento <keyword>**

Se usa para definir palabras o frases que describen el objeto de aprendizaje. Si existen más de una palabra clave se deben usar varias instancias del elemento.

Multiplicidad	SPM	Tipo de Dato	SPM
1 o más	10	LangString	1000

**Elemento <lifeCycle>**

Hijo:

&lt;version&gt;

**Elemento <version>**

Describe la edición del Componente del Modelo de Contenido. Un componente puede tener varias versiones o ediciones. Este elemento permite la descripción de la versión del componente en su ciclo de vida.

Multiplicidad	SPM	Tipo de Dato	SPM
1 y solo 1	-	LangString	50

### **Elemento <metametadata>**

Hijo:

<metadatascheme>

### **Elemento <metadatascheme>**

Representa el nombre y la versión de la especificación utilizada para crear las instancias de los metadatos. Si se encuentran varios valores, entonces la instancia de los metadatos se conformará de múltiples esquemas de metadatos.

Multiplicidad	SPM	Tipo de Dato	SPM
2 o más	10	CharacterString	30

### **Elemento <technical>**

Hijos:

<format>

<location>

### **Elemento <format>**

Es usado para identificar el formato del Componente del Modelo de Contenido.

Multiplicidad	SPM	Tipo de Dato	SPM
1 o más	40	CharacterString	500

Debe ser un tipo de datos MIME representado en RFC 2048:1996.

### *Elemento <location>*

Especifica la ubicación del Componente del Modelo de Contenido descrito por los metadatos.

Multiplicidad	SPM	Tipo de Dato	SPM
0 o más	10	CharacterString	1000

### **Elemento <educational>**

Hijos:

< learningResourceType >

< interactivlevel >

### *Elemento < learningResourceType >*

Describe el tipo específico de Componente del Modelo de Contenido. Este elemento puede repetirse para describir completamente los tipos de recursos usados en el componente.

Multiplicidad	SPM	Tipo de Dato	SPM
0 o más	10	Vocabulary	-

Vocabulario que debe usarse:

excercise, simulation, questionnaire, diagram , figure, graph, index, slide, table, narrative text, exam, experiment, problem statement, self assessment, lecture

### *Elemento <interactivlevel>*

Representa el grado de dificultad del componente para el usuario que fue creado.

Multiplicidad	SPM	Tipo de Dato	SPM
0 o 1	-	Vocabulary	-



Vocabulario que debe usarse:

very easy, easy, medium, difficult, very difficult

### ***Elemento <right>***

Hijos:

<cost>

< copyright and other Restrictions>

### ***Elemento <cost>***

Multiplicidad	SPM	Tipo de Dato	SPM
0 o más	-	Vocabulary	1000

Vocabulario que debe usarse:

yes

no

### ***Elemento <copyright and other Restrictions>***

Establece si el recurso está o no sujeto a derechos de copia y otras restricciones.

Multiplicidad	SPM	Tipo de Dato	SPM
0 o más	-	Vocabulary	-

Vocabulario que debe usarse:

yes

no

### **1.6.3 ¿Por qué SCORM?**

Como puede apreciarse, son variadas las áreas a las que se dirigen los estándares. Entre ellas se destacan, y son de interés para el desarrollo del presente trabajo, la utilización de metadatos y el empaquetamiento.

La disponibilidad de metadatos proporciona recipientes para almacenar información sobre cualquier recurso de aprendizaje que es útil para otras personas que deseen utilizar este recurso e integrarlo en sus propias iniciativas de aprendizaje. En este sentido, el estándar más aceptado es LOM (de IEEE) y se incluye en las especificaciones de IMS y de ADL. (Rebollo, 2004)

Con el término empaquetamiento se hace referencia a la recolección y descripción de los elementos de un curso. Entre las iniciativas más relevantes en la actualidad se encuentra la especificación para el empaquetamiento de contenidos *Content Packaging (CP)* de IMS, que está recogida y ampliada en el documento de ADL SCORM. (Rebollo, 2004)

Por otra parte, de la amplia variedad de plataformas de formación existentes, un buen número de ellas utiliza algún estándar completo (habitualmente IMS o SCORM) o parte de ellos, (Rebollo, 2004) ver Anexo 1. Moodle, plataforma de teleformación implantada en la UCI, soporta paquetes SCORM, siendo un estándar muy difundido a nivel mundial y teniendo en cuenta su futura utilización por la universidad y sus productos, se decide tomar esta opción.

Con todos los elementos relacionados y utilizando las tecnologías necesarias, se hace posible la creación de paquetes de OA compatibles en diferentes contextos.

## **1.7 Tecnologías utilizadas**

Existe gran variedad de tecnologías usadas en el desarrollo de aplicaciones web. Seguidamente se abarcan las que constituyen la base del desarrollo de Moodle, donde se pueden encontrar, principalmente: PHP, XML, CSS y JavaScript; estableciéndose como imprescindibles para este trabajo.

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

Esta tecnología proporciona un entorno rápido de creación de scripts y soporte para los estándares más importantes. Además de las aplicaciones tradicionales de bases de datos, las aplicaciones dinámicas de Internet prometen la integración de las comunicaciones

bidireccionales y los datos en tiempo real en las aplicaciones, en este sentido, PHP se perfila como un ejemplo necesario del lado del servidor.

#### **1.7.1.1 Preprocessed Hypertext Pages**

*Preprocessed Hypertext Pages (PHP)* es un lenguaje de script embebido en HTML que proporciona características dinámicas a una página web. Se interpreta y ejecuta directamente en el servidor, con lo que el cliente únicamente recibe el resultado buscado por el código en el que está escrito.

PHP posee propósito general, su naturaleza de código abierto (open-source) lo muestra como una alternativa de fácil acceso para todos. Es sencillo para su aprendizaje, soporta en cierta medida la orientación a objetos y es capaz de conectarse con la mayoría de los manejadores de bases de datos que se utilizan en la actualidad.

PHP contiene funciones de seguridad que previenen la inserción de órdenes dentro de una solicitud de datos, posee manejo de excepciones, posibilita la generación de archivos PDF, flash, así como imágenes en diferentes formatos y no soporta directamente punteros, eliminando así los problemas de depuración provocados por estos.

PHP es multiplataforma y está siendo utilizado con éxito en millones de sitios web. Al poderse encapsular dentro de código HTML se puede recoger el trabajo del diseñador gráfico e incrustar el código PHP posteriormente. (López, 2006)

XML y PHP pueden funcionar de una forma muy similar a como puede trabajar PHP con una base de datos, y para explotar este potencial se hace uso en este trabajo de la unión de estas tecnologías.

#### **1.7.1.2 Metalenguaje de marcas extensible**

*Extensible Markup Language (XML)*, es un lenguaje o formato que permite la lectura de datos a través de diferentes aplicaciones, estructurar, almacenar e intercambiar información. Permite compartir la información de una manera segura, fiable y fácil. Se propone como un estándar

para el intercambio de información estructurada entre diferentes plataformas, y se puede usar en bases de datos, editores de texto, hojas de cálculo, etc.

XML describe el contenido de lo que etiqueta, busca precisamente crear la capacidad de hacerlo todo en la Web, llevándolo a un nivel más alto en este sentido que el HTML. Éste último no indica lo que está representando, se preocupa principalmente por cómo representarlo.

XML es extensible, lo que quiere decir que una vez diseñado un lenguaje y puesto en producción, igual es posible extenderlo con la adición de nuevas etiquetas de manera que los antiguos consumidores de la vieja versión todavía puedan entender el nuevo formato, facilitando así el trabajo para los programadores de aplicaciones web. (Wikipedia, 2007)

Para cualquier aplicación basada en tecnologías como XML y PHP, se hace necesaria la utilización de un servidor web para su desarrollo. Apache es uno de los más usados en la actualidad debido a sus ventajas y facilidades.

### **1.7.1.3 Servidor Web Apache**

Un *servidor web* es un programa que implementa el protocolo *hypertext transfer protocol (HTTP)*. Este protocolo está diseñado para transferir lo que es llamado hipertextos, páginas web o páginas HTML: textos complejos con enlaces, figuras, formularios, botones y objetos incrustados, como animaciones o reproductores de música. Se encarga de mantenerse a la espera de peticiones HTTP llevadas a cabo por un cliente, que solemos conocer como navegador.

“*Apache* se encuentra muy por encima de sus competidores, tanto gratuitos como comerciales”. (XTech, 2006) Es un software de código abierto que funciona sobre cualquier plataforma. Se distribuye prácticamente con todas las implementaciones de Linux.

Tiene capacidad para servir páginas tanto de contenido estático como dinámico, a través de otras herramientas soportadas que facilitan la actualización de los contenidos mediante bases de datos, ficheros u otras fuentes de información.

Apache se ha adaptado siempre a una gran variedad de entornos a través de su diseño modular. (Apache, 2008) Este diseño permite a los administradores de sitios Web elegir qué características van a ser incluidas en el servidor seleccionando qué módulos se van a cargar, ya sea al compilar o al ejecutar el servidor. Estos módulos son responsables de conectar con los puertos de red de la máquina, aceptar las peticiones, y generar los procesos hijos que se encargan de servirlos.

### **1.7.2 Tecnologías del lado del cliente**

Las tecnologías del lado del cliente están orientadas preferentemente, como su nombre indica, para ejecutarse en los puestos cliente, por ejemplo HTML y JavaScript, entre otros. Esto proporciona las capacidades del cliente que hacen posible la creación de aplicaciones dinámicas de Internet al aprovechar el poder de procesamiento local de las computadoras y los dispositivos.

#### **1.7.2.1 JavaScript**

JavaScript es un lenguaje que no requiere compilación, utilizado principalmente en páginas web. Permite interactuar con el navegador de manera dinámica y eficaz, proporcionando a las páginas web dinamismo y vida. No es un lenguaje orientado a objetos, debido a que no dispone de herencia, es más bien un lenguaje basado en prototipos, script u orientado a documento.

Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado.

Entre las acciones típicas que se pueden realizar en JavaScript se encuentran dos vertientes. Por un lado los efectos especiales sobre páginas web, para crear contenidos dinámicos y

elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. Por el otro, JavaScript junto al HTML, permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que es posible crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo.

### **1.7.2.2 Lenguaje de Marcado de Hipertexto**

*HTML* (del inglés HyperText Markup Language), es el lenguaje estándar para describir el contenido y apariencia de las páginas web.

Es el lenguaje más utilizado para la presentación de textos estructurados en formato hipertexto, estándar de las páginas web. HTML es utilizado prácticamente por la totalidad de navegadores web del mercado con el fin de presentar al visitante de una página web el contenido de la misma tal como el diseñador quiere que se muestre al público.

HTML utiliza etiquetas o marcas que, combinadas con un buen uso de las CSS, determinan la forma en la que debe aparecer en su navegador el texto, así como también las imágenes y los demás elementos, en la pantalla de la computadora.

### **1.7.2.3 Hojas de estilo en cascada**

CSS (del inglés Cascading Style Sheets), es una tecnología utilizada para definir la presentación de un documento HTML o XML. Constituye un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir la información presente en ese documento. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre el estilo y formato de sus documentos.

Mientras que el lenguaje HTML/XHTML se utiliza para marcar los contenidos, es decir, para designar lo que es un párrafo, lo que es un titular o lo que es una lista de elementos, el lenguaje CSS se utiliza para definir el aspecto de todos los contenidos, es decir, el color, tamaño y tipo de letra de los párrafos de texto, la separación entre titulares y párrafos, la tabulación con la que se muestran los elementos de una lista, etc. El objetivo que se

encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación, haciéndose imprescindible para crear páginas web complejas.

## **1.8 Herramientas de desarrollo**

Existe una gran variedad de herramientas que son utilizadas para el desarrollo de aplicaciones Web. A continuación se describen las herramientas utilizadas, así como algunas de las características por las que fueron seleccionadas para el desarrollo de este trabajo.

### **1.8.1 Zend Studio**

*Zend Studio* es un ambiente de desarrollo que abarca todos los componentes necesarios para desarrollar una aplicación PHP. Está implementado en Java, es multiplataforma y su interfaz de usuario proporciona rasgos que ayudan en el proceso de desarrollo.

Lo más destacable es que contiene una ayuda contextual con todas las librerías de funciones del lenguaje que asiste en todo momento ofreciendo nombres de las funciones y parámetros que deben recibir. Esta ayuda contextual no solo se queda en las funciones definidas en el lenguaje, sino que también reporta ayudas con las funciones que se vayan creando por el usuario.

Zend Studio dispone de una herramienta muy útil de debug que permite ejecutar páginas y conocer en todo momento el contenido de las variables de la aplicación y las variables del entorno como las cookies, las recibidas por formulario o en la sesión. Es posible colocar puntos de parada (breakpoints) en los scripts y realizar las acciones típicas de depuración.

La colaboración entre desarrolladores que forman un proyecto, es una de las opciones que implementa Zend Studio, al integrar el sistema de trabajo con un sistema de control de versiones (CVS)<sup>10</sup>, permitiendo mantener un registro de todo el trabajo y los cambios en los ficheros utilizados.

---

<sup>10</sup> Concurrent Versions System.

### 1.8.2 Dia

Para la creación de Diagramas de Flujo de Datos (DFD) se utiliza Dia, el cual es la alternativa libre al programa Microsoft Visio para hacer diagramas.

Dia incluye multitud de objetos para hacer diagramas de entidad relación, UML, diagramas de flujo, diagramas de red y otros. (Laquna, 2004). Es una aplicación multiplataforma bajo licencia GPL que sólo requiere la plataforma *GTK+*. Se pueden crear todo tipo de diagramas y exportarlos a casi todos los formatos existentes. Es pequeño, liviano y fácil de usar, una muestra de la calidad existente en la comunidad del software libre. (Aplicaciones Empresariales, 2007)

### 1.9 Metodología de desarrollo de software (Métrica v3)

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software. Indican paso a paso todas las actividades a realizar para lograr el producto informático deseado, señalan además qué roles estarán presentes en el desarrollo de las actividades. Detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.

No existe una metodología universal que guíe todos los procesos de desarrollo de software. Las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigen que el proceso sea configurable.

Entre las principales metodologías utilizadas en la actualidad se encuentran: *Extreme Programming (XP)*, que consiste en una programación rápida teniendo al usuario final como parte del equipo de desarrollo, y se basa en “pruebas unitarias”<sup>11</sup>, “re fabricación”<sup>12</sup> y “programación en pares”<sup>13</sup>; *Feature Driven Development (FDD)*, que se basa en funcionalidades, utilizando iteraciones cortas y propone etapas de cierre cada dos semanas; *Rational Unified Process (RUP)*, que es una metodología robusta y adaptable a la inmensa

---

<sup>11</sup> Pruebas a los principales procesos para adelantarse a los posibles errores futuros.

<sup>12</sup> Reutilización de código, para lo cual se crean patrones y modelos estándares.

<sup>13</sup> Dos desarrolladores participan en una misma estación de trabajo.



mayoría de los procesos de desarrollo, aunque es orientada a objetos; y *Métrica v3*, que contempla el desarrollo de Sistemas de Información para las distintas tecnologías que actualmente están conviviendo y cubre distintos tipos de desarrollo: estructurado y orientado a objetos.

El desarrollo del presente trabajo tiene como base Moodle, que en su versión 1.8.2+ está implementado en PHP4 y no cumple con los paradigmas de la programación orientada a objetos, además, como la solución propuesta debe estar basada en la gestión de la información de los cursos en Moodle, se selecciona Métrica v3 como metodología a utilizar.

Métrica v3 toma como referencia el modelo del ciclo de vida de desarrollo propuesto por la norma *ISO-12.207*, distinguiéndose procesos principales (planificación, desarrollo y mantenimiento) e interfaces (gestión de proyectos, aseguramiento de la calidad, seguridad y gestión de configuración), que dan un enfoque orientado al proceso, concebido para abarcar el desarrollo completo de un software, sin importar la complejidad o magnitud del mismo, para el que se adapta y dimensiona en cada momento de acuerdo a las características particulares de cada proyecto. (Ministerio de Administraciones Públicas, 2007)

La metodología está compuesta por procesos, que se descomponen en actividades y éstas a su vez en tareas. El orden de las actividades no es importante mientras estén dentro de un mismo proceso, y no se puede comenzar otro proceso sin antes haber terminado el anterior.

Los procesos de la arquitectura principal de Métrica v3 son los siguientes:

*Planificación de sistemas de información (PSI)*, para facilitar la toma de decisiones a partir de un análisis estratégico de la situación, basadas en un catálogo de requisitos del PSI y la arquitectura de información como productos finales de este proceso.

*Desarrollo de sistemas de información*, que contiene todas las actividades y tareas que se deben llevar a cabo para desarrollar un sistema, incluyendo desde el análisis de requisitos hasta la instalación del software. Este proceso se divide en cinco subprocesos:

- Estudio de viabilidad del sistema (EVS).

- Análisis del sistema de información (ASI).
- Diseño del sistema de información (DSI).
- Construcción del sistema de información (CSI).
- Implantación del sistema de información (IAS).

*Mantenimiento de sistemas de información (MSI)*, con el objetivo de obtener una nueva versión de un sistema de información desarrollado con Métrica, a partir de las peticiones de mantenimiento que los usuarios realizan con motivo de un problema detectado en el sistema o por la necesidad de una mejora del mismo.

### **Conclusiones**

En este capítulo se abordaron los conceptos fundamentales que se basan en el e-Learning y toman parte en el desarrollo de este trabajo. Además se justificó la utilización de las principales aplicaciones, tecnologías y estándares que ayudaron en este proceso de desarrollo, así como las herramientas utilizadas y la metodología que guió el avance de esta aplicación web.

## CAPÍTULO 2: ANÁLISIS DE MOODLE Y LAS VERSIONES ANTERIORES DE C2SITE

### Introducción

Este capítulo realiza una caracterización de las versiones anteriores de un trabajo que constituye la base de la solución que se propone, su evolución y las particularidades de la versión de Moodle para la que fue confeccionada cada una.

### 2.1 Estructura general de los módulos

En Moodle todos los módulos están conformados por un paquete de ficheros organizados que responden a sus necesidades específicas. Según la documentación oficial de Moodle, todo módulo posee algunos elementos obligatorios:

- *mod.html*, un formulario para establecer o actualizar una instancia de este módulo.
- *icon.gif*, un icono de 16x16 para el módulo.
- *db/mysql.sql*, volcados SQL de todas las tablas y datos requeridos de una base de datos (para cada tipo de base de datos).
- *index.php*, una página para presentar la lista de todas las instancias en un curso.
- *view.php*, una página para ver una instancia en particular.
- *lib.php*, donde se pueden encontrar todas las funciones definidas para el módulo.
- *config.html* - (opcional) un formulario para ajustar las preferencias globales del módulo.

En estos ficheros están presentes las funciones que comunican al módulo con la plataforma y las que garantizan su funcionamiento, gestionando las peticiones realizadas a las tablas que contienen la información y los recursos externos a la base de datos, para crear o mostrar las instancias de estos módulos. Además, puede encontrarse el fichero *style.php* con estilos CSS y el directorio *lang* con las especificaciones del lenguaje.

Para garantizar que el módulo creado sea incluido en la gestión de las copias de seguridad<sup>14</sup> en Moodle, deben definirse un conjunto de funciones estándar dentro de la librería *backuplib.php* que empaqueten toda la información, y de igual manera en *restorelib.php* para restaurarla.

---

<sup>14</sup> Este concepto se tratará más adelante.

La cantidad de tablas pertenecientes a un módulo en la base de datos depende de la complejidad y cantidad de información que posea una instancia.

### 2.2 C2Site 1.1.1

*Courses to Site* (C2Site) 1.0.0 es un módulo que se integra a Moodle1.5.3+ como un generador de sitios web estáticos a partir de cursos previamente montados en Moodle, con el objetivo de distribuirlos en *Disco Compacto (CD)* u otro dispositivo de almacenamiento externo.

La base de datos de Moodle1.5.3+ tiene una estructura de 132 tablas (sin la instalación de bloques, actividades o módulos no estándares), y la codificación no es obligatoria a UTF8, lo que constituye una dificultad, pues las páginas pueden aparecer con caracteres incorrectos.

C2Site 1.0.0 no tiene un instalador, se le agregan manualmente algunas líneas de código al fichero `/blocks/admin/block_admin.php` (dirección a partir de la raíz de Moodle) para crear los enlaces que hacen funcionar el paquete que conforma la aplicación. Los cursos exportados se envían a una carpeta dentro del directorio de datos del sitio, donde hay que acceder directamente para obtenerlos. Estos problemas se resolvieron en la versión 1.1.1 de C2Site, que se instala en la plataforma como una actividad en funcionamiento.

Esta última versión también funciona de manera estable para Moodle 1.5.3+, y se agrega como una opción más del panel de administración de los cursos en la plataforma (Figura 2.1), permitiendo tanto a administradores como a profesores, exportar los mismos. Una vez exportados dichos cursos en forma de sitio web, se almacenan en los *Archivos del curso*, si es un profesor; o en *Archivos de sitio*, en el caso de un administrador.

Un profesor puede exportar solamente sus cursos, mientras que un administrador tiene la posibilidad de exportar cualquiera de estos. Los administradores son los únicos autorizados para conformar un sitio web con varios cursos, debido a que en esta versión de Moodle los archivos del sitio solo son accesibles para usuarios con privilegios administrativos.

Dentro del proceso de exportación existen diferentes pantallas que guían las acciones del usuario, otra forma de comenzar la exportación es a través de un enlace en el panel de administración de Moodle, que permite listar todos los cursos existentes en la plataforma para seleccionar uno a exportar. Luego el sistema da la posibilidad de seleccionar las actividades y recursos a exportar y la categoría a la cual pertenecerá dicho curso en el sitio web generado (Figura 2.2).



Figura. 2.1 C2Site 1.1.1 instalado



Figura. 2.2 Actividades del curso a exportar

El sitio generado conserva la estructura visual del curso original en Moodle, adaptándose al diseño del sitio web estático, donde se adicionan además, mecanismos que permiten navegar entre los cursos exportados o las páginas de un curso (Anexo 5).

La construcción del sitio está basada en la forma que usa la versión 1.5.3 de Moodle para mostrar la página central de un curso. La función `get_all_sections()` obtiene todas las secciones del curso y por cada sección se invoca `print_sect()`, que se encuentra en el fichero `lib.php`, y se encarga de extraer toda la información necesaria para cada módulo instanciado y almacenarlo, devolviendo una referencia al recurso creado. Estas referencias son usadas para integrarlos en un sitio web.

Las funciones antes mencionadas y otras funciones auxiliares son ejecutadas por una página principal, *index.php*, que controla el flujo de las acciones en el transcurso de la exportación, crea los directorios y ficheros que constituyen el sitio web, copia toda la información y recursos que no se encuentran en la base de datos y que forman parte de los archivos del curso.

La extracción de la información de la base de datos se realiza de forma directa, utilizando la función *get\_record()* de Moodle, cuando se necesita extraer una instancia y *get\_records()* si son varias. Esta característica de C2Site 1.1.1 elimina las posibilidades de compatibilidad en caso de ocurrir algún cambio que afecte la estructura general de la base de datos, o a una tabla consultada dentro del proceso de exportación.

En la construcción del sitio se utiliza CSS con el objetivo de estandarizar los estilos de la información que se muestra al usuario final y se mantienen las imágenes originales de los cursos montados en la plataforma y el formato que presentaba antes de ser exportados.

La programación de C2Site 1.1.1 no cumple con el estándar de codificación definido para desarrollar sobre Moodle<sup>15</sup>. Un ejemplo significativo lo constituye “C2Site” como nombre del módulo y prefijo en la mayoría de sus funciones y que según los estándares oficiales de Moodle, no deben contener números en el nombre.

Esta solución posee una estructura poco organizada, desde el punto de vista arquitectónico como en el código fuente. El código que construye cada una de las actividades está agrupado en una misma función de casi mil líneas.

### **2.3 Estilo del código de Moodle**

Debido a que Moodle es un proyecto colaborativo con una extensa comunidad de desarrolladores que aportan a su funcionamiento, se necesita que el código insertado se rijan por estándares de codificación definidos para el sistema, posibilitando la uniformidad en la estructura y apariencia del código.

---

<sup>15</sup> Se abordará más adelante

Según la documentación oficial de Moodle, las variables y funciones deben tener nombres en minúsculas con significado en inglés y fáciles de leer. Las variables que representan arreglos o matrices deben usar nombres en plural. Si el nombre es compuesto por varias palabras, estas deben escribirse juntas. Las constantes deben ser nombradas en mayúsculas y comenzar con el nombre del módulo al que pertenecen, seguido de un guión bajo.

Para nombrar las funciones debe utilizarse como prefijo el nombre del módulo al que pertenecen. Es importante no dejar espacio alguno entre el nombre y la apertura de paréntesis. Las funciones deben estar comentadas, utilizando el formato *phpDoc*, para explicar el flujo del código, el propósito y las variables significativas.

Los bloques de código deben estar siempre entre llaves, aunque aparezca solo una sentencia y se pueden usar espacios en blanco con bastante libertad, para ganar en claridad del código, aunque debería haber un espacio entre llaves y líneas normales, y ninguno entre llaves y variables o funciones.

## **2.4 Aspectos fundamentales de Moodle 1.8.2**

La versión 1.8.2 de Moodle fue liberada el 8 de Julio de 2007. Ésta incorpora mejoras en la accesibilidad, conforme a los principales estándares internacionales. Todos los formularios utilizan una API común para definir los campos y visualizarlos de forma consistente, proporcionando una vía ordenada y segura de validar y recibir información de ellos si utilizar HTML. Además, el interfaz de Moodle cumple ahora con XHTML Estricto 1.0 y la mayoría de los estándares de accesibilidad (NosoloRed, 2007).

Otro de los aspectos que se mejoraron fue la usabilidad y flexibilidad del sistema de roles y capacidades, existente ya desde la versión 1.7 de Moodle donde los usuarios autorizados pueden crear roles adaptados a sus necesidades y asignarlos a los usuarios.

### **2.4.1 Manejo de roles y capacidades**

El sistema de roles usado por Moodle está basado en 4 pilares fundamentales para gestionar la seguridad:

- Un *rol*, que es un identificador del estatus del usuario en un contexto particular.
- Una *habilidad*, que es una descripción de una funcionalidad particular de Moodle. Las habilidades (o capacidades) están asociadas a los roles.
- Un *permiso*, que es un valor que se asigna a una capacidad para un rol en particular.
- Un *contexto*, que es un "área" en Moodle en la cual se pueden asignar roles a los usuarios.

Los contextos se organizan de forma jerárquica y sus permisos se transfieren desde los contextos 'superiores' a los 'inferiores'. Moodle 1.8.2, además de haber solucionado/refinado problemas en los roles, ha separado el contexto *SYSTEM* del contexto *SITE*, lo que hace posible una mayor flexibilidad en las configuraciones de los permisos.

El contexto *SITE* es exclusivamente la página principal de la plataforma y sus actividades. El contexto *SYSTEM* es el contexto raíz y cualquier permiso otorgado en él se aplica a todos los cursos. Esta división hace más sencillo para los administradores el proceso de ajustar los permisos correctamente. Los administradores pueden ver los permisos recomendados para los roles y pueden "resetearlos" a sus valores por defecto (NosoloRed, 2007).

La jerarquía de contexto se establece de la siguiente forma (Moodle Docs, 2008):

- Contexto de *sistema* - accesible a través del bloque de administración (nivel superior).
- Contexto de *sitio* - accesible a través del bloque de administración (padre => sistema).
- Contexto de *categoría de curso* - accesible a través de la página de categorías de cursos (padre => sitio).
- Contexto de *curso* - accesible a través del bloque de administración del curso (padre => categoría de curso o sitio).
- Contexto de *módulo* - accesible mientras se edita el módulo (padre => curso).
- Contexto de *bloque* - accesible mientras el modo de edición está activado (padre => sitio o curso).
- Contexto de *usuario* - accesible a través de la pestaña de Roles en el perfil de usuario (padre => sitio).



Moodle maneja los roles y accesos basándose principalmente en la función *has\_capability()*, ubicada en */lib/acceslib.php*, encargada de verificar los permisos del usuario en un contexto determinado.

### 2.4.2 Manejo de idioma

Moodle se ha diseñado con el propósito de ser internacional. Cada 'cadena' de texto que forma parte de la interfaz se extrae de un conjunto de archivos de idioma. En el paquete estándar de la plataforma sólo se encuentra el inglés, aunque los restantes idiomas pueden ser descargados desde la página oficial de Moodle y se instalan con solo descomprimirlo dentro del directorio */lang* en la carpeta de datos del sitio. El administrador permite al usuario decidir en qué idioma disponible la navegación por la plataforma.

Para visualizar el idioma seleccionado, el sistema ejecuta la función *get\_string()* de la librería *lib/moodlelib.php*, que recibe como parámetros un identificador de la cadena a traducir y el módulo al que pertenece, luego se encarga de buscar en el fichero correspondiente al idioma seleccionado, las cadenas del lenguaje a mostrar en pantalla.

Los ficheros que contienen las especificaciones globales de los paquetes de idioma que estén instalados en la plataforma no contemplan las características privadas de cada bloque, por lo que se hace necesaria la fabricación de la traducción a las cadenas de texto que se utilizan en cada bloque. Estos ficheros con las especificaciones del lenguaje deben estar ubicados en el directorio *lang* del bloque.

## 2.5 Estructura de la base de datos

Moodle 1.8.2 una vez instalado utiliza una base de datos con 215 tablas y codificación UTF8. Cada tabla debe tener un campo numérico autoincremental (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.

El resto de las tablas asociadas con un módulo que contienen información sobre 'componentes', deberían ser llamadas con el nombre del módulo al que pertenecen y el de los 'componentes' (`mdl_modulo_componentes`). Los nombres de los campos deben ser sencillos y cortos, siguiendo las mismas reglas que los nombres de las variables.

Las tablas asociadas con un bloque deben estructurar su nombre de la siguiente manera `mdl_block_nombrebloque_añadidos`. La mayoría de las tablas tienen que tener un campo *timemodified*, que será actualizado con la fecha actual obtenida con la función `time()` de PHP.

## 2.6 Copias de seguridad

Moodle cuenta con un sistema de copias de seguridad, que posibilitan realizar salvadas de los contenidos de un curso con el objetivo de mantener cierta capacidad de recuperación de la información ante posibles pérdidas.

En Moodle una copia de seguridad es un paquete compactado que contiene un archivo XML (*moodle.xml*) con toda la información necesaria para restaurar la asignatura desde cero. Especifica los paneles que estarán visibles en el interfaz, los bloques temáticos, el contenido (recursos y actividades) y los archivos que hayamos añadido al curso.

Para lograr que las copias de seguridad sean compatibles entre versiones, Moodle ha mantenido uniformidad en la estructura de los contenidos almacenados dentro del paquete, constituyendo ésta la forma más estándar de manipular la información de los cursos.

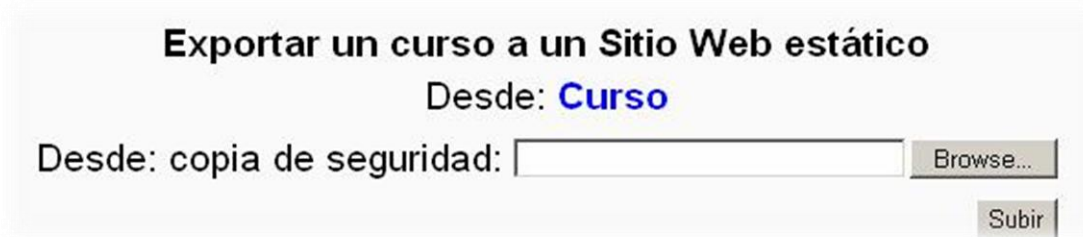
## 2.7 C2Site 2.0.1

C2Site 2.0.1 posee las mismas funcionalidades que C2Site 1.1.1 y está desarrollado con el objetivo de incorporarse a Moodle 1.8.2. Incorpora el uso del fichero *defaults.php* con el objetivo de facilitar la instalación automática del módulo. El mismo es invocado por el sistema al instalar el módulo, insertando en la base de datos la habilidad requerida para exportar cursos y automatizando la creación de los enlaces en el bloque de administración del sitio.

Otro aporte importante es la forma utilizada para extraer la información de los cursos, que está basada en las copias de seguridad de Moodle, proporcionando mayor independencia de la estructura de la base de datos y favoreciendo la compatibilidad de C2Site con otras versiones de Moodle.

Esta versión de C2Site separa la gestión de la información de cada módulo en librerías independientes, las cuales son invocadas por el sistema para procesar la información obtenida de una copia de seguridad (*backup*) que se realiza transparente al usuario. Cada una de estas librerías implementa una función *nombremodulo\_export\_mod*<sup>16</sup> que se encarga de extraer la información de cada instancia almacenada en el fichero *moodle.xml*, la cual es procesada por la función *nombremodulo\_print\_mod* para construir el recurso final.

Estas características permiten exportar un curso cargando una copia de seguridad previamente confeccionada en Moodle (Figura 2.3).



**Exportar un curso a un Sitio Web estático**  
Desde: **Curso**  
Desde: copia de seguridad:

Figura. 2.3. Opciones desde donde generar el sitio web

Esta versión no solo permite seleccionar los módulos a exportar, sino las instancias de cada uno de estos (Anexo 6), además posibilita editar el nombre de la categoría que agrupará el curso. Igualmente incorpora la opción de seleccionar el diseño a utilizar en el sitio web generado (Anexo 7).

Durante el período de funcionamiento de C2Site 2.0.1 se descubrieron imperfecciones en el proceso de elaboración de las copias de seguridad en Moodle 1.8.2. Este problema afecta directamente el correcto funcionamiento de esta versión, haciendo necesario buscar una

<sup>16</sup> “nombremodulo” se refiere al nombre del módulo en Moodle.

solución independiente al sistema de backup en Moodle, sin estar ligado directamente a la estructura de la base de datos.

Otra dificultad de esta solución es que se integra a Moodle como una actividad cuando no contribuye directamente al desarrollo del aprendizaje de los estudiantes. Por estas razones, se comienza a investigar vías alternativas para integrar nuevas funcionalidades a la plataforma.

## 2.8 Bloques de Moodle

Moodle ofrece un conjunto de bloques que realizan diversas tareas y gestionan diferentes tipos de información. Estos encierran un conjunto de funcionalidades que robustecen el sistema. Algunos se incluyen con el paquete estándar de Moodle, pero otros no estándar, pueden ser agregados por el administrador. En el sistema, los bloques se encuentran en ambos lados de la pantalla, dejando el centro para visualizar los contenidos del sitio. Siempre que tenga permisos de edición podrá ocultarlos, suprimirlos y desplazarlos lateral o verticalmente.

El objetivo principal de estos bloques no es la creación de contenidos educativos, ni proveer un medio de chequeo y control del desarrollo del aprendizaje, más bien están destinados a fortalecer las capacidades del sistema, originando nuevos eventos dentro de la plataforma.

Las ventajas que ofrecen las características de los bloques de Moodle, constituyen la base sobre la que se desarrollan las herramientas posteriores.

### 2.8.1 Creación de bloques para Moodle

Para desarrollar un bloque en Moodle, se debe crear un directorio en */blocks/nombre\_bloque* que incorpore un fichero llamado *block\_nombre\_bloque.php* en el cual se define la clase *block\_nombre\_bloque* que hereda de *block\_base* o *block\_list*, ambas pertenecientes a la librería */blocks/moodleblock.class.php*.

La clase creada debe redefinir básicamente dos funciones obligatorias: *init()* y *get\_content()* (Figura 2.4).

```
class block_nombre_bloque extends block_base {
    function init() {
        $this->title = get_string('nombre', 'block_nombre_bloque');
        $this->version = 2004111200;
    }

    function get_content() {
        if ($this->content != NULL) {
            return $this->content;
        }

        $this->content = new stdClass;
        $this->content->text = '¡El contenido de nuestro bloque nombre_bloque!';
        $this->content->footer = 'Pie de página aquí...';

        return $this->content;
    }
}
```

Figura. 2.4. Clase base del bloque nombre\_bloque.

Funciones obligatorias (Moodle Docs, 2008):

*init()*

Es obligatorio en la creación de bloques, no debe retornar nada y es necesario asignar valores significativos a las variables *\$this->title*<sup>17</sup> y *\$this->version*<sup>18</sup>.

*get\_content()*

Debe retornar la variable *\$this->content*<sup>19</sup> completamente construida, en caso de ya estarlo al entrar a este método, simplemente se devuelve, en caso contrario se hace necesario poblarla (definir *\$this->content->text* y *\$this->content->footer*, si el bloque deriva de *block\_base*; *\$this->content->items*, *\$this->content->icons* y *\$this->content->footer*, si deriva de *block\_list*).

Otras funciones pueden ser redefinidas con el objetivo de personalizar el funcionamiento de los bloques (Moodle Docs, 2008):

<sup>17</sup> Esta variable se explicará más adelante en este mismo acápite.

<sup>18</sup> Ídem.

<sup>19</sup> Ídem.

*applicable\_formats()*

Permite controlar a cuáles formatos puede ser agregado el bloque en desarrollo. Debe retornar un arreglo con claves que sean los nombres de los formatos de página (*course-view*, *site-index*, *course-format-weeks*, etc. y *all* para los no especificados) y los valores booleanos (*true* o *false*), luego el bloque será permitido en formatos con valor verdadero. El orden en que aparezcan los nombres de formato no hace ninguna diferencia.

*config\_print()*

Permite escoger cómo mostrar la pantalla de configuración global para el bloque en desarrollo. Se debe proveer de una forma para enviar las opciones que irán al formulario (`<input type="submit" />`) y retornar un valor que haga saber el éxito o falla de las acciones internas.

*config\_save()*

Permite sobrescribir el mecanismo de almacenamiento de los datos de configuración. El argumento recibido es un arreglo asociativo, siendo las claves los nombres de la configuración con sus valores de configuración. Debe retornar un valor booleano que muestre si se ejecutaron o no las acciones esperadas.

*has\_config()*

Debe retornar un valor booleano que denota si el bloque presentará una interfaz de configuración al administrador del sitio.

*hide\_header()*

Retorna un valor booleano que denota si el bloque muestra su título.

*html\_attributes()*

Debe retornar un arreglo asociativo de atributos HTML que le serán entregados al elemento contenedor del bloque cuando Moodle construya la salida HTML.

*instance\_allow\_config()*

Debe retornar un valor booleano, indicando si el bloque tendrá configuración por instancia.

*instance\_allow\_multiple()*

Debe retornar un valor booleano, indicando si se quiere permitir múltiples instancias de este bloque en una misma página. Si se permite múltiples instancias, se asume la configuración por instancia y no es necesaria la función `instance_allow_config`.

*instance\_config\_print()*

Permite seleccionar cómo mostrar la pantalla de configuración de instancia para el bloque en desarrollo, es bastante parecido a `config_print()` que se refiere a la configuración global.

*instance\_config\_save(\$data)*

Permite sobrescribir el método de almacenamiento para los datos de configuración de una instancia del bloque. El argumento recibido es un arreglo asociativo de los nombres de configuración con sus valores de configuración. Es el homólogo de `config_save()` para una instancia.

*preferred\_with()*

Debe retornar un valor entero, el cuál es el número de píxeles de ancho que el bloque tomará cuando sea mostrado.

*refresh\_content()*

Debe hacer que el bloque recalcule su contenido inmediatamente, retornando el nuevo valor de `this->content` después de refrescarlo.

*\$this->config*

Guarda los datos de configuración para una instancia de bloque específica (objeto). Es inicializada después de la construcción del objeto bloque. Es posible que el bloque no tenga configuración de instancia, donde la variable será NULL. Existe una relación directa entre esta variable y el campo `configdata` en la tabla `mdl_block_instance`.

*\$this->content\_type*

Especifica el tipo de contenido que tendrá el bloque, diferenciando los bloques de texto de los bloques de lista. Los únicos valores válidos para esta variable son las dos constantes llamadas *BLOCK\_TYPE\_TEXT* y *BLOCK\_TYPE\_LIST*.

*\$this->content*

Guarda el contenido real que se muestra dentro de cada bloque. Los valores válidos son NULL, o un objeto de clase *stdClass*, el cual debe tener variables miembros específicos que corresponden a su tipo, como se explicó en el método *get\_content()*.

*\$this->instance*

Guarda información específica que diferencia una instancia de bloque de otra. Es un objeto del tipo *stdClass* recuperado al llamar *get\_record* en la tabla *mdl\_block\_instance*. Sus miembros variables, entonces, se corresponden directamente con los campos en esa tabla.

*\$this->title*

Es un string (texto plano) que contiene el nombre del bloque que se pone cuando el bloque es mostrado en la pantalla.

*\$this->version*

Debe guardar el número de la versión del bloque en la forma *AAAAMMDDHHmm*, por convención a través de Moodle.

**2.9 C2Site 3.0.1**

C2Site 3.0.1 presenta como principal diferencia con respecto a las versiones anteriores su integración a Moodle en forma de bloque. Éste implementa la clase *block\_ctsite* que redefine los dos métodos necesarios para su integración con la plataforma. El método *get\_content()* se encarga de visualizar las opciones de exportación dependiendo del contexto en que sea invocado. Si el contexto en que se inicia el proceso es SITIO, el bloque posibilita escoger el curso a exportar entre los disponibles en la plataforma; por el contrario, si el contexto es CURSO, éste será el seleccionado para exportar.



Otros métodos no obligatorios redefinidos en la clase:

- *after\_install()*: Esta función es utilizada para verificar la existencia de la *habilidad* que debe tener el usuario que exporta un curso, en caso de no existir, la crea. Además, asegura la existencia de la entrada de lenguaje para esta habilidad. Ésta es invocada por el sistema al finalizar la instalación del bloque.
- *before\_delete()*: Es utilizada para eliminar la *habilidad* creada para exportar un curso. Es ejecutada por el sistema antes de desinstalar el bloque.

Con el objetivo de disminuir la dependencia directa de la estructura de la base de datos, esta versión incorpora el uso de las facilidades que brinda Moodle para gestionar sus contenidos. Para extraer las instancias de los módulos del curso seleccionado, es utilizada la función *get\_all\_instances\_in\_course()* de la librería estándar */lib/datalib.php*. Estas instancias son enviadas a las librerías de los módulos para ser procesadas y construir los recursos finales.

## Conclusiones

En este capítulo se realizó un análisis de la evolución de C2Site, describiendo las principales características utilizadas de las versiones de Moodle para las que fueron desarrolladas, resaltando además los aportes fundamentales introducidos por cada versión de la herramienta. Aunque cada una de estas versiones cumplía con los objetivos que dieron paso a su creación, presentaban algunas deficiencia de configuración y usabilidad.

## CAPÍTULO 3: SOLUCIÓN PROPUESTA

### Introducción

En este capítulo se presenta una descripción de los procesos de exportación de los bloques desarrollados (C2Site y C2SCORM). Se detallan las funciones más utilizadas y la forma de extraer los datos necesarios, apoyados en la notación de los DFD como herramienta fundamental. Además, se analizan las configuraciones ofrecidas por cada bloque, las diferencias y mejoras fundamentales sobre las versiones anteriores.

### 3.1 Aspectos comunes utilizados en la solución

Para darle solución a los objetivos planteados en este trabajo, se determinó crear diferentes bloques, posibilitando entre otras cosas: gestiones independientes, mejor visibilidad, facilidad de actualización y corrección de errores. Estos son C2Site, C2SCORM.

Para la construcción de los bloques propuestos se utiliza básicamente la misma filosofía, partiendo de la necesidad de extraer la información de los recursos y las actividades de Moodle para utilizarla con diferentes propósitos, logrando así reutilizar parte del código de la solución en la implementación de ambos sistemas<sup>20</sup>.

#### 3.1.1 Principales funciones utilizadas

Con el objetivo de lograr una mayor comprensión de la solución propuesta, se hace una descripción detallada de las principales funciones utilizadas.

##### 3.1.1.1 Extracción de información

- `get_records`

Esta función se encuentra en la librería *lib/dmllib.php* de Moodle y retorna un arreglo de objetos, un objeto por cada tupla encontrada en la tabla especificada de la base de datos.

---

<sup>20</sup> Moodle establece como estándar la utilización de caracteres alfanuméricos en los nombres de sus módulos y bloques, por lo que en el código de la aplicación *C2Site* aparece como “*ctsite*” y *C2SCORM* aparece como “*scorm\_generator*”.

Nombre Método	get_records		
Tipo retorno	Arreglo		
Parámetros	Nombre	Tipo	Obligatorios
	\$table	string	x
	\$field	string	
	\$value	string	
	\$sort	string	
	\$fields	string	
	\$limitfrom	int	
	\$limitnum	int	

- `get_record`

Esta función se encuentra en la librería `lib/dmllib.php` de Moodle y retorna una tupla de la Base de Datos en forma de objeto.

Nombre Método	get_record		
Tipo retorno	Objeto		
Parámetros	Nombre	Tipo	Obligatorios
	\$table	string	x
	\$field1	string	x
	\$value1	string	x
	\$field2	string	
	\$value2	string	
	\$field3	string	
	\$value3	string	

- `get_field`

Esta función se encuentra en la librería `lib/dmllib.php` de Moodle y retorna el valor de una columna de la tabla donde los campos especificados coincidan.

Nombre Método	get_field		
Tipo retorno	Valor		
Parámetros	Nombre	Tipo	Obligatorios
	\$table	string	x
	\$return	string	x
	\$field1	string	x
	\$value1	string	x
	\$field2	string	
	\$value2	string	
	\$field3	string	
	\$value3	string	

- `get_all_instances_in_course`

Esta función se encuentra en la librería *lib/datalib.php* de Moodle y retorna un arreglo con todas las instancias de un módulo en particular, en el orden en que fueron definidas.

Nombre Método	get_all_instances_in_course		
Tipo retorno	Arreglo		
Parámetros	Nombre	Tipo	Obligatorios
	\$modulename	string	x
	\$course	string	x
	\$userid	string	
	\$includeinvisible	string	

- `get_all_sections`

Esta función se encuentra en la librería *course/lib.php* de Moodle y retorna un arreglo de objetos con todas las secciones del curso especificado.

Nombre Método	get_all_sections		
Tipo retorno	Valor		
Parámetros	Nombre	Tipo	Obligatorios
	\$courseid	int	x

### 3.1.1.2 Verificaciones generales

- `< nombre_bloque >21_valid_mods()`

Esta función se encuentra en la librería *lib.php* de los bloques desarrollados y retorna un arreglo con los módulos permitidos para realizar el proceso de exportación.

Nombre Método	<code>&lt; nombre_bloque &gt;_valid_mods()</code>
Tipo retorno	Arreglo

- `< nombre_bloque >_section_available`

Esta función se encuentra en la librería *lib.php* de los bloques desarrollados, es la encargada de verificar si una sección está disponible para exportar, creando para esto un arreglo con los módulos elegidos y devolviéndolo por referencia. Además, retorna verdadero si existe alguna sección disponible para exportar y falso en caso contrario.

Nombre Método	<code>&lt; nombre_bloque &gt;_section_available</code>		
Tipo retorno	bool		
Parámetros	Nombre	Tipo	Obligatorios
	\$sect	Objeto	x
	\$course_modules	Arreglo	x

<sup>21</sup> Referencia al nombre de los bloques desarrollados.

### 3.1.1.3 Manejo de ficheros

- *<nombre\_bloque>\_copy\_file*

Función que se encuentra en el fichero *lib.php* de los bloques desarrollados, es la encargada de copiar un fichero. Además verifica que el directorio donde se va a copiar el archivo exista y en caso contrario lo crea recursivamente. Se usa en sustitución de la función *copy*, que define PHP.

Nombre Método	<i>&lt;nombre_bloque&gt;_copy_file</i>		
Tipo retorno	bool		
Parámetros	Nombre	Tipo	Obligatorios
	<i>\$from_file</i>	<i>string</i>	x
	<i>\$to_field</i>	<i>string</i>	x

- *<nombre\_bloque>\_copy\_dir*

Función que se encuentra en el fichero *lib.php* de los bloques desarrollados, es la encargada de copiar recursivamente un directorio en otro. Verifica que el directorio destino exista y en caso contrario lo crea recursivamente.

Nombre Método	<i>&lt;nombre_bloque&gt;_copy_dir</i>		
Tipo retorno	bool		
Parámetros	Nombre	Tipo	Obligatorios
	<i>\$from_file</i>	<i>string</i>	x
	<i>\$to_field</i>	<i>string</i>	x

- *<nombre\_bloque>\_user\_image*

Función que se encuentra en el fichero *lib.php* de los bloques desarrollados y es la encargada de copiar la imagen de un usuario retornando su referencia.

Nombre Módulo	<i>&lt;nombre_bloque&gt;_user_image</i>		
Tipo retorno	string		

Parámetros	Nombre	Tipo	Obligatorios
	\$user_id	int	x

- *<nombre\_bloque>\_icons*

Función que se encuentra en el fichero *lib.php* de los bloques desarrollados, es la encargada de copiar los íconos de las actividades y retornar una referencia a los mismos. En caso de tratarse de un recurso, un objeto con su contenido es pasado como segundo parámetro con información necesaria, para poder tratar el tipo de recurso al que representa.

Nombre Método	<i>&lt;nombre_bloque&gt;_icons</i>		
Tipo retorno	string		
Parámetros	Nombre	Tipo	Obligatorios
	\$modname	string	x
	\$mod	object	

### 3.1.1.4 Funciones generales

- *< nombre\_bloque >\_extract\_mods\_information*

Esta función se encuentra en la librería *lib.php* de los bloques desarrollados, es la encargada de devolver por referencia un objeto con todas las configuraciones necesarias para exportar el curso.

Nombre Método	<i>&lt; nombre_bloque &gt;_extract_mods_information</i>		
Tipo retorno	bool		
Parámetros	Nombre	Tipo	Obligatorios
	\$preferences	Objeto	x
	\$course	Objeto	x

- `<nombre_bloque>_filter`

Función que se encuentra en el fichero *lib.php* del bloque C2Site. Es la encargada de filtrar los bloques de texto que puedan contener imágenes, copiarlas y devolver una cadena en la que las referencias a las mismas estén sustituidas por otra con referencia a su nueva ubicación.

Nombre Método	<code>&lt;nombre_bloque&gt;_filter</code>		
Tipo retorno	string		
Parámetros	Nombre	Tipo	Obligatorios
	<code>\$content</code>	string	X
	<code>\$striphtml</code>	bool	
	<code>\$level</code>	int	

- `<nombre_bloque>_filter2`

Función recursiva que se encuentra en el fichero *lib.php* del bloque C2Site. Es invocada en primera instancia por la función `<nombre_bloque>_filter` y se encarga de asegurar algunas de sus funcionalidades.

Nombre Método	<code>&lt;nombre_bloque&gt;_filter2</code>		
Tipo retorno	void		
Parámetros	Nombre	Tipo	Obligatorios
	<code>\$content</code>	string	X
	<code>\$references</code>	array	X

### 3.1.2 Configuraciones de los bloques desarrollados

La configuración es un conjunto de datos que determina el valor de algunas variables de un programa o sistema de software. Estas opciones generalmente son cargadas al inicio de la ejecución, definiendo las reacciones del software y particularizando el proceso. 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.

Las configuraciones de los bloques en Moodle pueden ser accedidas mediante el panel de “Administración del sitio”, siguiendo la secuencia: “Módulos”>> “Bloques”, en donde se muestra un listado de bloques instalados, permitiendo acceder a las diferentes configuraciones, siempre que estas estén implementadas.

A continuación se muestran las configuraciones con que cuentan los bloques desarrollados, una breve descripción de éstas, además de especificar su valor por defecto y a qué bloque pertenecen.

#### Configuraciones Globales:

1. *Nombre en C2Site:* block\_ctsite\_export\_empty\_sections  
*Nombre en C2SCORM:* scorm\_generator\_export\_empty\_sections  
*Valor por defecto:* No  
*Descripción:* Encargada de informar al sistema si el usuario desea que las secciones que no tengan actividades disponibles para exportar, entren en el *proceso*.
2. *Nombre en C2Site:* block\_ctsite\_export\_hidden\_sections  
*Nombre en C2SCORM:* scorm\_generator\_export\_hidden\_sections  
*Valor por defecto:* No  
*Descripción:* Encargada de informar al sistema si el usuario desea que las secciones ocultas entren en el proceso de exportación.
3. *Nombre en C2Site:* block\_ctsite\_export\_hidden\_activities  
*Nombre en C2SCORM:* scorm\_generator\_export\_hidden\_activities  
*Valor por defecto:* No  
*Descripción:* Encargada de informar al sistema si el usuario desea que las actividades ocultas entren en el proceso de exportación.

## Configuraciones de los Foros:

4. *Nombre en C2Site:* block\_ctsite\_foro\_post  
*Nombre en C2SCORM:* scorm\_generator\_foro\_post  
*Valor por defecto:* Si  
*Descripción:* Encargada de informar al sistema si el usuario desea exportar las discusiones de los foros.
  
5. *Nombre en C2Site:* block\_ctsite\_foro\_whithout\_posts  
*Nombre en C2SCORM:* scorm\_generator\_foro\_whithout\_posts  
*Valor por defecto:* Si  
*Descripción:* Encargada de informar al sistema si el usuario desea exportar los foros sin discusiones.
  
6. *Nombre en C2Site:* block\_ctsite\_foro\_user\_images  
*Nombre en C2SCORM:* scorm\_generator\_foro\_user\_images  
*Valor por defecto:* Si  
*Descripción:* Encargada de informar al sistema si el usuario desea exportar las imágenes de los usuarios en los post.
  
7. *Nombre en C2Site:* block\_ctsite\_foro\_attachment  
*Nombre en C2SCORM:* scorm\_generator\_foro\_attachment  
*Valor por defecto:* No  
*Descripción:* Encargada de informar al sistema si el usuario desea exportar los ficheros adjuntos de los mensajes.

## Configuraciones de los Glosarios:

8. *Nombre en C2Site:* block\_ctsite\_glossary\_aliases  
*Nombre en C2SCORM:* scorm\_generator\_glossary\_aliases  
*Valor por defecto:* No

*Descripción:* Encargada de informar al sistema si el usuario desea exportar los alias<sup>22</sup> en las entradas del glosario.

9. *Nombre en C2Site:* block\_ctsite\_glossary\_ratings

*Nombre en C2SCORM:* scorm\_generator\_glossary\_ratings

*Valor por defecto:* No

*Descripción:* Encargada de informar al sistema si el usuario desea exportar las calificaciones de las entradas del glosario.

10. *Nombre en C2Site:* block\_ctsite\_glossary\_attachment

*Nombre en C2SCORM:* scorm\_generator\_glossary\_attachment

*Valor por defecto:* No

*Descripción:* Encargada de informar al sistema si el usuario desea exportar los ficheros adjuntos en las entradas del glosario.

Configuraciones de las Wiki:

11. *Nombre en C2Site:* block\_ctsite\_wiki\_entries

*Nombre en C2SCORM:* scorm\_generator\_wiki\_entries

*Valor por defecto:* Si

*Descripción:* Encargada de informar al sistema si el usuario desea exportar las entradas de las Wiki.

12. *Nombre en C2Site:* block\_ctsite\_wiki\_author\_data

*Nombre en C2SCORM:* scorm\_generator\_wiki\_author\_data

*Valor por defecto:* No

*Descripción:* Encargada de informar al sistema si el usuario desea exportar los datos de usuario en cada entrada, incluyendo su foto.

13. *Nombre en C2Site:* block\_ctsite\_workshop\_empty\_submissions

*Nombre en C2SCORM:* scorm\_generator\_workshop\_empty\_submissions

---

<sup>22</sup> Sinónimos del término

*Valor por defecto:* No

*Descripción:* Encargada de informar al sistema si el usuario desea exportar envíos vacíos del taller.

### 3.1.3 Estructura de la información exportada

En la siguiente tabla (Tabla 3.0) se muestran un conjunto de módulos existentes para la versión 1.8.2+ 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.

Módulos	Estructura que se extrae	Por Defecto	Observaciones
assignment	nombre descripción	x	Sólo permite visualizar la tarea.
book	nombre descripción capítulos título contenido	—	
forum	nombre descripción discusiones nombre asunto mensaje autor respuestas asunto mensaje autor	x	

glossary	nombre descripción entradas concepto definición fichero adjunto alias calificación	X	
label	nombre	x	
lesson	nombre descripción páginas título contenido respuestas	x	Sólo permite visualizar las preguntas.
quiz	nombre descripción tiempo límite preguntas nombre texto de la pregunta respuestas	x	No se califican automáticamente. Las preguntas de tipo “ <i>Emparejamiento aleatorio de respuestas cortas</i> ” no son exportadas.
resource	nombre descripción contenido	x	
scorm	nombre descripción recursos	x	Sólo permite visualizar el paquete.
wiki	nombre descripción	x	Sólo permite visualizar los contenidos.

	entradas páginas nombre contenido versión autor		
workshop	nombre descripción entradas descripción envíos de usuarios autor título descripción	x	Sólo permite visualizar los envíos realizados.

Tabla 3.1: *Estructura de la información exportada.*

### 3.1.4 Diagramas de flujo de datos

Como es conocido, la información se transforma a medida que fluye por un sistema basado en computadora. Un sistema acepta entradas sobre las que aplica elementos de hardware, software y humanos para transformarlas en salidas, produciéndolas en una gran variedad de formas.

Se puede, de forma efectiva, construir un modelo del flujo de la información para cualquier sistema de computadora, independientemente del tamaño y la complejidad del mismo. Para ello se cuenta con diferentes herramientas entre las que se encuentran los Diagramas de Flujo de Datos.

Un diagrama de flujo de datos es una técnica gráfica que sirve para construir un modelo lógico del sistema, mostrando el recorrido y la transformación de los datos dentro del mismo, estos

permiten visualizar un sistema como una red de procesos funcionales, conectados entre sí por conductos y tanques de almacenamiento de datos.

Los diagramas derivados de los procesos principales se clasifican en niveles, los cuales son: (Wikimedia Foundation, Inc., 2008)

- Nivel 0: Diagrama de contexto

En el diagrama de contexto sólo se modela o dibuja el proceso principal del problema con sus respectivas entidades. Cada proceso debe tener al menos una entrada y una salida (de datos).

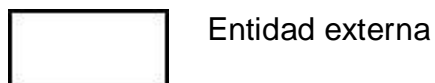
- Nivel 1: Diagrama 0 o de nivel superior

En este nivel se plasman todos los procesos que describen al proceso principal.

- Nivel 2: Diagrama de detalle o expansión

En el diagrama de detalle se generan procesos provenientes del nivel anterior.

Los estereotipos utilizados en el presente trabajo son los siguientes:



Con el objetivo de explicar las características lógicas de los flujos de datos, las entidades externas, almacenes y procesos representados en el DFD, se hace uso en este trabajo del Diccionario de Datos como herramienta complementaria.

Para la escritura del diccionario se usan símbolos especiales con la finalidad de limitar la cantidad de texto necesario empleado para describir las relaciones entre los datos y al mismo tiempo mostrar con claridad las relaciones estructurales. La notación empleada es la siguiente (Rodríguez, 2008):

Símbolo	Significado	Explicación	Uso
=	Es equivalente a	Alias	Denota sinónimos
+	Y	Concatenación, componentes que siempre están incluidos en una estructura	Denota una relación de secuencia
[]	Uno u otro	Define opciones entre los componentes de una estructura	Denota una relación de selección
{}	Iteraciones de	Define la repetición de un componente de la estructura	Denota una relación de iteración
()	Opcional	Define componentes de la estructura que puede o no estar presente una sola vez	Denota una relación opcional

Tabla 3.2. Notación empleada en el diccionario de datos.

Con el objetivo de representar una descripción detallada de los procesos más importantes y de más bajo nivel, en el presente trabajo se utilizará pseudocódigo. Para esto se describirán los posibles algoritmos utilizando una mezcla de frases en lenguaje común, instrucciones de programación y palabras claves.



## 3.2 C2Site

C2Site (Courses To Site), es un bloque programado para Moodle 1.8.2+. El objetivo principal de C2Site es exportar cursos, que posean un formato por semanas o por temas, en forma de sitio web estático. Dicho sitio puede contener uno o varios cursos, según lo estime el usuario al cual se le hayan dado los permisos necesarios para exportarlos.

### 3.2.1 Diferencias principales de esta versión con las anteriores

Tomando como experiencia versiones anteriormente desarrolladas y teniendo en cuenta el poco conocimiento sobre posibles cambios o agregaciones en la forma de gestionar los contenidos de los cursos por la plataforma, surgió la necesidad de desarrollar un sistema capaz de adaptarse fácilmente a posteriores versiones de Moodle.

Con este objetivo se utiliza de forma óptima las potencialidades que brinda Moodle para su extensión y personalización, previendo de cierta forma que cambios, como el de la arquitectura de la base de datos, puedan interferir con el funcionamiento de los bloques. Constituye ésta la mayor diferencia de la actual versión con las anteriores.

Otra diferencia sustancial está determinada por las configuraciones agregadas al bloque, las cuales permiten personalizar una serie de parámetros que se tienen en cuenta al generar el sitio web. Además, se le agregó al bloque un sistema de administración de cursos exportados, que permite gestionar los cursos exportados por el sistema, y del cual se hablará en detalle más adelante.

### 3.2.2 Arquitectura del bloque

Con el objetivo de modelar la solución propuesta, se procede a la elaboración y descripción del modelo de datos y de procesos (DFD) como parte del análisis del sistema de información (ASI) del bloque. Para ello, se realiza una descomposición de dichos procesos siguiendo un enfoque descendente (*top-down*<sup>23</sup>), en varios niveles de abstracción, donde cada nivel proporciona una visión más detallada del proceso definido en el nivel anterior.

---

<sup>23</sup> Que parte de un alto nivel (general) y obra hacia los niveles inferiores (más específicos).

Con el fin de representar el sistema completo y los flujos de datos que muestran las interfaces del sistema con los terminadores externos (junto con los almacenes externos) se muestra en la Figura 3.1 el Diagrama de Contexto del sistema.

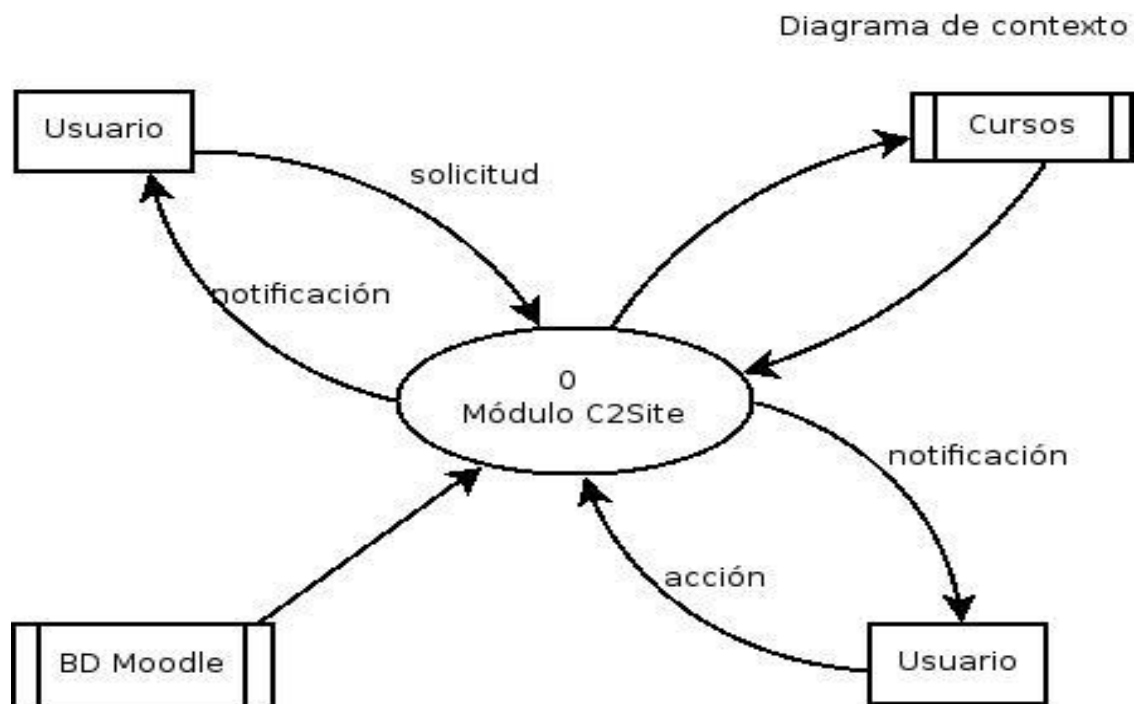


Figura. 3.1. *Diagrama de Contexto*

Como se muestra en la figura anterior, el único agente externo que interactúa con el sistema es el usuario, que solicita exportar un curso o administrar los cursos exportados. Estos acontecimientos constituyen los flujos de datos que inician todo el proceso, éstos a su vez pueden invocar otros, conformándose así una red de procesos de la cual representamos el nivel superior en la Figura 3.2.

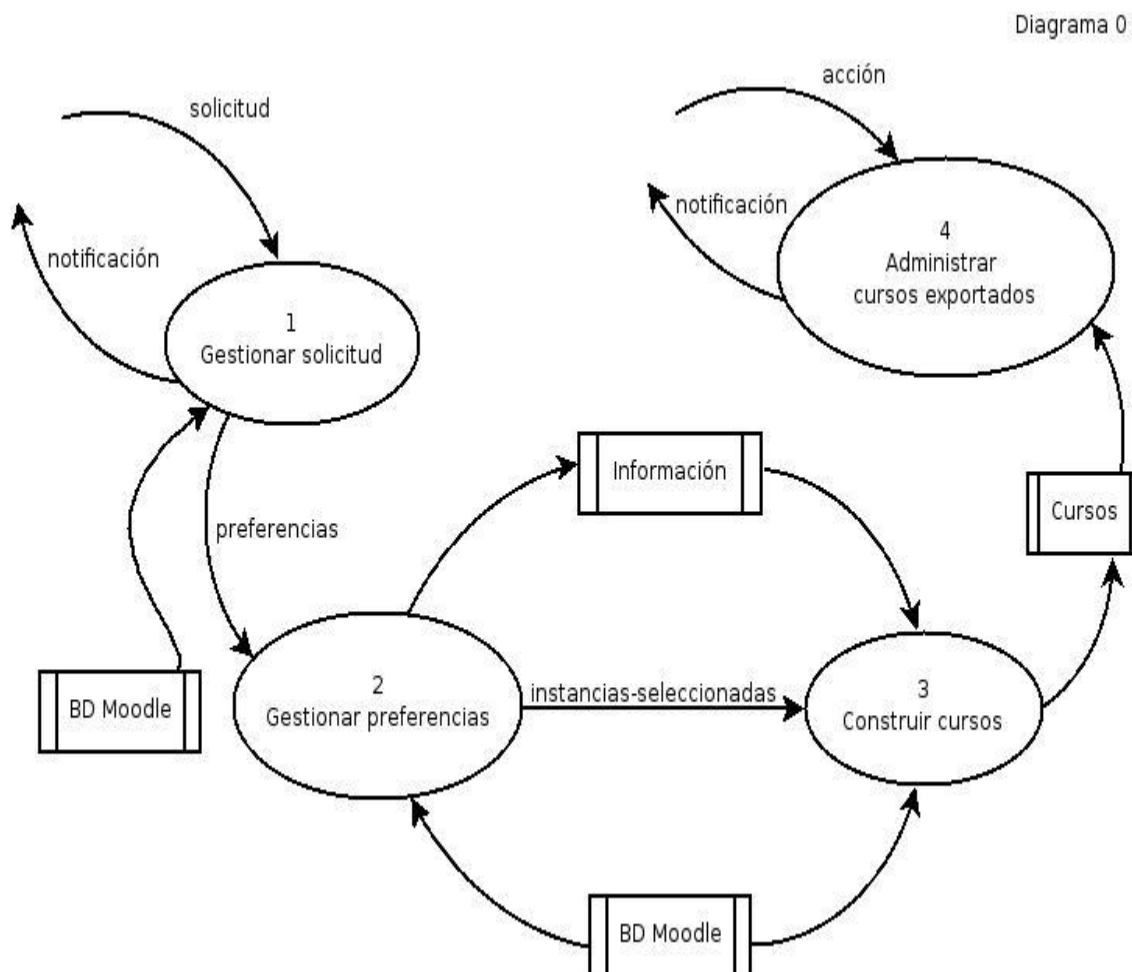


Figura. 3.2. Diagrama 0 o de Nivel Superior

En el anterior diagrama de contexto se destacan los principales procesos del sistema, sus relaciones entre sí y con los almacenes. El proceso se inicia en “Gestionar solicitud” donde después de recibir una solicitud el sistema permite personalizar la exportación teniendo en cuenta las preferencias de cada usuario, las cuales son enviadas a “Gestionar preferencias”, quién a su vez se encarga de ofrecerle al proceso “Construir curso” todos los datos necesarios para culminar la exportación.

Por otro lado, el proceso “Administrar cursos exportados” se encarga de manejar los cursos previamente exportados por el sistema. El mismo puede ser ejecutado independientemente del resto del proceso de exportación.

### 3.2.2.1 Gestionar solicitud del usuario

El proceso “Gestionar solicitud”, representado en la Figura 3.3, se encarga de recopilar toda la información necesaria para exportar un curso personalizado por el usuario. Al recibir la solicitud, el proceso brinda una serie de parámetros que el usuario puede configurar de acuerdo a sus preferencias.

Diagrama 1

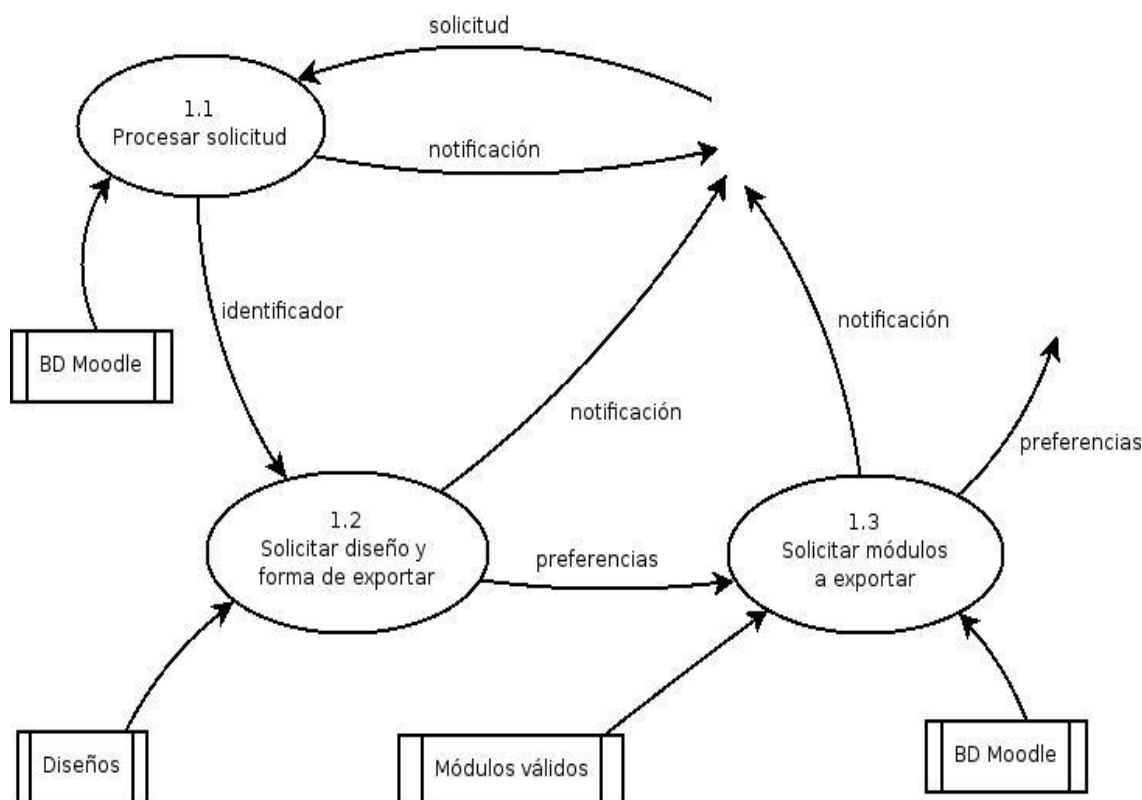


Figura. 3.3 Diagrama 1 (Gestionar solicitud)

### 3.2.2.2 Gestionar preferencias escogidas

En este proceso, el sistema utiliza las preferencias del usuario para crear la ruta y obtener las instancias de los módulos que serán exportados.

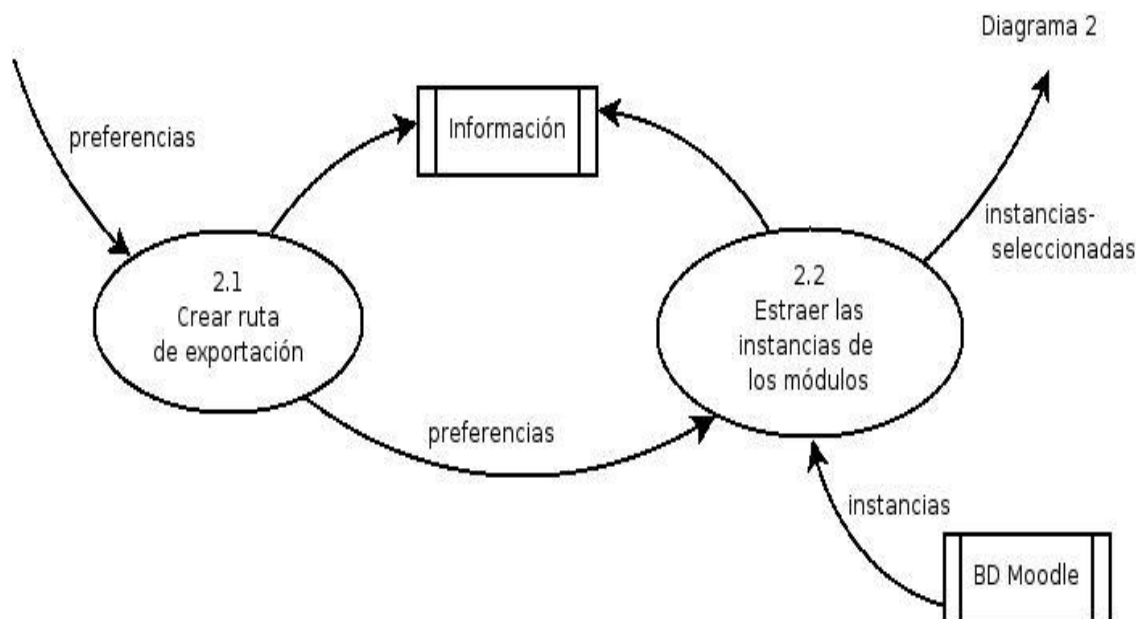


Figura. 3.4 Diagrama 2 (Gestionar preferencias)

### 3.2.2.3 Construir sitio web

A partir de las instancias de los módulos seleccionados por el usuario, el sistema exporta cada uno de ellos y crea un sitio web estático que los integra.

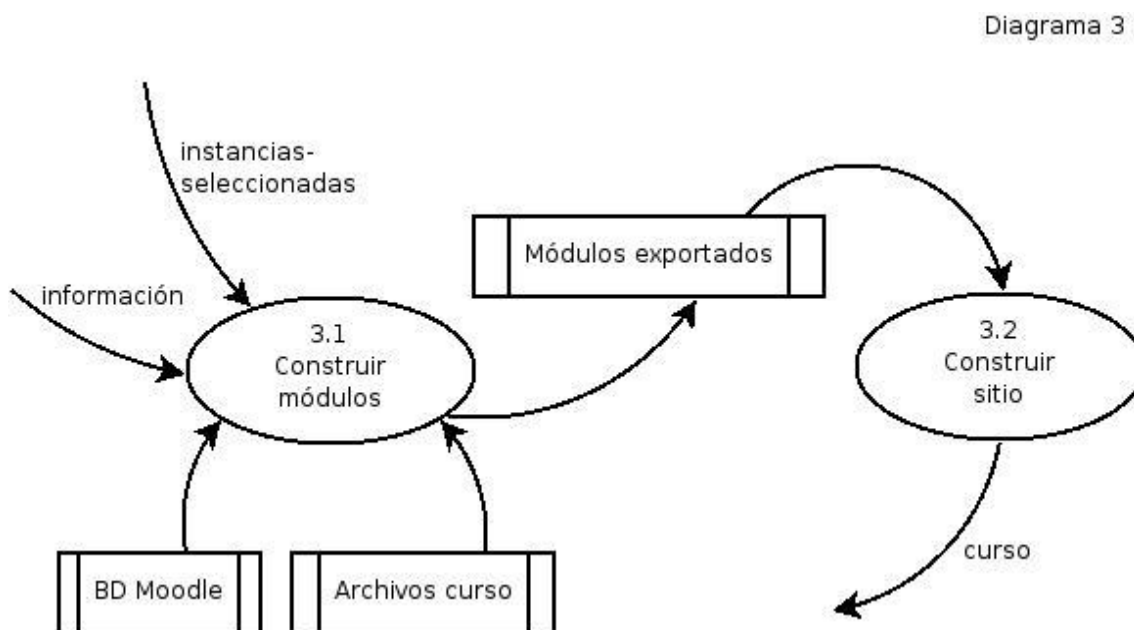


Figura. 3.5 Diagrama 3 (Construir curso)

Debido al nivel de complejidad y la importancia que tiene, se hace necesario expandir el proceso 3.1. En la siguiente figura, se muestra el flujo de información requerido para la construcción de los módulos seleccionados.

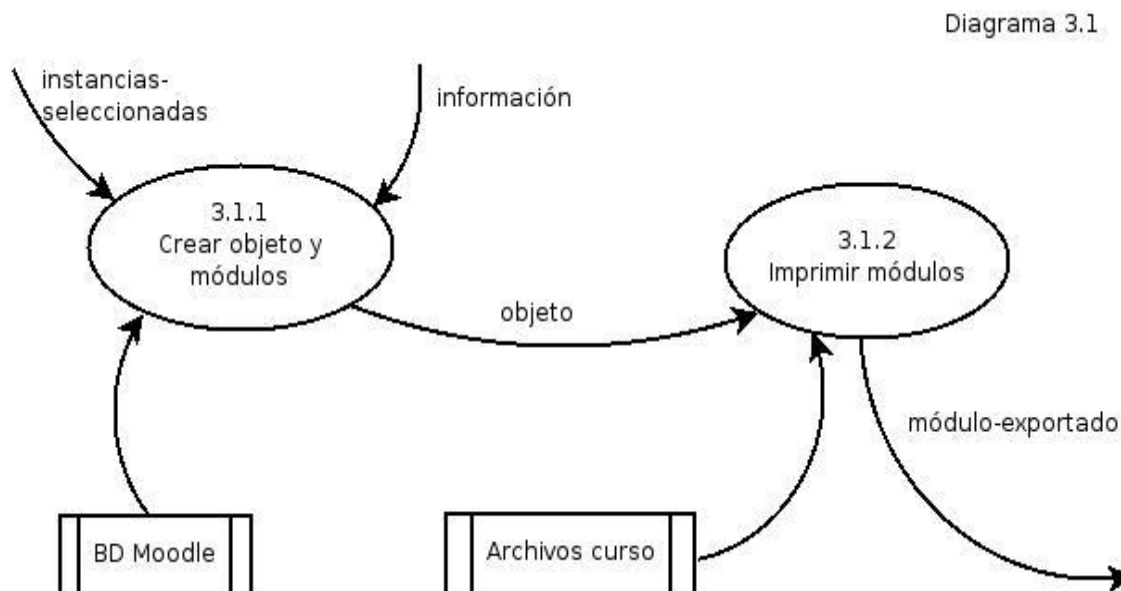


Figura. 3.6 Diagrama 3.1 (Construir módulos)

#### 3.2.2.4 Panel de administración de los cursos exportados

Como se dijo anteriormente, este proceso es una de las nuevas funcionalidades de la herramienta, que trabaja de forma independiente del resto de la aplicación. Permite gestionar con facilidad los cursos exportados posibilitando, entre otras cosas, reutilizar cursos previamente exportados.

Entre sus funcionalidades se encuentran:

- *Crear nuevo paquete*  
Un paquete solo puede ser creado en el directorio raíz de los cursos exportados. Dentro de un paquete no se puede crear otro.
- *Renombrar paquete*  
Permite renombrar un paquete de cursos existente.
- *Copiar*

Permite copiar cursos entre paquetes aumentando la reutilización de estos. Pueden copiarse varios cursos a la vez.

- *Mover*

Permite mover uno o varios cursos entre paquetes.

- *Eliminar*

Permite eliminar cursos de un paquete, así como paquetes en general.

- *Compactar*

Permite compactar y descargar un paquete de cursos, o cursos de forma independiente.

Para administrar los cursos exportados es necesario que el usuario tenga los permisos requeridos para administrar archivos (moodle/course:managefiles) en el contexto del sitio.

### 3.2.2.5 Diccionario de Datos

Con el objetivo de definir la terminología utilizada en los diagramas de flujo de datos del bloque, se hace uso del siguiente diccionario de datos.

Entidad Externa: Usuario

Descripción: Cualquier usuario con el permiso “moodle/site:export”, necesario para realizar el proceso de exportación.

Estructura: [Estudiante| Administrador| Creadores de curso| Profesor |Otros]

Flujo de datos: solicitud

Descripción: Solicitar exportar un curso en forma de sitio web.

Flujo de datos: notificación

Descripción: Respuesta del sistema ante diferentes eventos.

Estructura: [error al exportar| éxito en la exportación| cancelar el proceso| solicitud de acción]

Almacén: BD Moodle

Descripción: Base de Datos de Moodle.

Flujo de datos: sección

Descripción: Forma de agrupar los módulos dentro de un curso, generalmente por temas o semanas.

Estructura: numero de sección+ descripción+ módulos

Flujo de datos: curso

Descripción: Conjunto de información relacionada con una materia que en Moodle está compuesto por módulos.

Estructura: identificador + nombre-curso + descripción + sección

Almacén: Cursos

Estructura: {curso}

Proceso: Gestionar Solicitud

Descripción: Personalizar el proceso de exportación.

Flujo de datos: paquete-cursos

Descripción: Conjunto de cursos exportados unidos en un sitio.

Flujo de datos: preferencias

Descripción: Datos que el usuario selecciona para personalizar el proceso de exportación.

Estructura: diseño + forma-exportar + nombre-categoría + (paquete-cursos) + módulos

Proceso: Gestionar preferencias

Descripción: Crear la ruta de exportación y extraer de la BD Moodle todas las instancias de los módulos seleccionados por el usuario.

Flujo de datos: código

Descripción: Este valor se genera teniendo en cuenta el identificador de sección de usuario y el tiempo actual en microsegundos. Garantiza que cada proceso de exportación sea único.



Flujo de datos: camino

Descripción: Directorio raíz donde se exportará el curso.

Flujo de datos: forma-exportar

Descripción: Controla si el curso va a ser exportado en un paquete de cursos o de forma independiente.

Flujo de datos: diseño

Descripción: Diseño seleccionado por el usuario con el que va a ser exportado el curso.

Almacén: Diseños

Estructura: {diseño}

Flujo de datos: nombre-categoría

Descripción: Nombre de la categoría a la que pertenece el curso.

Almacén: Información

Descripción: Esta información es guardada en una variable de sesión, en la cual se almacenan los datos generales para exportar el curso.

Estructura: {@código}1 + {camino}1 + {forma-exportar}1 + {diseño}1 + {nombre-categoría}1

Flujo de datos: Instancia

Descripción: Objeto con la información básica del módulo, extraída de la BD Moodle.

Estructura: identificador+ nombre+ descripción

Instancias= {instancia}

Flujo de datos: instancias-seleccionadas

Descripción: Instancias de los módulos seleccionados por el usuario para ser exportados.

Proceso: Construir curso

Descripción: Obtiene de la BD Moodle toda la información necesaria y construye cada uno de los módulos, así como el sitio en general.

Proceso: Administrar cursos exportados

Descripción: Subsistema de gestión de los cursos exportados.

Proceso: Procesar solicitud

Descripción: Se encarga de verificar la existencia de un identificador de curso, en caso contrario muestra un listado de cursos de donde el usuario seleccione.

Flujo de datos: identificador

Descripción: Identificador del curso.

Proceso: Solicitar diseño y forma de exportar

Descripción: Da la posibilidad al usuario de seleccionar la forma de exportar el curso, el diseño y la categoría.

Proceso: Solicitar módulos a exportar

Descripción: Construir el listado de módulos disponibles, para que el usuario seleccione los que desea exportar. El listado puede ser ordenado por secciones (temas) o por tipo de módulo.

Proceso: Crear ruta de exportación

Descripción: Construye la ruta donde se exportará el curso según las preferencias. Cuando es hacia un paquete de cursos los datos se almacenarán en los archivos de sitio, en caso contrario en los archivos del curso.

Proceso: Extraer las instancias de los módulos

Descripción: Extraer de la BD Moodle los objetos de los módulos seleccionados por el usuario.

Proceso: Construir módulos

Descripción: Construye un objeto con toda la información del módulo en cuestión y lo imprime.

Flujo de datos: módulo

Descripción: Actividad o recurso que representa una unidad de aprendizaje.

Almacén: Archivos del curso

Descripción: Ficheros pertenecientes al curso.

Proceso: Construir sitio

Descripción: Integrar todas los “Módulos exportados” en un sitio web.

Flujo de datos: módulos-válidos

Descripción: Módulos que el sistema es capaz de exportar.

Proceso: Crear objeto de módulo

Descripción: Construye un objeto con toda la información de la actividad o recurso que representa el módulo.

Flujo de datos: objeto

Descripción: Objeto con toda la información de la actividad o recurso que representa el módulo.

Proceso: Imprimir módulo

Descripción: Construye y almacena en formato HTML los módulos.

Flujo de datos: módulo-exportado

Descripción: Contenido de un módulo impreso.

Almacén: Módulos exportados

Estructura: {módulo-exportado}

Flujo de datos: acción

Descripción: Solicitud del usuario que determina la acción a realizar en la administración de los cursos exportados.

Estructura: [renombrar| crear paquete| eliminar| mover| copiar| compactar]

### 3.2.2.6 Pseudocódigos más importantes

Representación en pseudocódigo de los principales procesos que forman parte de C2Site.

- Proceso: 1.1- **Procesar solicitud**

COMIENZA

SI **identificador** EXISTE

    DESPLEGAR (proceso 1.2) **Solicitar diseño y forma de exportar**

OTRO

    EXTRAER **cursos** de **BD Moodle**

    MIENTRAS haya **curso** en **cursos**

**MOSTRAR curso**

    FIN\_MIENTRAS

    DESPLEGAR **identificador**

FIN\_SI

FINALIZA

- Proceso: 1.2- **Solicitar diseño y forma de exportar**

COMIENZA

MIENTRAS haya **diseño** en **Diseños**

**MOSTRAR diseño**

FIN\_MIENTRAS

MOSTRAR formas de exportar

SI **forma-exportar** = “varios”

**MOSTRAR paquete-cursos**

FIN\_SI

MOSTRAR **nombre-categoría**

DESPLEGAR **preferencias**

FINALIZA

- Proceso: 1.3- **Solicitar módulos a exportar**

COMIENZA

MIENTRAS haya **módulo** en **módulos-válidos**

EXTRAER **instancias** de **módulo** en **BD Moodle**

MIENTRAS haya **instancia** en **instancias**

MOSTRAR **instancia**

FIN\_MIENTRAS

FIN\_MIENTRAS

DESPLEGAR **preferencias**

FINALIZA

- Proceso: 2.1- **Crear ruta de exportación**

COMIENZA

SI no existe **código**

DESPLEGAR **notificación** “Missing data”

FIN\_SI

SI **forma-exportar** = “varios”

**camino** = ruta dentro de los archivos del sitio

OTRO

**camino** = ruta dentro de los archivos del curso

FIN\_SI

TERMINA

- Proceso: 2.2- **Extraer instancias de módulos**

COMIENZA

MIENTRAS haya **módulo** en **módulos-válidos**

EXTRAER **instancias** de **modulo** en **BD Moodle**

MIENTRAS haya **instancia** en **instancias**

SI **instancia** en **preferencias**

**instancias-seleccionadas = instancias**

FIN\_SI

FIN\_MIENTRAS

FIN\_MIENTRAS

DESPLEGAR **instancias-seleccionadas**

FINALIZA

- Proceso: 3.1.1- **Crear objeto de módulo**

COMIENZA

**objeto** = “datos pasados por parámetro provenientes de la instancia”MIENTRAS haya más información importante del modulo en **BD Moodle**AGREGAR a **objeto** información importante del modulo en **BD Moodle**

FIN\_ MIENTRAS

DESPLEGAR **objeto**

FINALIZA

- Proceso 3.1.2 **Imprimir módulo**

COMIENZA

MIENTRAS haya información importante del **modulo** en el **objeto**GUARDAR información del **objeto**

FIN\_ MIENTRAS

DESPLEGAR **módulo**

FINALIZA

- Proceso: 3.2 **Construir sitio**

COMIENZA

MIENTRAS haya **sección** en **curso** en **BD Moodle**MIENTRAS haya **módulo** en **sección**IMPRIMIR nombre de **módulo** con vínculo al **módulo-exportado**

FIN\_ MIENTRAS

FIN\_ MIENTRAS

DESPLEGAR **curso**

FINALIZA

- Proceso: 4- **Administrar cursos exportados**

COMIENZA

CASO **acción**

CASO **acción** = “renombrar”

COMPROBAR existencia **paquete-curso**

RENOMBRAR **paquete-curso**

CASO **acción** = “crear paquete”

SI existe nombre de **paquete-curso**

COMPROBAR no existencia **paquete-curso**

CREAR **paquete-curso**

OTRO

MOSTRAR interfaz para crear **paquete-curso**

FIN\_SI

CASO **acción** = “eliminar”

SI existe **confirmación**

COMPROBAR existencia **curso** o **paquete-curso**

ELIMINAR **curso** o **paquete-curso**

OTRO

MOSTRAR mensaje de **confirmación**

FIN\_SI

CASO **acción** = “mover”

SI existe dirección de **curso**

MOVER **curso**

FIN\_SI

CASO **acción** = “copiar”

SI existe dirección de **curso**

COPIAR **curso**

```

    FIN_SI
CASO acción = "compactar"
    SI existe curso o paquete-curso
        COMPACTAR curso o paquete-curso
    FIN_SI
FIN_CASO
FINALIZA

```

### 3.3 C2SCORM

C2SCORM (Course To SCORM), es un bloque que se le incorpora a Moodle 1.8.2+, que tiene como objetivo principal exportar cursos con formato por temas o por semanas en forma de paquete SCORM, con la finalidad de compartir, reutilizar y exportar los contenidos de Moodle, permitiendo reutilizarlos en cualquier plataforma compatible.

Entre sus principales funcionalidades se encuentran las siguientes:

- Exportar un curso de Moodle posibilitando seleccionar los recursos y actividades involucrados en el proceso.
- Permitir el llenado manual de metadatos a los recursos.
- Configurar el proceso de exportación a su preferencia.
- Descargar el paquete previamente exportado.

#### 3.3.1 Arquitectura del bloque

Este módulo se describe siguiendo los mismos principios utilizados para modelar el módulo C2Site. Con el fin de representar la vista de más alto nivel de las principales funciones del sistema, así como los flujos de datos que muestran las interfaces del sistema con los terminadores externos (junto con los almacenes externos), se muestra en la Figura 3.7 el Diagrama de Contexto del sistema.



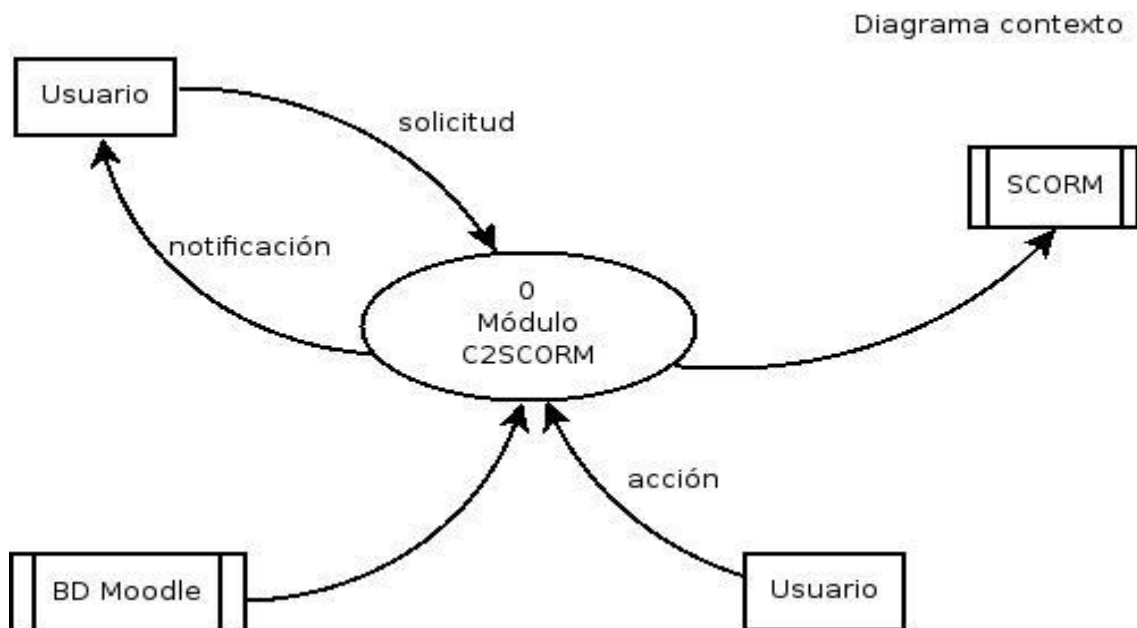


Figura. 3.7. Diagrama de Contexto

A continuación se presenta (Figura 3.8) la primera descomposición funcional del sistema. Al recibir la solicitud de exportar un curso, el sistema brinda la posibilidad de personalizar los recursos que conformarán el paquete antes de finalizar el proceso de exportación.

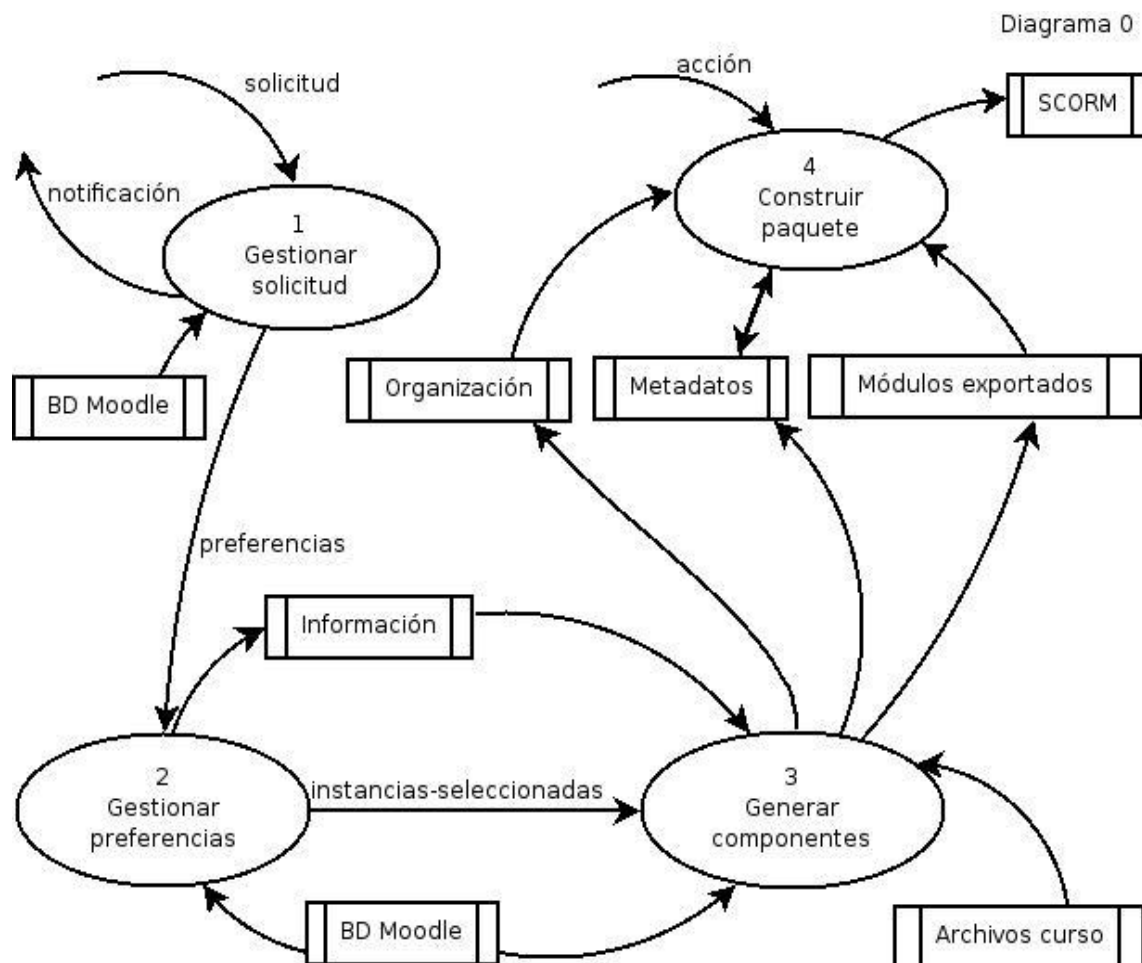


Figura. 3.8. Diagrama 0 o de Nivel Superior

### 3.3.1.1 Gestionar solicitud del usuario

En la siguiente figura (3.9) se muestra la descomposición de proceso “Gestionar solicitud”, donde se verifican los datos necesarios para personalizar la solicitud del usuario.

Diagrama 1

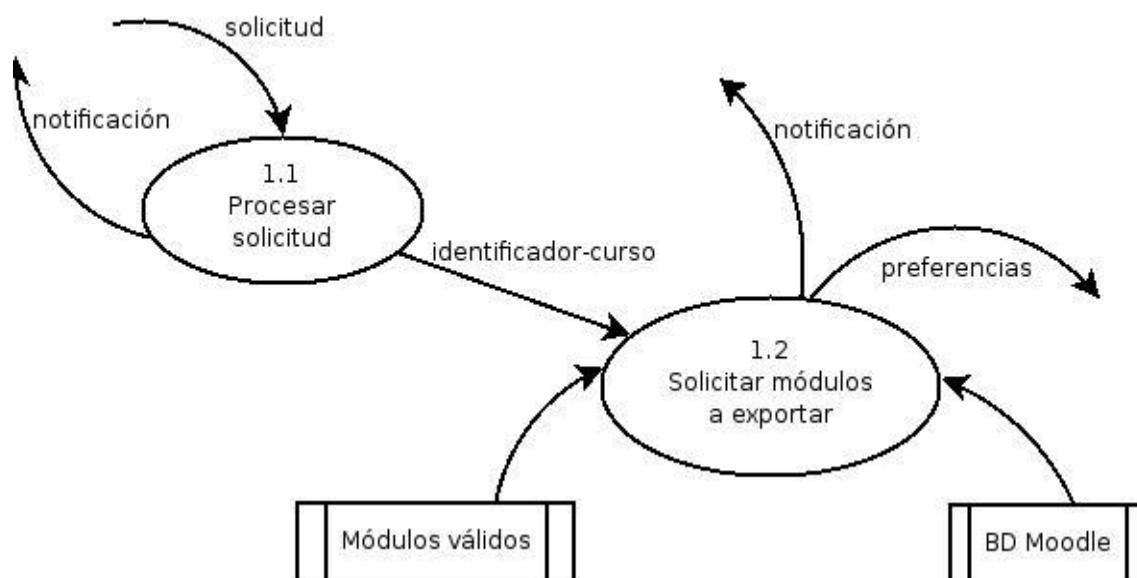


Figura. 3.9. Diagrama 1 (Gestionar solicitud)

### 3.3.1.2 Gestionar Preferencias

En este proceso, el sistema utiliza las preferencias del usuario para construir las rutas necesarias y obtener las instancias de los módulos que serán exportados.

Diagrama 2

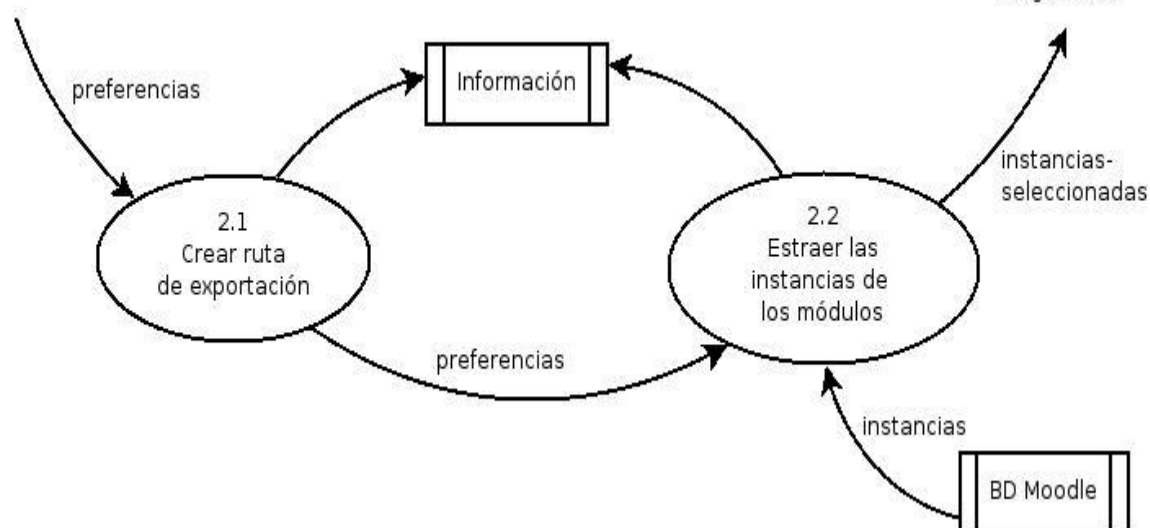


Figura. 3.10. Diagrama 2 (Gestionar Preferencias)

### 3.3.1.3 Generar componentes

A continuación se muestra (Figura 3.11) todo el flujo de información involucrado en el proceso de creación de los componentes necesarios, para construir el paquete SCORM.

Diagrama 3

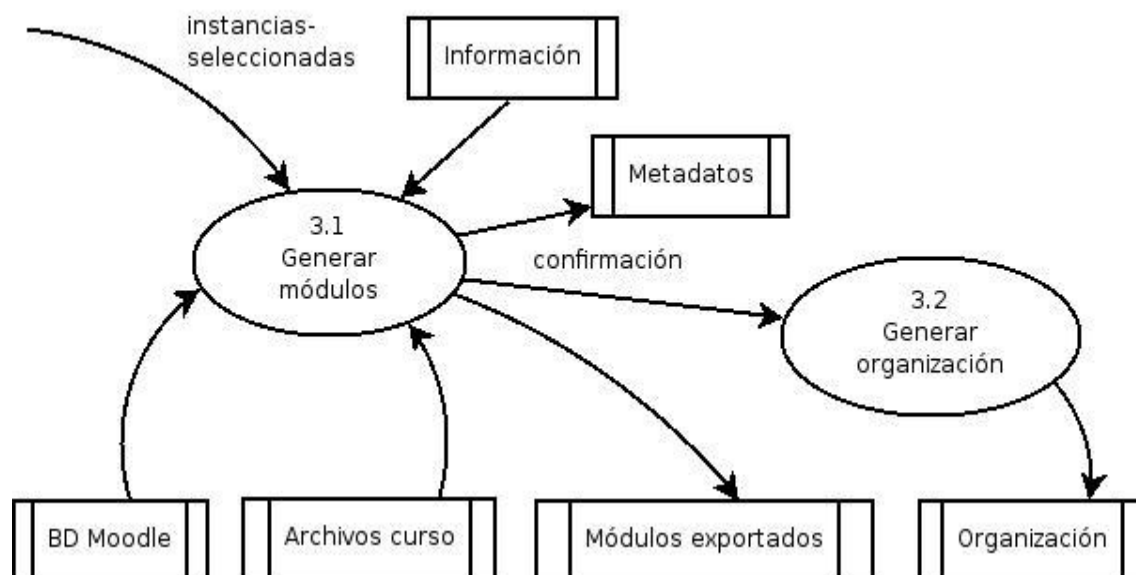


Figura. 3.11. Diagrama 3 (Generar componentes)

En la siguiente figura (3.12), se muestra el flujo de información requerido para la construcción de los módulos seleccionados.

Diagrama 3.1

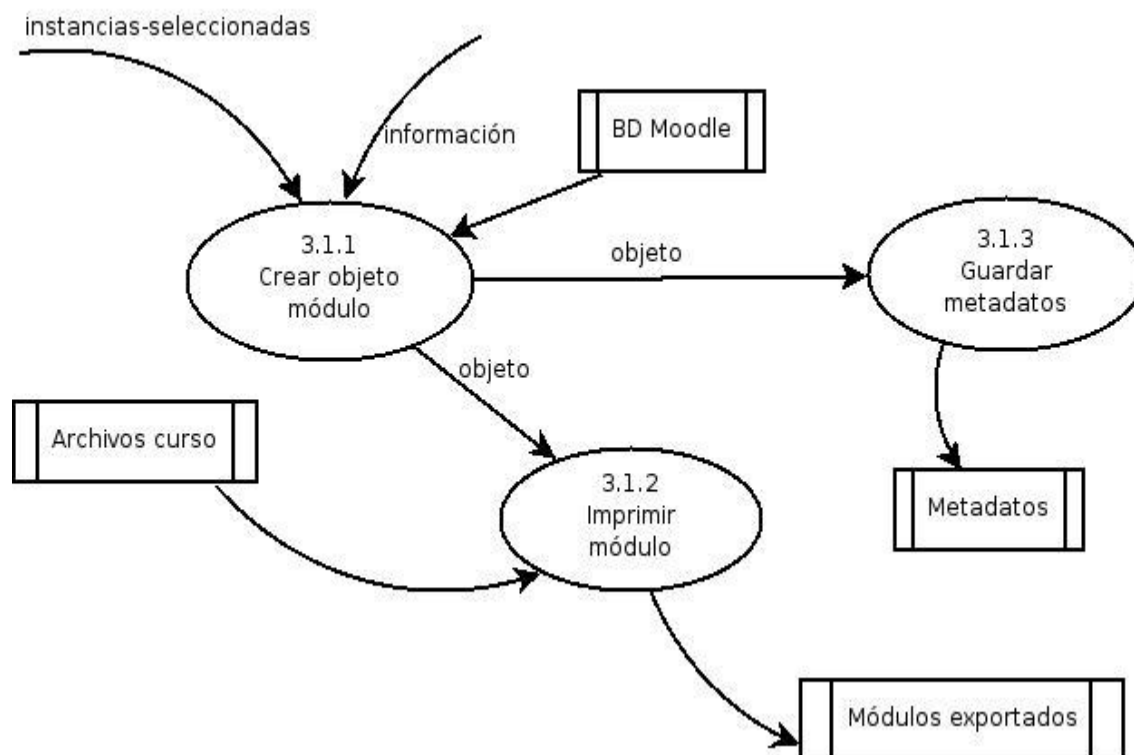


Figura. 3.12. Diagrama 3.1 (Generar módulos)

#### 3.3.1.4 Construir paquete SCORM

Antes de finalizar el proceso de construcción, se brinda la posibilidad de modificar los metadatos generados mostrando una vista previa de la estructura del paquete y de cada recurso como tal. Ver Figura 3.13.

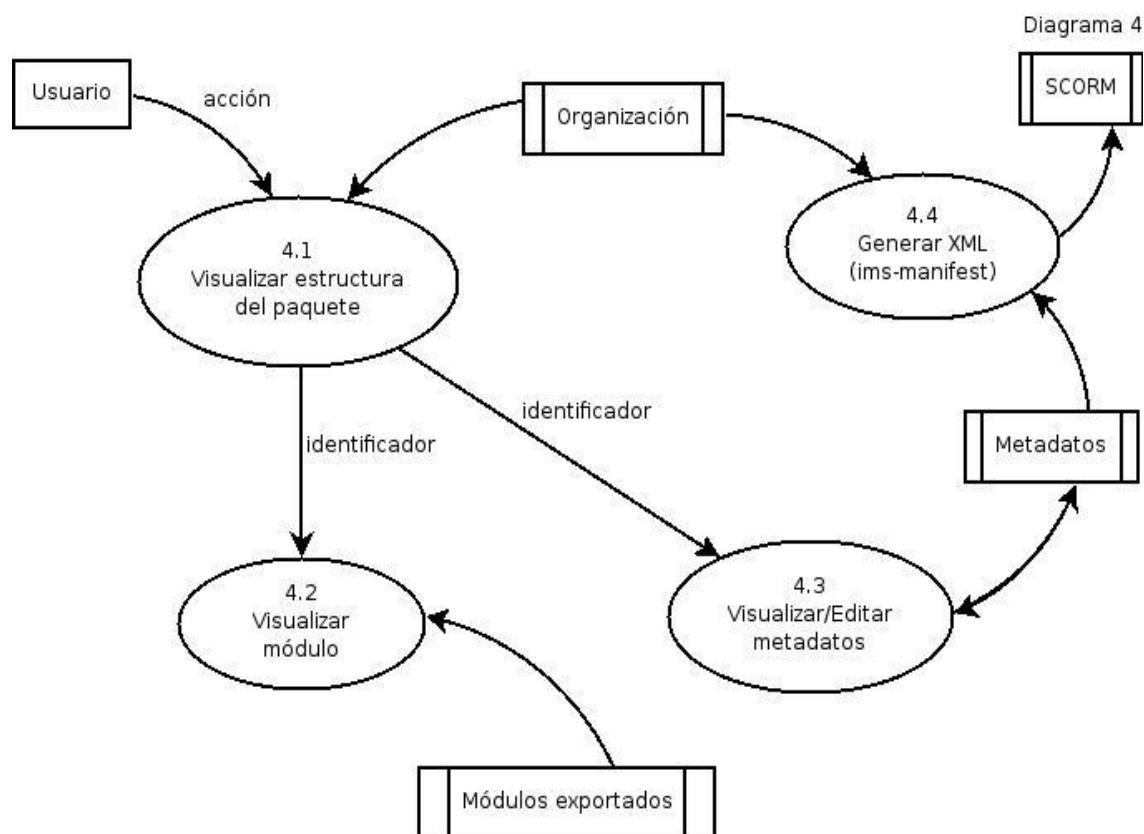


Figura. 3.13. Diagrama 4 (Construir paquete)

### 3.3.1.5 Diccionario de Datos

Con el objetivo de definir los términos utilizados en los diagramas de flujo de datos del bloque, se hace uso del siguiente diccionario de datos.

Entidad Externa: Usuario

Descripción: Cualquier usuario con el permiso “moodle/site:export”, necesario para realizar el proceso de exportación.

Estructura: [Estudiante| Administrador| Creadores de curso| Profesor |Otros]

Flujo de datos: solicitud

Descripción: Solicitar exportar un curso en forma de paquete SCORM.

Flujo de datos: notificación

Descripción: Respuesta del sistema ante diferentes eventos.

Estructura: [error al exportar| éxito en la exportación| cancelar el proceso]

Almacén: BD Moodle

Descripción: Base de Datos de Moodle.

Almacén: SCORM

Descripción: Paquetes SCORM generados por el sistema almacenados en los archivos del sitio.

Flujo de datos: sección

Descripción: Forma de agrupar los módulos dentro de un curso, generalmente representan un tema o semana.

Proceso: Gestionar solicitud

Descripción: Personalizar el proceso de exportación.

Flujo de datos: preferencias

Descripción: Datos que el usuario selecciona para personalizar el proceso de exportación

Estructura: módulos.

Proceso: Gestionar preferencias

Descripción: Crear la ruta de exportación y extraer de la BD Moodle todas las instancias de los módulos seleccionados por el usuario.

Flujo de datos: código

Descripción: Este valor se genera teniendo en cuenta el identificador de sección de usuario y el tiempo actual en microsegundos. Garantiza que cada proceso de exportación sea único.

Flujo de datos: camino-scorm

Descripción: Directorio raíz donde se exportará el paquete SCORM.

Flujo de datos: camino-metadatos

Descripción: Directorio donde se guardarán temporalmente los metadatos de cada módulo.

Almacén: Información

Estructura: {@código}1+ {camino-scorm}1 + {camino-metadatos}1

Flujo de datos: instancia

Descripción: Objeto con la información básica del módulo, extraída de la BD Moodle.

Estructura: identificador+ nombre+ descripción

Instancias= {instancia}

Flujo de datos: instancias-seleccionadas

Descripción: Instancias de los módulos seleccionados por el usuario para ser exportados.

Proceso: Generar módulos

Descripción: Construye los recursos que conforman el paquete con sus respectivos metadatos.

Proceso: Generar componentes

Descripción: Construye los recursos que conforman el paquete con sus respectivos metadatos, además de almacenar su organización.

Flujo de datos: metadato

Descripción: Información sobre los recursos generados.

Almacén: Metadatos

Estructura: {metadato}

Almacén: Organización

Descripción: Disposición original de los recursos en el curso de Moodle.



Proceso: Construir paquete

Descripción: Brinda al usuario la posibilidad de visualizar los recursos que contendrá el paquete, así como editar los metadatos generados automáticamente y culminar el proceso de exportación.

Proceso: Procesar solicitud

Descripción: Se encarga de verificar y solicitar en caso de ausencia los datos necesarios para comenzar el proceso.

Flujo de datos: identificador-curso

Descripción: Identificador del curso.

Proceso: Solicitar módulos a exportar

Descripción: Construir el listado de módulos disponibles, para que el usuario seleccione los que desea exportar. El listado puede ser ordenado por secciones (temas) o por tipo de módulo.

Proceso: Crear ruta de exportación

Descripción: Construye las rutas donde se crearán el paquete SCORM y los archivos temporales que contendrán los metadatos.

Proceso: Extraer las instancias de los módulos

Descripción: Extraer de la BD Moodle los objetos de los módulos seleccionados por el usuario.

Proceso: Construir módulos

Descripción: Construye un objeto con toda la información del módulo en cuestión y lo imprime.

Flujo de datos: módulo

Descripción: Actividad o recurso que constituye un contenido educativo.

Almacén: Archivos del curso

Descripción: Ficheros pertenecientes al curso.

Flujo de datos: módulo-válido

Descripción: Módulos que el sistema es capaz de exportar.

Almacén: Módulos-Válidos

Estructura: {módulo-válido}

Proceso: Crear objeto de módulo

Descripción: Construye un objeto con toda la información de la actividad o recurso que representa el módulo.

Flujo de datos: objeto

Descripción: Objeto con toda la información de la actividad o recurso que representa el módulo.

Proceso: Imprimir módulo

Descripción: Construye y almacena en formato HTML los módulos.

Flujo de datos: módulo-exportado

Descripción: Contenido de un módulo impreso.

Almacén: Módulos exportados

Estructura: {módulo-exportado}

Flujo de datos: acción

Descripción: Solicitud del usuario ante las posibilidades que brinda el sistema antes de concluir el proceso.

Estructura: [visualizar recursos| editar metadatos]

Flujo de datos: identificador

Descripción: Identificador del módulo exportado.

Flujo de datos: confirmación

Descripción: Informa al sistema la necesidad de incorporar un recurso a la organización del paquete.

Estructura: [SI | NO]

Proceso: Generar organización

Descripción: Incorpora un recurso a la organización del paquete.

Proceso: Guardar metadatos

Descripción: Genera metadatos por defecto para un recurso y los almacena en ficheros temporales.

Proceso: Visualizar módulo

Descripción: Muestra una vista previa de los recursos.

Proceso: Visualizar/Editar metadatos

Descripción: Muestra los metadatos del recurso y da la posibilidad de editarlos manualmente.

Proceso: Visualizar estructura del paquete

Descripción: Muestra en forma de árbol la organización de los recursos brindando la posibilidad de visualizarlos y/o editar sus metadatos.

Proceso: Generar XML

Descripción: Genera el imsmanifest y compacta el paquete.

### 3.3.1.6 Pseudocódigos más importantes

Representación en pseudocódigo de los principales procesos que forman parte de C2SCORM.

- Proceso: 1.1- **Procesar solicitud**  
COMIENZA

SI **identificador** EXISTE

DESPLEGAR (proceso 1.2) **Solicitar módulos a exportar**

OTRO

EXTRAER **cursos** de **BD Moodle**

MIENTRAS haya **curso** en **cursos**

MOSTRAR **curso**

FIN\_MIENTRAS

DESPLEGAR **identificador**

FIN\_SI

FINALIZA

- Proceso: 1.2- **Solicitar módulos a exportar**

COMIENZA

MIENTRAS haya **módulo** en **módulos-válidos**

EXTRAER **instancias** de **módulo** en **BD Moodle**

MIENTRAS haya **instancia** en **instancias**

MOSTRAR **instancia**

FIN\_MIENTRAS

FIN\_MIENTRAS

DESPLEGAR **preferencias**

FINALIZA

- Proceso: 2.1- **Crear ruta de exportación**

COMIENZA

SI no existe **código**

DESPLEGAR **notificación** "Missing data"

FIN\_SI

**camino-scorm** = ruta donde se exportan los paquetes + **código**

**camino-metadatos** = ruta temporal de los metadatos + **código**

TERMINA

- Proceso: 2.2- **Extraer instancias de módulos**

COMIENZA

MIENTRAS haya **módulo** en **módulos-válidos**

EXTRAER **instancias** de **modulo** en **BD Moodle**

MIENTRAS haya **instancia** en **instancias**

SI **instancia** en **preferencias**

**instancias-seleccionadas = instancias**

FIN\_SI

FIN\_MIENTRAS

FIN\_MIENTRAS

DESPLEGAR **instancias-seleccionadas**

FINALIZA

- Proceso: 3.1.1- **Crear objeto de módulo**

COMIENZA

**objeto** = “datos pasados por parámetro provenientes de la **instancia**”

MIENTRAS haya más información importante del **modulo** en **BD Moodle**

AGREGAR a **objeto** información importante del **modulo** en **BD Moodle**

FIN\_ MIENTRAS

DESPLEGAR **objeto**

FINALIZA

- Proceso 3.1.2 **Imprimir módulo**

COMIENZA

MIENTRAS haya **información** importante del **modulo** en el **objeto**

GUARDAR **información** del **objeto**

FIN\_ MIENTRAS

DESPLEGAR **módulo**

FINALIZA

- Proceso 3.1.3 **Guardar metadatos**

COMIENZA

GENERAR **metadato** por defecto

SERIALIZAR **información** de los **metadato**

GUARDAR **información** serializada en **Metadatos**

FINALIZA

- Proceso 4.1 **Visualizar estructura del paquete**

COMIENZA

CASO **acción**

CASO **acción** = “visualizar”

DESPLEGAR (proceso 4.2) **Visualizar módulo**

CASO **acción** = “editar”

DESPLEGAR (proceso 4.3) **Visualizar/Editar metadatos**

CASO **acción** = “continuar”

DESPLEGAR (proceso 4.4) **Generar XML**

OTRO

MIENTRAS haya **sección** en **Organización**

MIENTRAS haya **recurso** en **sección**

MOSTRAR **recurso** con opciones para editar y visualizar

FIN\_MIENTRAS

FIN\_MIENTRAS

FIN\_CASO

OFRECER opción de continuar

FINALIZA

- Proceso 4.4 **Generar XML**

COMIENZA

CREAR documento xml

ESCRIBIR cabeceras del imsmanifest

ESCRIBIR sección “metadata” del imsmanifest

ESCRIBIR secciones “organization” y “resources”

MIENTRAS haya **sección** en **Organización**

MIENTRAS exista **recurso** en **sección**

ADICIONAR **recurso** a la **sección** “organization”

ADICIONAR **recurso** a la **sección** “resources”

FIN\_MIENTRAS

FIN\_MIENTRAS

GUARDAR xml

COMPACTAR **paquete**

FINALIZA

### 3.3.2 Mapeo de Metadatos

Los metadatos de los recursos creados se generan dependiendo de la existencia de la información en Moodle., aunque el sistema brinda la posibilidad de editarlos.

A continuación se muestra un listado de los metadatos generados por la herramienta, especificando si pueden ser editados y una breve descripción de estos.

Categoría “General”:

- *identifier*

Metadato no editable.

Se obtiene usando el prefijo “IDE-” seguido de un identificador generado con la función *uniqueid*<sup>24</sup> de PHP, pasándole como prefijo el tipo de módulo y el identificador de la instancia en la base de datos. El resultado es convertido a mayúsculas por cuestiones de uniformidad.

- *title*

Metadato editable.

Nombre de la actividad o recurso en el curso en Moodle.

- *description*

Metadato editable.

---

<sup>24</sup> Genera una clave a partir del tiempo en milisegundos.

Descripción de la actividad o recurso en el curso en Moodle. La cadena es escapada para que no contenga caracteres HTML.

- *keyword*

Metadato editable.

Moodle no proporciona la información necesaria para el completamiento de este metadato, pero debido a que es obligatorio según ADL, se pone a disposición del usuario la posibilidad de su llenado.

#### Categoría “Life Cycle”:

- *version*

Metadato editable.

Se toma como consenso poner “1.0”.

- *status*

Metadato editable.

Usando como base el vocabulario propuesto por la IEEE LOM, y partiendo de que los contenidos provienen de un curso, se toma como valor predeterminado “Revised”.

#### Categoría “Meta-Metadata”

- *metadata scheme*

Metadato no editable.

Para generar los metadatos se utilizó la especificación LOM 1.0, por lo que este metadato tiene predefinido “LOMv1.0”.

#### Categoría “Technical”

- *format*

Metadato no editable.

Se obtiene por el MIME del recurso.

- *location*

Metadato no editable.

Se toma la URL del recurso relativa al imsmanifest.



### Categoría “Educational”

- learning resource type

Metadato editable.

Se llena dependiendo del recurso partiendo del vocabulario que propone la IEEE LOM.

- interactivity level

Metadato editable.

Se llena dependiendo del recurso a partir del vocabulario que propone la IEEE LOM.

### Categoría “Rights”

- cost

Metadato editable.

Se considera que los recursos no tienen costo alguno. Las opciones disponibles son las propuestas por la IEEE LOM.

- copyright and other restrictions

Metadato editable.

Se considera que los recursos son de libre acceso para todos, por lo que no están sujetos a restricciones de derecho de autor. Las opciones disponibles son las propuestas por la IEEE LOM.

### Conclusiones

En este capítulo se realizó un análisis de los procesos de exportación de los bloques desarrollados (C2Site y C2SCORM) apoyados en los DFD como herramienta fundamental. Además, se describieron las diferencias y mejoras fundamentales sobre las versiones anteriores, brindando una documentación detallada de los módulos propuestos.

## CONCLUSIONES

Después de realizar un análisis detallado de las funcionalidades de Moodle, cumpliéndose los objetivos planteados en este trabajo se concluye:

- Las herramientas desarrolladas facilitan la reutilización de los contenidos de Moodle.
- C2SCORM permite exportar los contenidos de Moodle 1.8.2 a un formato SCORM.
- C2Site permite exportar los contenidos de Moodle 1.8.2 en formato de sitio web estático.
- La documentación realizada en este trabajo, constituye una guía para futuros desarrollos sobre Moodle.

**RECOMENDACIONES**

Con el objetivo de perfeccionar y ampliar las funcionalidades de las herramientas desarrolladas (C2Site y C2SCORM), se proponen las siguientes recomendaciones:

- Lograr que en los cursos exportados con C2Site exista un mayor nivel de interactividad en las actividades.
- Extender las herramientas para que sean capaces de exportar otros módulos.
- Permitir la personalización de los diseños disponibles en C2Site.
- Incorporar otros metadatos de LOM a C2SCORM.
- Integrar los resultados a la comunidad de desarrolladores de Moodle.
- Continuar perfeccionando las herramientas desarrolladas a partir de los nuevos requisitos que puedan surgir como resultado de su explotación.

## BIBLIOGRAFÍA

- ADL SCORM. 2004.** Advanced Distributed Learning. [Online] ADL, 2004.  
<http://www.adlnet.gov/>. s.n.
- Angel, Miguel. 2000.** asenmac.com. *Informática y comunicaciones*. [Online] 2000. [Cited: Enero 8, 2008.] <http://www.asenmac.com/elearning.htm>. s.n.
- Apache. 2008.** The apache software foundation. [Online] 2008. <http://www.apache.org/>.
- Aplicaciones Empresariales. 2007.** Software y soluciones tecnológicas Open Source para las PYMES. [Online] 2007. <http://www.aplicacionesempresariales.com/general/dia-un-editor-de-diagramas-gratuito-y-compatible-con-visio.html>.
- CNICE. 2007.** Educación. *Estándares y especificaciones en e-learning*. [Online] 2007. [Cited: Enero 9, 2008.] <http://ares.cnice.mec.es/informes/16/contenido/3.htm>. s.n.
- Guzmán, Clara López. 2005.** *Los Repositorios de Objetos de Aprendizaje como soporte a un entorno e-Learning*. Salamanca : s.n., 2005. s.n.
- Internet 3D. 2007.** E-Learning and internet 3D. [Online] Octubre 30, 2007. [Cited: Enero 8, 2008.] <http://elearning3d.blogstop.com/.897>. s.n.
- Kruchten, Philippe. 2007.** Wikipedia, la enciclopedia libre. [Online] Wikimedia Foundation, Inc., 2007. [Cited: mayo 29, 2008.] [http://es.wikipedia.org/wiki/Arquitectura\\_de\\_software](http://es.wikipedia.org/wiki/Arquitectura_de_software).
- Laquna, Universidad de. 2004.** Software libre para Windows. [Online] Oficina de Software Libre, 2004. <http://windowslibre.osl.ull.es/web/ofimatica.html>.
- López, Vicente. 2006.** Video digital. *Programación PHP*. [Online] 2006. [Cited: Febrero 05, 2008.] <http://www.pabloschiavo.com.ar/v2/secciones.php?seccion=php>. s.n.
- Machado, Amira. 2007.** Ambientes virtuales en educación superior. *AVES*. [Online] 2007. [Cited: Enero 8, 2008.] <http://www.aves.edu.co>. s.n.
- Ministerio de Administraciones Públicas. 2007.** *Metodología MÉTRICA Versión 3*. Madrid : s.n., 2007. s.n.
- Moodle Docs. 2008.** Moodle Docs. [Online] 2008. <http://docs.moodle.org>.
- NosoloRed. 2007.** NosoloRed. [Online] 2007. [Cited: Abril 21, 2008.] <http://www.demonosolored.com/joomla/desarrollo/moodle-1.8.2-1.7.2-y-1.6.4-disponibles-para-instalar-en-nosolored-3.html>.
- Rebollo, Miguel. 2004.** Tesina de Master en Enseñanza y Aprendizaje Abiertos y a Distancia. *El estándar SCORM para EaD*. [Online] 2004. s.n.

- Red TTnet. 2005.** La formación sin distancia. *Servicio público de empleo estatal*. [Online] 2005. [Cited: Noviembre 15, 2007.]  
[http://www.inem.es/otras/TTnet/pdfs/LIBRO\\_laformacionsindistancia.pdf](http://www.inem.es/otras/TTnet/pdfs/LIBRO_laformacionsindistancia.pdf).
- Rodríguez, Gerardo Barquero. 2008.** Rincon del Vago. [Online] 2008. [Cited: Mayo 29, 2008.] <http://html.rincondelvago.com/analisis-de-sistemas.html>.
- Rouyet, Juan Ignacio and Martín, Victor. 2006.** Simposio pluridisciplinar sobre diseño, evaluación y descripción de contenidos educativos reutilizables. *SPDECE*. [Online] 2006. [Cited: Enero 15, 2008.] [http://spdece.uah.es/papers/Rouyet\\_Final.pdf](http://spdece.uah.es/papers/Rouyet_Final.pdf). s.n.
- Surós, Annia and Pernía, Reinier. 2006.** *Repositorio de Objetos de Aprendizaje*. Ciudad Habana : s.n., 2006. s.n.
- Wikimedia Foundation, Inc. 2008.** Wikimedia, la enciclopedia libre. [Online] Mayo 9, 2008. [Cited: Mayo 20, 2008.] [http://es.wikipedia.org/wiki/Diagrama\\_de\\_Flujo\\_de\\_Datos](http://es.wikipedia.org/wiki/Diagrama_de_Flujo_de_Datos).
- Wikipedia. 2007.** La enciclopedia libre. [Online] Comunidad, 2007. [Cited: Febrero 10, 2008.] <http://es.wikipedia.org/wiki/XML>. s.n.
- XTech. 2006.** Soluciones Linux para empresas. *Servidor web Apache - XTech*. [Online] Septiembre 14, 2006. [Cited: Febrero 2, 2008.]  
[http://wiki.xtech.com.ar/index.php/Servidor\\_web\\_Apache](http://wiki.xtech.com.ar/index.php/Servidor_web_Apache). s.n.

## ANEXOS

**Anexo 1.** Implementación de estándares en las plataformas más utilizadas.

<b>Plataforma</b>	<b>Descripción</b>
ATutor 1.4.2	Soporta la IMS Content Packaging Specification 1.1.3. El sistema proporciona soporte para estándares industriales abiertos para el intercambio de datos, incluyendo SCORM, para facilitar la interoperabilidad, reutilización de objetos y portabilidad de los contenidos.
BlackBoard 6	Soporta los siguientes estándares: SCORM 1.2, IMS Metadata 1.2.1, IMS Content Packaging 1.1.2 y Microsoft LRN 3.0. Incluye herramientas para facilitar la migración de cursos desde distintas versiones del software.
Bodington	Proporciona el formato IMS QTI.
Click2learn Aspen 2.0	El fabricante garantiza el cumplimiento de SCORM 1.2
eCollege AU+	El fabricante garantiza el cumplimiento de IMS metadata specification 1.1.
Educator	El fabricante garantiza el cumplimiento de IMS metadata specification 1.2.2 and IMS Content Packaging 1.1.3.
Learnwise	Soporta el cumplimiento de los siguientes estándares: IMS Content Packaging, IMS Metadata/IEEE LOM, IMS Enterprise, IMS LIP, IMS QTI 1.2, y SCORM 1.2 CMI.
Moodle	Soporta la utilización de objetos SCORM e IMS Content Packaging. Incluye herramientas que facilitan la migración de cursos entre versiones distintas.
Teknical Virtual Campus	El fabricante garantiza que cumple IMS Meta-data v1.2.1, IMS Content Packaging v1.1.2, IMS QTI lite v1.1, IMS QTI v1.1 y SCORM Versión 1.2 LMS-RTE2
The Learning Manager Enterprise Edition	El fabricante garantiza el cumplimiento de SCORM 1.2 (LMS-RTE3).
WebCT 4.1 Campus Edition	El fabricante garantiza el cumplimiento de: IMS Content Packaging 1.1.2, IMS QTI 1.1, IMS Enterprise 1.1, y Microsoft LRN 2.0. El fabricante puede trabajar con la institución para ayudarle a migrar los cursos existentes al sistema.

**Anexo 2.** Metadatos obligatorios en SCORM (ADL SCORM, 2004).

<b>Nombre</b>	<b>Agregación de contenido</b>	<b>Organización de contenido, actividad, SCO</b>	<b>Elemento básico</b>
1.0 General		X	X
1.1 Identificador		X	X
1.1.2 Entrada de catálogo		X	X
1.2 Descripción		X	X
1.3 Palabra clave		X	
2.0 Ciclo de vida		X	
2.1 Versión		X	
2.2 Estatus		X	
3.0 Metadatos		X	X
3.1 Identificador		X	X
3.1.2 Entrada de catálogo		X	X
3.3 Esquema de metadatos		X	X
4.0 Técnica		X	X
4.1 Formato		X	X
6.0 Derechos		X	X
6.1 Coste		X	X
6.2 Copyright y otras restricciones		X	X

**Anexo 3.** Multiplicidad de los elementos.

Multiplicidad	Explicación
1 y solo 1	El elemento debe existir 1 y solo 1 vez dentro del elemento padre.
0 o más	El elemento puede existir 0 o más veces dentro del elemento padre.
1 o más	El elemento debe existir 1 o más veces dentro del elemento padre.
2 o más	El elemento debe existir 2 o más veces dentro del elemento padre.
0 o 1	El elemento puede existir 0 o 1 vez dentro del elemento padre.



#### **Anexo 4. Tipos de Datos.**

IEEE LOM contiene varios tipos de datos, que son usados para describir los elementos de LOM.

##### **CharacterString**

Es un tipo de datos usado para capturar un grupo de caracteres que no son entendibles en un lenguaje humano.

Los caracteres representados por este tipo de datos pueden estar soportados por ISO/IEC 10646-1003:2000, que proporciona un estándar para la codificación de caracteres para la comunicación y el intercambio de información electrónica.

Puede tener como nodo hijo <vcard> y este elemento no es requerido en la versión 1.3 de SCORM.

##### **LangString**

Representa una o más cadenas de caracteres comprensibles en un lenguaje humano, este lenguaje también está definido en este elemento.

Se representa con la etiqueta <langstring>, que contiene la cadena de caracteres. Su SPM está determinado por el nodo padre y puede aparecer 0 o más veces dentro de este. El nodo tiene un atributo, xml:lang, que representa el lenguaje del contenido del nodo <lanstring>. Este atributo es del tipo de datos CharacterString con SPM de 100 caracteres y su uso es opcional. Se representa de la siguiente manera Código Lenguaje – Código País.

Código Lenguaje es un código de lenguajes definido en ISO 639:1988. Este valor es obligatorio.

Código País es un código de países definido en ISO 3166-1997. Este valor es opcional.

Este elemento cambia por <string> en la versión 1.3 de SCORM y el atributo por la etiqueta language.

## Vocabulary

Es una lista de valores, es representado como un una pareja fuente/valor. Esto indica que para cada elemento de este tipo existe una fuente (o propietario) y un valor (símbolo del vocabulario).

Hijos:

<source>

Indica la fuente, o propietario, de los valores del vocabulario. Para aquellos elementos que requieren el uso de los vocabularios de LOM, este elemento tendrá como valor "LOMv1.0", para los que no, puede representarse como un valor del tipo LangString con SPM de 1 000 caracteres.

<value>

Indica el valor definido por la fuente. Si el elemento <source> es "LOMv1.0" el valor tiene que estar en la lista definida por LOM. Para los que no sean de este tipo, el valor debe estar definido en la fuente, debe representarse como del tipo LangString y debe tener SPM de 1 000 caracteres.

El atributo xml:lang debe tener el valor "x-none", que indica que este es el lenguaje exacto para representar el elemento.

Estos elementos deben estar 1 y solo 1 vez dentro de los nodos padres.

En la versión SCORM 2004 estos elementos se representan de tipo CharacterString.

## DateTime

Se usa para describir un instante de tiempo. Está compuesto por dos elementos:

<dateTime>

Representación del tipo CharacterString del instante de tiempo con SPM de 200 caracteres.

Se representa de acuerdo con la norma ISO 8601:2000.

**<Description>**

Es una descripción del instante de tiempo. Este elemento es de tipo LangString y tiene SPM de 1 000 caracteres. Ocurren 0 o 1 vez dentro de los nodos padres.

**Duration**

Este tipo de datos es usado para describir un intervalo en el tiempo. Está compuesto por dos elementos:

**<duration>**

Representación del tipo CharacterString del intervalo de tiempo con SPM de 200 caracteres. Se representa de acuerdo con la norma ISO 8601:2000.

**<description>**

Es una descripción del intervalo de tiempo. Este elemento es de tipo LangString y tiene SPM de 1 000 caracteres. Ocurren 0 o 1 vez dentro de los nodos padres.

Anexo 5. Pantalla de un sitio generado con C2Site 1.0 que contiene varios cursos.

UCI Universidad de las Ciencias Informáticas

Formación a Distancia

Mis cursos

**Generales**  
Investigación de Operaciones  
Gestión de Software  
Formación Pedagógica

**Cuarto Año**  
Comercio Electrónico

Home

Cursos disponibles

**Generales**

»»» **Investigación de Operaciones**

La asignatura *Investigación de Operaciones* tiene un carácter eminentemente práctico, aunque se sustenta sobre una base teórica que permite, ante el planteamiento de un problema, que el estudiante sea capaz de identificar los datos necesarios, reconocer sus características, plantearlo, darle solución por un método adecuado y analizar los resultados obtenidos; estos problemas deben estar referidos a situaciones prácticas, preferiblemente vinculadas a la carrera.

»»» **Gestión de Software**

Curso para abordar los temas de **Gestión y Control de la Calidad del proceso de desarrollo del Software.**

»»» **Formación Pedagógica**

Este curso se desarrollará como asignatura del Plan de estudios para los estudiantes de 5to año, por la modalidad semi-presencial desde la Plataforma Moodle en el primer semestre, con 32 horas en total que incluye el autoaprendizaje mediante los materiales y actividades que comprende el curso y 3 encuentros presenciales definidos en el mismo.

Además, como asignatura en adelanto se ofertará a 4to año, en el segundo semestre, con iguales características.

**Cuarto Año**

»»» **Comercio Electrónico**

Escriba aquí la descripción del curso

Curso generado con Moodle | Contáctenos | Acerca de.. | Créditos

## Anexo 6. Contenidos a exportar (imagen C2Site 2.0.1).

Incluir:

---

Foros  Datos de Usuario  
 Foro  Datos de Usuario

Recursos  Datos de Usuario

<input checked="" type="checkbox"/> Dosificación del Curso	<input checked="" type="checkbox"/> Datos de Usuario
<input checked="" type="checkbox"/> Clases	<input checked="" type="checkbox"/> Datos de Usuario
<input checked="" type="checkbox"/> Libros	<input checked="" type="checkbox"/> Datos de Usuario
<input checked="" type="checkbox"/> rincón del CÓDIGO	<input checked="" type="checkbox"/> Datos de Usuario
<input checked="" type="checkbox"/> Clases	<input checked="" type="checkbox"/> Datos de Usuario
<input checked="" type="checkbox"/> Libros	<input checked="" type="checkbox"/> Datos de Usuario
<input checked="" type="checkbox"/> rincón del CÓDIGO	<input checked="" type="checkbox"/> Datos de Usuario
<input checked="" type="checkbox"/> Clases	<input checked="" type="checkbox"/> Datos de Usuario
<input checked="" type="checkbox"/> Libros	<input checked="" type="checkbox"/> Datos de Usuario
<input checked="" type="checkbox"/> rincón del CÓDIGO	<input checked="" type="checkbox"/> Datos de Usuario
<input checked="" type="checkbox"/> 1ra Tarea Extraclase	<input checked="" type="checkbox"/> Datos de Usuario
<input checked="" type="checkbox"/> 2da Tarea Extraclase	<input checked="" type="checkbox"/> Datos de Usuario
<input checked="" type="checkbox"/> Repositorio de Soluciones	<input checked="" type="checkbox"/> Datos de Usuario
<input checked="" type="checkbox"/> Guías de Ejercicios	<input checked="" type="checkbox"/> Datos de Usuario

Archivos del curso:

## Anexo 7. Opciones para formar el sitio web

### Configuraciones finales

#### Seleccione el diseño




---

#### Forma de exportar el curso ?

Formando un sitio Web con otros cursos  
 Un sitio con este curso solamente

---

Categoría:

**GLOSARIO**

1. *Metadatos*: Son datos altamente estructurados que describen información de la información o datos sobre los datos.
2. *Recursos de Moodle*: Elementos que permiten a los estudiantes acceder a los contenidos.
3. *Actividades de Moodle*: Son elementos que forman parte de los cursos montados en Moodle y permiten la interacción de los estudiantes con el entorno de forma evaluativa o constructiva.
4. *IMS Content Packaging*: Describe cómo empaquetar el contenido. De éste modo los LMS conocen cómo está estructurado el contenido y cómo se debe mostrar. Usa XML para describir cómo está realizado el contenido.
5. *CVS*: Aplicación informática que implementa un sistema de control de versiones, mantiene el registro de todo el trabajo y los cambios en los ficheros (código fuente principalmente) y permite que distintos desarrolladores colaboren. Difundido bajo la licencia GPL.
6. *Archivos del curso*: Permite tener un repositorio privado de ficheros para el curso.
7. *Archivos de Moodle*: Permite tener un repositorio de ficheros pertenecientes a la plataforma completa.
8. *Parsear*: Función que reconoce frases válidas de un lenguaje (en este caso XML) analizando la estructura sintáctica de un conjunto de elementos para utilizar la información interna.
9. *Contexto de ejecución*: El contexto de ejecución está determinado por el usuario o el inicio de sesión que está conectado a la sesión o que está ejecutando un módulo. Establece la identidad para la que se comprueban los permisos para ejecutar instrucciones o realizar acciones.
10. *Moodlers*: Refiere a desarrolladores para la plataforma Moodle.
11. *block\_base*: Clase padre de la que deben derivar todas las funciones utilizadas en el desarrollo de bloques.
12. *block\_list*: Clase padre de la que deben derivar las funciones utilizadas en el desarrollo de bloques de lista.

13. *BLOCK\_TYPE\_TEXT*: Este es uno de los dos valores válidos para el miembro variable `$this->content_type` de cada bloque.
14. *BLOCK\_TYPE\_LIST*: Este es uno de los dos valores válidos para el miembro variable `$this->content_type` de cada bloque.
15. *stdClass*: Clase genérica que permite estructurar instancias.
16. *AAAAMMDDHHmm*: Formato que representa la versión de un bloque teniendo en cuenta la fecha.
17. *phpDoc*: PHPDoc es una adaptación de Javadoc para PHP, un estándar formal para comentar el código PHP.
18. *ISO-12.207*: Es una norma que describe los procesos que pueden formar parte del ciclo de vida de un software, y presenta una relación de documentos y tareas a cumplir.
19. *GTK+*: Es un grupo de bibliotecas o rutinas para desarrollar interfaces gráficas de usuario (GUI). Es una dependencia para el funcionamiento de Dia en GNOME, entorno Linux.