

**Repositorio de Objetos de Aprendizaje para la
reutilización de contenidos en plataformas de
teleformación.**

**Trabajo de diploma para optar por el título de Ingeniería en
Informática**

Autores: Annia Surós Vicente (annias@uci.cu)

Reynier Pernia Rodríguez (rpernia@uci.cu)

Tutores: Ing. Daymy Tamayo Ávila (daymy@uci.cu)

MSc. David Leyva Leyva (davidl@uci.cu)

Ciudad de La Habana, Cuba
Junio, 2006

Resumen:

En la actualidad la Educación a Distancia y el uso de las Tecnologías de la Información y las Comunicaciones (TICs) han potenciado el desarrollo de la Enseñanza Virtual o e-learning. Su aplicación se ha convertido en un requisito casi indispensable en los centros de educación superior a nivel mundial.

Para lograr que los diversos recursos educativos, empleados en este tipo de enseñanza, sean asumidos por los diferentes sistemas de la misma forma, organizaciones internacionales se han dedicado a proponer estándares que garanticen la interoperabilidad, reusabilidad, durabilidad y accesibilidad. Para fomentar esto, los recursos, considerados Objetos de Aprendizaje (OAs), deben estar albergados en Repositorios.

El objetivo de este trabajo es desarrollar un Repositorio de Objetos de Aprendizaje empaquetados según el estándar SCORM (Modelo de Referencia para Objetos de Contenidos Intercambiables). Primeramente se hace un análisis de este estándar, para dar a conocer sus especificaciones y las principales organizaciones involucradas.

Se realiza un análisis de las herramientas y metodologías usadas para lograr su implementación, que se detalla junto al modelo de datos.

Por último se hace un estudio de la factibilidad del sistema, llegando a conclusiones sobre su efectividad.

Índice

Introducción	1
Capítulo 1. Fundamentos Teóricos	5
1.1 Elementos de e-learning.....	5
1.2 Objetos de Aprendizaje	7
1.3 Metadatos	8
1.4 Estándares y especificaciones para e-learning	8
1.5 ¿Por qué SCORM?	12
1.6 Modelo de Agregación de Contenidos (CAM)	12
1.6.1 Modelo de Contenidos.....	13
1.6.2 Modelo de Empaquetado	15
1.6.2.1 Componentes del Paquete de Contenidos	15
1.6.2.2 Componentes del Manifiesto	16
1.6.2.3 Construyendo Paquetes de Contenidos	16
1.6.2.4 Perfiles de aplicación del Paquete de Contenido SCORM	23
1.6.3 Metadatos.....	23
1.6.3.1 Perfil de Aplicación de los metadatos:	35
1.7 Repositorios de Objetos de Aprendizaje.....	39
1.8 Herramientas para el desarrollo de la aplicación.	43
1.9 Metodologías para el desarrollo de Sistemas Informáticos.....	47
Conclusiones del Capítulo	49
Capítulo 2. Descripción de la solución propuesta	51
2.1 Modelo de Dominio.....	51
2.1.1 Descripción del Problema de Dominio.....	52
2.1.2 Glosario de Términos	53
2.1.3 Diagrama de clases del dominio.	54
2.2 Requisitos	55
2.2.1 Funcionales	55
2.2.2 No Funcionales	55
2.3 Actores del Sistema	57
2.4 Diagrama de Casos de Uso del Sistema.....	57
2.5 Descripción de los Casos de Uso	58
2.6 Diseño.....	64
2.6.1 Diagrama de clases del diseño	64
2.6.2 Diseño de la Base de Datos (DER)	76
2.7 Tratamiento de errores	78
2.8 Implementación.....	79
2.8.1 Diagrama de Despliegue	79
Conclusiones del Capítulo	79
Capítulo 3. Análisis de factibilidad y Resultados	80
3.1 Estimación del esfuerzo.	80
3.2 Beneficios Tangibles e Intangibles.	85
3.3 Análisis Costo – Beneficio.....	85
3.4 Valoración de Sostenibilidad del Producto Informático.....	86
3.5 Resultados	86

Conclusiones del Capítulo	86
Conclusiones	88
Recomendaciones	89
Referencias Bibliográficas	90
Bibliografía	92
Anexos	95
Anexo 1. Multiplicidad de los elementos	95
Anexo 2. Tipos de Datos	96
Anexo 3. Plataformas que siguen los estándares principales.	99
Anexo 4. Requerimientos de los elementos del manifiesto según el Perfil de Aplicación del paquete de contenido de SCORM.	102
Anexo 5. Requerimientos de los elementos según el Perfil de Aplicación de los metadatos de SCORM.	104

Introducción

Las sociedades están en constante cambio. Para dar cumplimiento al encargo social que se le plantea a la escuela: lograr un individuo con determinados niveles de preparación requeridos por la sociedad, se hace cada vez más necesario la incorporación de nuevas técnicas en correspondencia con estas necesidades. La Educación a Distancia, proceso de formación en el que la distancia física separa a los estudiantes, los profesores y la tecnología, se abre paso entre estos modelos, asumida cada vez más en las universidades.

Diversas instituciones educacionales en todo el mundo, sobre todo de nivel superior, han comenzado a utilizar las Tecnologías de la Información y Comunicaciones (TICs) para enriquecer los procesos educativos actuales y ampliar su universo de posibilidades de formación al ofrecer nuevas formas de acceso al conocimiento, lo que ha propiciado el surgimiento de nuevos modelos y sistemas de formación.

Como nuevo concepto de la Educación a Distancia muy integrado al uso de las TICs, aparece el e-learning como una herramienta muy poderosa en la educación. El término e-learning procede del inglés, y se puede definir como el uso de las ventajas que ofrece un entorno Internet / Intranet para el proceso de enseñanza aprendizaje (Educación Virtual). En otras palabras, es la adaptación del ritmo de aprendizaje al alumno y la disponibilidad de las herramientas de aprendizaje independientemente de límites de horarios o geográficos.

Las herramientas que componen esta estrategia de educación son, por un lado, diferentes utilidades para la presentación de los contenidos (textos, animaciones, gráficos, vídeos) y por otro, herramientas de comunicación entre alumnos o entre alumnos y tutores de los cursos (correo electrónico, chat, foros). Una de las modalidades más usadas del e-learning son las plataformas de teleformación, en las cuales se integran muchas de las herramientas mencionadas anteriormente.

Las diversas comunidades vinculadas al e-learning han centrado su atención en señalar técnicas, métodos y recursos educativos de manera estandarizada para facilitar su

interoperabilidad, reusabilidad, durabilidad y accesibilidad. Ha sido preocupación de diversas instituciones la propuesta de estándares aceptados internacionalmente. Las organizaciones más importantes que han trabajado para lograr especificaciones y estándares son 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). A su vez, las diferentes herramientas tratan de asumir al máximo estas especificaciones.

Estas propuestas definen las características de los Objetos de Aprendizaje, de manera que cualquier sistema preparado para soportar este tipo de objeto pueda asumirlos, lográndose así cierto grado de interoperabilidad, reusabilidad, durabilidad y accesibilidad en estos sistemas. Sin embargo, esto no es suficiente, pues sería muy útil poder reutilizar los OAs, que cualquiera pueda contribuir en su construcción y hacer uso de ellos. Es necesario tener un lugar donde puedan estar almacenados, donde se pueda agregar este tipo de recursos para compartirlos. Es por esto que muchos autores plantean que no es posible pensar en Objetos de Aprendizaje si no se conciben albergados en repositorios, un lugar donde se encuentran almacenados los OAs y que brinde facilidades de búsqueda.

En el mundo existen varios Repositorios de Objetos de Aprendizaje, muchos de ellos brindan servicios gratuitos de inscripción y uso, pero su código no es abierto; otros lo son, pero no tienen en cuenta a los estándares. Algunos sólo soportan un tipo específico de OA, como multimedia; o son para un tipo de enseñanza o nivel educativo en particular. En resumen, no existe un repositorio de código abierto que implemente los requisitos definidos en los estándares.

Derivándose de la situación actual expresada, se ha planteado el **problema fundamental de esta investigación**: ¿Cómo favorecer la accesibilidad, reusabilidad, durabilidad e interoperabilidad de los Objetos de Aprendizaje basados en el estándar SCORM para plataformas e-learning?

Enmarcado en el proceso de gestión de Objetos de Aprendizaje como **objeto de estudio**, su **campo de acción** constituye el proceso de gestión de los Objetos de Aprendizaje basados en el estándar SCORM.

Concluyendo el análisis en la siguiente **hipótesis**: La implantación en la Universidad de las Ciencias Informáticas (UCI) de un Repositorio de Objetos de Aprendizaje, elaborados acorde a estándares internacionales, basado en tecnología Web, debe permitir la accesibilidad, reusabilidad, durabilidad e interoperabilidad de los mismos.

Se plantea como **objetivo general**: desarrollar un Repositorio de Objetos de Aprendizaje, de código abierto, que ofrezca mecanismos para su gestión.

Para dar cumplimiento al objetivo se han planteado las siguientes **tareas**:

- ✓ Realizar un estudio de las especificaciones y estándares desarrollados para e-learning y para los Objetos de Aprendizaje.
- ✓ Realizar un estudio de los Repositorios de Objetos de Aprendizaje existentes.
- ✓ Hacer el análisis y diseño del Repositorio de OAs.
- ✓ Implementación del Repositorio de OAs.
- ✓ Implantación del Repositorio de OAs.

Para la realización de estas tareas se utilizarán los diferentes **métodos de investigación**. Dentro de los Teóricos:

Análisis y síntesis:

Es de gran utilidad para realizar mentalmente la descomposición del problema en sus partes y cualidades, para luego haciendo uso de la síntesis establecer la unidad entre dichas partes e integrarlas como un todo, permitiendo descubrir relaciones y características generales entre los elementos de la realidad.

Hipotético-deductivo:

Se utilizan en la elaboración de la hipótesis y gracias al mismo, siguiendo la deducción, se llegará a la verificación de la hipótesis planteada o se reelaborará la misma en caso de que se llegue a una contradicción.

Histórico y lógico.

Se usa para estudiar la evolución de los fenómenos en su trayectoria y las leyes generales de su funcionamiento y desarrollo.

El presente trabajo consta de introducción, tres capítulos, conclusiones, recomendaciones, bibliografía y anexos:

En el Capítulo 1, **Fundamentos Teóricos**, se abordan de forma general los aspectos teóricos más importantes relacionados con el e-learning y sus tendencias actuales. Se hace una descripción del estándar SCORM, con el cual debe cumplir el sistema. Además se justifican el uso de las herramientas necesarias para dar cumplimiento a los objetivos trazados en esta investigación, así como de la metodología para el desarrollo de software utilizada.

En el Capítulo 2, **Descripción de la solución propuesta**, se describe el sistema a través de las fases que propone la metodología RUP para su desarrollo. Se hace una descripción de la estructura de la Base de Datos que se muestra a través de un Diagrama Entidad-Relación (DER).

En el Capítulo 3, **Análisis de Factibilidad**, se realiza un estudio de los esfuerzos requeridos para la construcción del sistema, y se valora la sostenibilidad del producto.

Capítulo 1. Fundamentos Teóricos

Este capítulo dará una visión general de algunos de los elementos relacionados con el e-learning, así como de la necesidad de la incorporación de estándares para favorecer su desarrollo, de manera que, se responderán interrogantes tales como ¿Qué es el e-learning? ¿Qué es un LCMS o plataforma de teleformación? ¿Qué es un objeto de aprendizaje? ¿Qué son los estándares o especificaciones? ¿Para qué necesitamos los estándares? Para ello se inicia dando elementos de e-learning como modalidad educativa. A continuación se introducen las nociones de Objetos de Aprendizaje y metadatos educativos, fundamentales para entender los conceptos con los que trabajan los estándares. Se definen los conceptos de estándar y especificación, estableciendo sus diferencias y similitudes. Se mencionan algunas de las iniciativas relacionadas con la propuesta de estándares educativos y las principales organizaciones involucradas en ello. Luego se da una descripción más detallada de aquellos aspectos del estándar SCORM que son de utilidad para el desarrollo del presente trabajo. Se dan a conocer otras iniciativas de construcción de repositorios de objetos de aprendizaje, de las cuales se hace un análisis crítico. Por último se explica el por qué de la selección de la metodología para el desarrollo de sistemas informáticos y las herramientas empleadas en el presente trabajo.

1.1 Elementos de e-learning

El concepto de *e-learning* se define de muchas formas diferentes fundamentalmente debido a que los actores que de él hacen uso son muy diversos, cada uno con su idiosincrasia y su ámbito de aplicación, aunque la acepción más común para e-learning es la enseñanza a través de un entorno Internet/Intranet. En español se utiliza el término teleformación. Puede ser definido como:

“aquella actividad que utiliza de manera integrada y pertinente computadores y redes de comunicación, en la formación de un ambiente propicio para la construcción de la experiencia de aprendizaje.” [15]

Esta actividad puede desarrollarse de forma sincrónica: compartiendo el profesor y el resto de los compañeros el mismo tiempo; o asincrónica, si las intervenciones de unos y otros quedan diferidas en el tiempo.

De esta definición, y muchas otras semejantes que se encuentran en la literatura, se aprecia que se trata de un caso muy particular de la Educación a Distancia (EaD), cuya única diferencia es la utilización de ordenadores y redes de telecomunicaciones como soporte para los contenidos y la comunicación.

En los sistemas de *e-learning*, se identifican tres elementos. En primer lugar, una plataforma que da soporte a todas las actividades formativas y de gestión que tienen lugar durante el aprendizaje. Se suele utilizar mucho sus siglas en inglés: LCMS (*Learning Content Management System*) o LMS (*Learning Management System*). En segundo lugar, los contenidos para el estudio o *courseware*. Y por último, herramientas de comunicación, tanto sincrónica como asincrónica, que permiten el contacto entre los participantes del curso.

El núcleo central de los sistemas de *e-learning* es la plataforma de formación o entorno virtual de enseñanza-aprendizaje. Básicamente, se trata de una aplicación (o un conjunto de aplicaciones) basadas en tecnologías Web cuyas funciones principales son:

- ✓ la gestión de los usuarios: inscripción, control de acceso, control de los aprendizajes e itinerarios formativos, generación de informes, etc.
- ✓ la gestión de los cursos, que incluye su distribución, el registro de la actividad de los usuarios (alumnos principalmente, pero también de los profesores), interacciones con el material formativo, tests, evaluaciones, tiempos de acceso, etc.
- ✓ gestión de los servicios de comunicación como apoyo del material didáctico y soporte para los aprendizajes. Las principales herramientas son el correo electrónico, foros de discusión, chats, etc. [15]

1.2 Objetos de Aprendizaje

El aporte que ha tenido mayor impacto en el *e-learning* en esta década es el concepto de *objeto de aprendizaje (OA) o learning object (LO)*. En la literatura podemos encontrar diversas formas de hacer referencia a este mismo término y la vaguedad presente en los distintos conceptos que de él se ofrecen en ocasiones tiende a provocar confusiones.

Así en el contexto del proyecto ARIADNE (2000) se ha utilizado el término “documentos pedagógicos” y en el MERLOT (Multimedia Educational Resource for Learning and On-Line Teaching) se utiliza “materiales de aprendizaje on-line”. También encontramos “objetos didácticos” [12]; “contenidos educativos reutilizables” [13]; o “material autosuficiente” [1]. Otros autores [2] prefieren el término “objeto de enseñanza”.

Algunas definiciones:

El Comité de Estandarización de Tecnologías Educativas (sigla en inglés LTSC Learning Technology Standards Committee) de IEEE (Institute of Electrical and Electronics Engineers) lo define como *cualquier entidad, digital o no, susceptible de ser usada en aprendizaje, educación o formación.* [7]

Cualquier recurso digital que puede reutilizarse para soportar el aprendizaje. [20]

Una entidad digital que puede utilizarse, reutilizarse y referenciarse durante el aprendizaje soportado por tecnología. [16]

Cualquier recurso digital, reproducible, y susceptible de referenciarse, empleado para llevar a cabo actividades de aprendizaje o de soporte, y disponible para utilizarse por otros. [9]

En este punto es conveniente aclarar que en el marco de esta investigación resulta relevante sólo aquel conjunto de OAs que se emplean en un entorno e-learning, es decir, que son recursos digitales.

De las definiciones ofrecidas el lector debe tener ya la idea de que objetos de aprendizaje son piezas individuales de contenido que sirven a fines instruccionales,

ejemplos de ellos son páginas Web, documentos, figuras o gráficas digitales, archivos de video y audio, etc.

Los objetos de aprendizaje pueden combinarse de infinitas formas para construir lecciones, cursos, módulos, etc. La selección de los objetos de interés puede realizarse previamente por el diseñador de los contenidos.

Otro punto común en las definiciones expuestas es la posibilidad que tienen los OAs de reutilizarse. Para favorecer esta característica tan deseable es necesario que estos recursos estén acompañados de cierta descripción, que puede lograrse mediante el empleo de metadatos.

1.3 Metadatos

El concepto de metadato no es novedoso, aunque es cierto que se ha popularizado mucho debido sobre todo a su aplicación en la Sociedad de la Información. Por metadato se entiende la “información sobre la información” o los “datos sobre los datos”. Un folleto publicitario sobre un libro con su índice, reseña bibliográfica del autor, precio, tipo de encuadernación, número de páginas, etc. es un ejemplo perfectamente claro de metadatos sobre el propio libro. En el mundo moderno, donde todo es digital, los metadatos también tienden a serlo. Si se tiene un documento pdf, puede lograrse su descripción anexándole un archivo de texto plano que recoja las principales características del mismo: tamaño, idioma, palabras clave, autor, etc.

Ya se ha planteado el uso de metadatos como una vía para describir objetos de aprendizaje, contribuyendo así a facilitar su intercambio y reutilización. Pero esto por sí solo no es suficiente, es necesario que sean conocidos e interpretados de la misma forma por los creadores de los mismos y por los usuarios finales. Esto solo es posible mediante el empleo de estándares y/o especificaciones.

1.4 Estándares y especificaciones para e-learning

Se define un estándar [10] como *“un documento, establecido por consenso y aprobado por un cuerpo reconocido, que proporciona, para el uso común y repetido, reglas, guías o*

características para actividades o sus resultados, dirigidas a alcanzar el grado de orden óptimo en un contexto dado."

Frecuentemente los términos estándar y especificación se utilizan indistintamente, no obstante, es importante puntualizar su diferencia. Si una tecnología, formato o método ha sido ratificado por algún organismo oficial de estandarización, se trata de un estándar. Pero si una tecnología, formato o método propuesto no ha sido aprobado por algún organismo oficial de estandarización, se trata de una especificación. Aunque, en algunos casos, una especificación puede considerarse un estándar *de facto* si su uso es extendido y entretanto se ratifica como estándar.

Durante la pasada década surgieron varias iniciativas de organismos reconocidos internacionalmente encaminadas a la propuesta de estándares para los distintos componentes que forman la arquitectura de las plataformas de formación: alumnos, contenidos, pruebas, estrategias docentes, etc. Entre todas ellas se destacan cuatro: AICC (Aviation Industry CBT Committee), IEEE, IMS (Instruction Management Systems) y ADL (Advanced Distributed Learning).

IEEE-LTSC

Su aporte más significativo es el estándar LOM (Learning Objects Metadata), que define elementos para describir los objetos de aprendizaje. El objetivo de LOM es servir de guía en el mercado de recursos educativos para con ello potenciar su búsqueda, evaluación, obtención y utilización. Define un esquema de nueve categorías para anotar los atributos de un recurso de aprendizaje, entre los que se encuentran aspectos técnicos, educativos, de derechos de autor, clasificación, o de ciclo de vida. Dentro de cada categoría otras subcategorías se definen. Además, con este estándar es posible desarrollar catálogos que consideren diferencias culturales y lingüísticas. Es importante destacar el hecho de que este estándar no especifica cómo implementar el esquema de los metadatos de un OA sino que su alcance se limita a servir como referencia a otros estándares que se ocupen de esta tarea.

IMS

Este consorcio pone en práctica las recomendaciones de AICC y de IEEE utilizando XML (eXtensible Markup Language) para describir aspectos claves de cursos, lecciones, asignaturas, alumnos y grupos. IMS elabora las siguientes especificaciones:

1. Especificaciones usadas para describir, descubrir e intercambiar contenidos.

Learning Object Metadata (LOM): Esta especificación entrega una guía sobre cómo los contenidos deben ser identificados y sobre cómo se debe organizar la información de los alumnos de manera que se puedan intercambiar entre los distintos servicios involucrados en una plataforma. La especificación para metadatos del IMS consta de tres documentos: *IMS Learning Resource Metadata Information Model*, *IMS Learning Resource XML Binding Specifications*, *IMS Learning Resource Meta-data Best Practices and Implementation Guide*.

Content Packaging (CP): Esta especificación 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.

Question and Test Interoperability (QTI): El IMS QTI propone una estructura de datos XML para codificar preguntas y pruebas, así como su correspondiente reporte de resultados. El objetivo de esta especificación es permitir el intercambio de estas pruebas y datos de evaluación entre distintas plataformas.

Digital Repositories: El IMS está en el proceso de creación de especificaciones y recomendaciones para la interoperación entre repositorios digitales.

2. Especificaciones para la interacción con el contenido y el seguimiento.

Simple Sequencing (SS): Esta especificación define reglas que describen el flujo de instrucciones a través del contenido según el resultado de las interacciones de un alumno con el contenido.

Learning Design (LD): Este grupo de trabajo del IMS investiga sobre las maneras de describir y codificar las metodologías de aprendizaje incorporadas en una solución *e-learning*.

3. Especificaciones para la interoperabilidad de las aplicaciones

Learner Information Packaging (LIP): Esta especificación define estructuras XML para el intercambio de información de los alumnos entre sistemas de gestión de aprendizaje, sistemas de recursos humanos, sistemas de gestión del conocimiento, y cualquier otro sistema utilizado en el proceso de aprendizaje.

ADL

La iniciativa ADL (*Advanced Distributed Learning*) es un programa del Departamento de Defensa de los Estados Unidos que se creó en 1997 con el objetivo de garantizar el acceso a la educación y a materiales de calidad ajustables a las necesidades individuales, así como facilitar su disponibilidad.

No es un organismo que se dedique a definir sus propios estándares, sino que recoge e integra las propuestas de otros grupos, en concreto de los tres anteriores: AICC, IEEE e IMS. Su propuesta es la denominada SCORM (*Shareable Content Object Reference Model*), que podría traducirse como Modelo de Referencia para Objetos de Contenidos Intercambiables.

SCORM está formado por un conjunto de tres libros técnicos que recogen los diversos aspectos de la especificación, más un cuarto libro que ofrece una visión general sobre el estándar. Los libros son:

Overview: Introducción a los elementos de SCORM y su terminología.

Content Aggregation Model (CAM): Información sobre la confección, etiquetado y empaquetamiento de los contenidos de aprendizaje.

Run-Time Environment (RTE): Requerimientos para la ejecución de los contenidos, comunicación, seguimiento, transferencia de datos y gestión de errores.

Sequence Information & Navigation (SN): Descripción de las reglas que marcan los secuencia de navegación entre distintos objetos de aprendizaje.

1.5 ¿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. [15]

Con el término empaquetamiento se hace referencia a la recolección y a la descripción de los elementos de un curso. Entre las iniciativas más relevantes en la actualidad se encuentra la especificación para el empaquetamiento de contenidos CPS de IMS, que está recogida y ampliada en el documento de ADL SCORM. [15]

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 (ver Anexo 3). [15] Moodle, plataforma de teleformación que se implanta en la UCI, soporta objetos SCORM.

Por estas razones se decide tomar a SCORM como el estándar en el que deben estar empaquetados los recursos que se almacenarán en el repositorio. Por tanto se hace un estudio del Modelo de Agregación de Contenidos (CAM) de SCORM.

1.6 Modelo de Agregación de Contenidos (CAM)

El Modelo de Agregación de Contenidos (CAM) proporciona una manera consistente para describir la estructura de dichos contenidos, el contenido de aprendizaje, la metainformación sobre los distintos componentes y su estructura y una serie de reglas que determinan los recorridos permitidos sobre los contenidos. Lo anterior facilita la

búsqueda, y por tanto favorece la reutilización, de los paquetes de contenido y sus recursos. Además garantiza la construcción de contenidos que se comportarán de manera similar independientemente del sistema –que brinde soporte al estándar– donde se utilicen o de la herramienta de autor empleada para su creación.

Representa una taxonomía neutra en lo referente a un modelo pedagógico que sirva como base al estándar. En principio, este hecho debería facilitar la utilización de esta especificación en cualquier contexto educativo.

El CAM da soporte al proceso de agregación y organización de recursos de aprendizaje para formar componentes más complejos que puede asociarse a actividades de aprendizaje para su realización por parte de los estudiantes.

Las reglas que establecen este comportamiento se dividen en cuatro partes:

- ✓ el **Modelo de Contenidos** proporciona una terminología común para ser usada en todo el CAM.
- ✓ el **Modelo de Empaquetado** ofrece la descripción y los requerimientos para componer objetos de aprendizaje y formar unidades más complejas.
- ✓ **Metadatos** para la descripción de los componentes del modelo de contenidos.
- ✓ normas de **Secuenciación y Presentación** que luego serán utilizadas en las normas del libro **SN**.

1.6.1 Modelo de Contenidos

El Modelo de Contenidos de SCORM describe los componentes que define esta inciativa para construir recursos de aprendizaje, y cómo estos recursos pueden combinarse en unidades de más alto nivel para formar unidades de instrucción.

La forma más básica es un Recurso (*Asset*). Los Recursos son representación electrónica de de textos, imágenes, sonidos, objetos de evaluación o cualquier otra entidad que pueda entregarse a través de un navegador. Un recurso puede combinarse

con otros para crear nuevos recursos. Se describen utilizando metadatos que permiten su búsqueda en repositorios y su reutilización.

Un SCO (Objeto de Contenido Intercambiable –*Shareable Content Object*–) es una colección de uno o más Recursos (*Assets*) que representan un recurso de aprendizaje capaz de comunicarse y de ser lanzado por una plataforma de formación. Es la unidad más pequeña que la plataforma puede manejar.

La única diferencia entre SCO y un Recurso es la capacidad del SCO para comunicarse con la plataforma. Para ello, emplea el *ECMAScript API for Content to Runtime Services Communication* de IEEE. A través de este API el SCO es capaz de localizar el punto de entrada a la plataforma, iniciar la comunicación con ella, operar sobre la plataforma leyendo y escribiendo datos en ella y terminar la comunicación cuando ya no sea necesaria. De todos estos métodos, los únicos obligatorios para todos los SCO son los que permiten iniciar (*Initialize("")*) y terminar (*Terminate("")*) la comunicación. El resto dependerá de la naturaleza del contenido.

Para mejorar la reutilización de los SCO, estos deben ser todo lo independiente posible del contexto de aprendizaje. Si mantiene relaciones muy estrechas con otros objetos, será muy difícil poder aprovecharlo en otros entornos. Además, de esta manera puede integrarse fácilmente en unidades de instrucción de más alto nivel (actividades).

Un SCO se puede describir mediante el empleo de metadatos, lo que permite su búsqueda y descubrimiento dentro de los repositorios, de tal modo se incrementan sus oportunidades de reutilización.

Una Organización de Contenidos es un mapa que representa el uso de los distintos contenidos (Recursos o SCO) a través de un conjunto de unidades de instrucción denominadas Actividades. Estas actividades pueden estar a su vez formadas por más actividades, sin que exista límite en el nivel de anidamiento ni ninguna relación preestablecida con alguna taxonomía (cursos, capítulos, lecciones, etc.). Tanto las Organizaciones de Contenidos como las Actividades pueden tener asociados los metadatos que permiten reutilizarlas y localizarlas en repositorios.

1.6.2 Modelo de Empaquetado

Una vez que el contenido de aprendizaje es diseñado y construido, es necesario ponerlo a disposición de los alumnos, herramientas de autor, repositorios o plataformas de formación. Para ello, el modelo de empaquetado de SCORM utiliza de forma estricta la especificación IMS Content Packaging y añade requerimientos y orientaciones para el empaquetamiento de recursos (*Assets* y *SCOs*) y organizaciones de contenidos.

Esta sección ofrece una descripción de este modelo y para ello se organiza como sigue:

Componentes del Paquete de Contenidos: Se ofrece una descripción del paquete de contenidos y de sus componentes.

Componentes del Manifiesto: Define la composición del manifiesto.

Construyendo Paquetes de Contenidos: Se centra en la creación de un paquete de contenidos y del archivo manifiesto. Describe los componentes XML del manifiesto.

Perfiles de aplicación del Paquete de Contenido SCORM: Describe los dos perfiles de aplicación y los requerimientos asociados con ellos.

1.6.2.1 Componentes del Paquete de Contenidos

Un Paquete de Contenidos está formado por dos componentes: un documento en XML que describe la estructura del contenido y los recursos, llamado manifiesto (*imsmanifest.xml*), y los ficheros físicos o URL con el contenido real del paquete. Representa una unidad de aprendizaje que tiene relevancia instruccional y puede repartirse independientemente. Queda a criterio del diseñador decidir qué es una unidad relevante.

El manifiesto y todos los ficheros de contenidos se agrupan en un único archivo comprimido, que en SCORM se denomina PIF (*Package Interchange File*). 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). Los ficheros con este formato son los que se intercambian entre plataformas de formación.

1.6.2.2 Componentes del Manifiesto

El manifiesto contiene la información necesaria para describir el contenido del paquete. Está formado por cuatro secciones:

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

Organizaciones: Representa las posibles formas de Organización de los Contenidos y su descomposición en actividades. Las actividades se enlazan con los recursos que utilizan, que se encuentran en la siguiente sección, a través del identificador de estos últimos. Este elemento puede incorporar también las instrucciones de secuenciación y navegación. En caso de que existan varias formas de organización debe especificarse cuál de ellas se tomará por defecto.

Recursos: Describe los recursos externos y locales que utiliza el paquete, a través de URL. Los recursos locales se encontrarán comprimidos en el mismo PIF. 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: Como regla general el paquete siempre contiene un manifiesto de nivel superior que puede contener uno o más submanifiestos. El manifiesto de nivel superior describe el paquete y cualquier submanifiesto anidado describe el contenido al nivel en el cual se encuentra, como por ejemplo un curso. En este momento (febrero 13, 2006) ADL recomienda que no se usen submanifiestos.

1.6.2.3 Construyendo Paquetes de Contenidos

Esta sección define los requerimientos para la construcción del manifiesto. El archivo conocido como manifiesto tiene que llamarse *imsmanifest.xml*. Como lo indica su nombre es un archivo XML y en esta sección se describen algunos de los elementos que lo componen así como los requerimientos definidos para cada uno de ellos por la

Especificación de Empaquetamiento de Contenidos de IMS. Este documento XML Debe cumplir con la especificación XML 1.0 de W3C.

Algunos elementos usan el término SPM (el máximo permitido más pequeño) para describir la multiplicidad de los elementos. El SPM indica a las aplicaciones que procesan paquetes de contenido que tienen que procesar al menos ese número de elementos o caracteres, pero son libres de excederlo.

A continuación se expresan una serie de detalles técnicos que son relevantes para el desarrollo del presente trabajo a la hora de tratar los paquetes de contenido.

Existen herramientas de autor (ejemplo: Reload Editor) que permiten la creación de paquetes de contenido con abstracción total de los detalles técnicos recogidos en las especificaciones que estandarizan su formato.

Elemento <manifest>

Este es el elemento 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 echas en este elemento. Aunque esto no es un requerimiento, según las especificaciones de XML, ADL lo recomienda como una buena práctica.

Atributos:

identifier (obligatorio): Este atributo identifica el manifiesto y es único dentro de él. Típicamente es proporcionado por el autor o la herramienta de autor empleada durante el desarrollo del paquete.

version (opcional): Identifica la versión del manifiesto. Suele usarse para distinguir entre manifiestos con el mismo identificador (*identifier*). Tiene un valor SPM de 20 caracteres.

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

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>

Se usa para describir 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. Se requiere que el contenido de este elemento sea **CAM 1.3** indicando que el paquete de contenido se construye de acuerdo con el Modelo de Agregación de Contenidos de SCORM versión 1.3.

Elemento <organizations>

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

Atributos:

default (obligatorio – si el paquete está previsto para ser entregado a un usuario final, como por ejemplo un estudiante –): 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 que organización utilizar en dependencia de a quien vaya dirigido el contenido. Tiene un SPM de 200 caracteres.

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 de anidamiento. Esta estructuración 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.

identifieref (opcional): referencia el identificador de un recurso (*identifier*) en la sección recursos (*resources*) o en un submanifiesto. Tiene un valor SPM de 2000 caracteres.

Hijos:

<title>

<item>

<metadata>

Elemento <title>

Describe el título del elemento <item>. Tiene un valor SPM de 200 caracteres.

Elemento <resources>

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

Atributos:

xml:base (opcional): 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. Típicamente proporcionado por un autor o herramienta de autor.

type (obligatorio): Indica el tipo del recurso. Este valor tiene un SPM de 1000 caracteres. 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. Tiene un SPM de 2000 caracteres, en esta longitud se incluyen las de los valores de los `xml:base` aplicados a él. Deja de ser opcional para convertirse en obligatorio cuando un elemento <item> referencia al recurso <resource>.

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

adlcp:scormType (obligatorio): Define el tipo del recurso SCORM. Es una extensión de ADL al Modelo de Información de Empaquetamiento de Contenidos de IMS. Su valor tiene que ser **asset** o **sco**.

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. Su valor se afecta por el uso de los valores de xml:base. Tiene un SPM de 2000 caracteres, incluyendo en esta longitud las de los valores de los xml:base aplicados a él.

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:

identfierref (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. Este valor tiene un SPM de 2000 caracteres.

1.6.2.4 Perfiles de aplicación del Paquete de Contenido SCORM

Existen actualmente dos perfiles de aplicación del paquete de contenido SCORM:

Paquete de recursos: define un mecanismo para empaquetar recursos sin especificar ninguna organización para los mismos.

Paquete de agregación de contenidos: su propósito principal es la entrega de contenido al usuario final (ejemplo: a través de una plataforma de formación) Debe utilizarse para empaquetar recursos de aprendizaje y su estructura (ejemplo cursos, módulos, lecciones, etc.).

Para cada Perfil de aplicación del Paquete de Contenido, SCORM define requerimientos relacionados con los elementos/atributos descritos anteriormente en la sección *Construyendo Paquetes de Contenidos*. (ver Anexo 4.)

1.6.3 Metadatos

Una vez que los componentes han sido creados es útil describirlos mediante el empleo de metadatos, lo que facilita su búsqueda en plataformas de teleformación y repositorios.

El Perfil de Aplicación de Metadatos de SCORM se basa directamente en los estándares IEEE 1484.12.1-2002 LOM e IEEE 1484.12.3 que especifica una vía para implementar LOM usando XML.

SCORM recomienda el uso del IEEE LOM para describir los Componentes del Modelo de Contenido de SCORM. Sin embargo, otros esquemas de metadatos pueden ser utilizados. Estos esquemas de metadatos pueden o no ser reconocidos por los diferentes sistemas.

Metadatos de SCORM

Según IEEE, cada elemento de los metadatos de LOM es opcional, lo que implica que el desarrollador opcionalmente puede escoger cuales elementos usar. SCORM establece

requisitos adicionales, que indican cuales elementos son de carácter obligatorio (ver Anexo. 5).

El Modelo de Información de LOM está compuesto por nueve categorías: }

La categoría *General* agrupa información general del Componente del Modelo de Contenido como un todo.

La categoría *Ciclo de Vida* contiene características relacionadas con la historia y el estado actual del Componente del Modelo de Contenido y qué lo ha afectado en su evolución.

La categoría *Meta-metadato* se usa para describir información sobre los metadatos.

La categoría *Técnica* se usa para describir requerimientos y características técnicas de los Componentes del Modelo de Contenido.

La categoría *Educativa* describe características educacionales y pedagógicas de los Componentes del Modelo de Contenido.

La categoría *Derechos* contiene los derechos de propiedad intelectual y condiciones de uso del Componente del Modelo de Contenido.

La categoría *Relación* agrupa características que definen relaciones entre el Componente del Modelo de Contenido y otros componentes.

La categoría *Comentario* se usa para proveer comentarios del uso educacional del Componente del Modelo de Contenido e información sobre cuándo y por quiénes fueron creados los comentarios.

La categoría *Clasificación* se usa para describir dónde se encuentra el Componente del Modelo de Contenido, de acuerdo a un sistema de clasificación particular.

Algunos elementos usan el término SPM (el máximo permitido más pequeño) para describir la multiplicidad de los elementos. 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.

Elemento <lom>

Todas las instancias de metadatos tienen a <lom> como el elemento raíz. En un paquete de contenido, todos los metadatos se encuentran dentro del elemento <metadata> encontrado en el fichero imsmanifest.xml. El nodo raíz encapsula todas las categorías descritas anteriormente. No hay un orden para las nueve categorías, los elementos hijos pueden aparecer en cualquier orden.

Todas las declaraciones de espacio de nombres deben estar dentro del elemento <lom>. Esto incluye cualquier espacio de nombre que sea considerado ampliación de los metadatos. Sin embargo, esto no es considerado un requerimiento teniendo en cuenta las especificaciones de XML, ADL considera que esto es una “buena práctica” y estimula que se provea esa información. Este elemento al igual que sus nodos hijos contiene como espacio de nombre: <http://ltsc.ieee.org/xsd/LOM> y como prefijo de espacio de nombre: lom.

El elemento <lom> contiene importantes elementos que SCORM requiere para describir todos los Componentes del Modelo de Contenido. Este elemento debe aparecer sólo una vez dentro de su nodo padre.

Hijos:

<general>

<lifeCycle>

<metametadata>

<technical>

<educational>

<rights>

<relation>

<annotation>

<clasification>

A continuación se hace una descripción sólo de los metadatos que son de interés para el desarrollo de este trabajo. La multiplicidad de los elementos se puede encontrar en el Anexo 1 y el tipo de datos en el Anexo 2.

Elemento <general>

La categoría General agrupa la información general que describe al recurso como un todo. El recurso a describir en este caso son los componentes del Modelo de Contenido de SCORM (Recurso, SCO, Actividad, una Organización de Contenidos o el paquete como un todo). Esta información general es importante para describir al componente.

Es un elemento padre y debe aparecer solo una vez dentro de su nodo padre.

Hijos:

<title>

<description>

<keyword>

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>

Agrupar características relacionadas a la historia y al estado actual del Componente del Modelo de Contenido y qué lo afectó en su evolución. Incluye el estado del componente, versión y una lista de organizaciones que han afectado su estado de una forma u otra.

Es un elemento padre y debe aparecer 1 sola vez dentro de su nodo padre.

Hijo:

<version>

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>

En la versión 1.3 de SCORM este elemento cambia por <metaMetadata>.

Contiene elementos que describen los metadatos del componente y no el componente en cuestión. Esta categoría describe cómo los metadatos pueden ser identificados, quien creó la instancia, cómo, cuándo y con cuáles referencias.

Es un elemento padre y debe aparecer solo 1 vez dentro de su nodo padre.

Hijos:

<language>

<metadatascheme>

Elemento <language>

Representa el lenguaje de las instancias de metadatos. Es el valor por defecto de las instancias de metadatos de tipo LangString.

Multiplicidad	SPM	Tipo de Dato	SPM
0 o 1	-	CharacterString	100

Se representa de la siguiente manera CodLenguaje-CodPaís.

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

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

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.

SCORM requiere el siguiente esquema de metadatos:

LOMv1.0 Indica que se usó la versión 1.0 del Esquema Base de LOM.

Si otros esquemas de metadatos son relevantes, pueden aparecer. Sin embargo, esto no es relevante para SCORM 2004.

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

Para la versión SCORM 2004 este elemento cambia por <metadataSchema>. Su multiplicidad cambia para 2 o más, siendo requerido por SCORM en esta versión también este esquema de metadatos:

ADLv1.0 Indica que se usó el Perfil de Aplicación de Metadatos de SCORM.

Elemento <technical>

Describe las características y requerimientos técnicos del Componente del Modelo de Contenido.

Es un elemento padre y debe aparecer solo 1 vez dentro de su nodo padre.

Hijos:

<format>

<size>

<location>

<duration>

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 <size>

Representa el tamaño del Componente del Modelo de Contenido en bytes. Si el componente está compactado, este elemento debe hacer referencia al tamaño del componente descompactado.

Multiplicidad	SPM	Tipo de Dato	SPM
0 o 1	-	CharacterString	30

Elemento <location>

Especifica la ubicación del Componente del Modelo de Contenido descrito por los metadatos. El uso de este elemento no es recomendado por SCORM, ya que la ubicación del componente se encuentra disponible en los elementos <resource> y <file>.

Atributo:

type, que debe tener uno de los siguientes valores:

URI si es un recurso que está disponible en Internet.

TEXT una descripción de donde se encuentra el recurso.

La versión SCORM 2004 no requiere el uso de este atributo.

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

Elemento <duration>

Representa el tiempo de duración del Componente del Modelo de Contenido en la velocidad prevista. Este elemento es usado para sonidos, películas, simulaciones y similares.

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

Elemento <educational>

Describe las características educacionales y pedagógicas del Componente del Modelo de Contenido.

Es un elemento padre y puede aparecer 0 más veces dentro de su nodo padre.

Hijos:

<learningresourcetype>

<intendedenduserrole>

<context>

<typicalagerange>

<difficulty>

Elemento <learningresourcetype>

En la versión 2004 de SCORM este elemento cambia por <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 <intendedenduserrole>

Este elemento representa el usuario principal para el que fue creado el componente. Si aparecen varios elementos, el dominante es el primero.

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

Vocabulario que debe usarse :

Teacher
Author
Learner
Manager

En la versión 1.3 de SCORM este elemento cambia para <intendedEndUserRole>

Elemento <context>

Describe el ambiente educativo donde el aprendizaje y el uso del componente deben hacerse.

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

Vocabulario que debe usarse :

Primary Education

Secondary Education

Higher Education

University First Cycle

University Second Cycle

University Postgrade

Technical School First Cycle

Technical School Second Cycle

Professional Formation

Continuous Formation

Vocational Training

Para la versión 1.3 de SCORM debe utilizarse el siguiente vocabulario:

school

higher education

training

other

Elemento <typicalagerange>

Representa la edad del usuario final. Este elemento debe referirse a la edad de desarrollo mental de estos usuarios y no a su edad real.

El estándar IEEE recomienda que el valor debe estar en el formato:

Edad Mínima – Edad Máxima o Edad Mínima –

Multiplicidad	SPM	Tipo de Dato	SPM
0 o más	5	LangString	1000

Para SCORM 1.3 este elemento cambia para <typicalAgeRange>

Elemento <difficulty>

Representa el grado de dificultad del Componente para el usuario que fue creado. Estos usuarios finales deben estar caracterizados en los elementos <context> y <typicalagerange>.

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

Vocabulario que debe usarse :

very easy

easy

medium

difficult

very difficult

Elemento <annotation>

Contiene comentarios del uso educacional del Componente del Modelo de Contenido e información, de cuándo y por quién fueron creados los comentarios. Esta categoría permite a los educadores compartir sus valoraciones del Componente del Modelo de Contenido, sugerencias para su uso, etc.

Es un elemento padre y puede aparecer 0 o más veces dentro de su nodo padre.

Hijos:

<person>

<date>

<description>

Elemento <person>

En la versión SCORM 2004 este elemento cambia por <entity>

Define la entidad o entidades que crearon el comentario.

Multiplicidad	SPM	Tipo de Dato	SPM
0 o 1	-	CharacterString	1000

Elemento <date>

Identifica en qué fecha fue creado el comentario.

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

Elemento <description>

Es usado para representar el contenido del comentario

Multiplicidad	SPM	Tipo de Dato	SPM
0 o 1	-	LangString	1000

1.6.3.1 Perfil de Aplicación de los metadatos:

SCORM propone que los metadatos se apliquen en los niveles: Recurso, SCO, Actividad, Organización y Paquete.

Los metadatos pueden aparecer en el fichero XML de dos maneras: físicamente en el documento o en otro fichero referenciado en el elemento <location>.

A nivel de manifiesto:

Describen el paquete como un todo. Los metadatos en este nivel constituyen los *Metadatos de la Agregación de Contenido*.

```
<manifest>
  <metadata>
    <schema>ADL SCORM</schema>
    <schemaversion>CAM 1.3</schemaversion>
    <lom xmlns="http://ltsc.ieee.org/xsd/LOM">
      <general />
      <classification />
      <annotation />
      <lifeCycle />
      <technical />
      <metaMetadata />
      <educational />
      <relation />
      <rights />
    </lom>
  </metadata>
</manifest>
```

Metadatos contenidos en el manifiesto del paquete.

```
<manifest>
  <metadata>
    <schema>ADL SCORM</schema>
    <schemaversion>CAM 1.3</schemaversion>
    <adlcp:location>contentAggregationMetadata.xml
  </adlcp:location>
  </metadata>
</manifest>
```

Metadatos que se encuentran en otro fichero que está referenciado a través del elemento <location>.

A nivel de organización:

Describen la organización del contenido como un todo. Los metadatos en este nivel constituyen los *Metadatos de la Organización de Contenido*.

```
<organizations>
  <organization>
    <title>Introduction to the SCORM</title>
    <item>...</item>
    <item>...</item>
    <metadata>
      <adlcp:location>contentOrganizationMetadata.xml
    </adlcp:location>
    </metadata>
  </organization>
</organizations>
```

Metadatos a nivel de organización.

A nivel de actividad:

Describen el nivel de anidamiento de las Actividades. Los metadatos en este nivel constituyen los *Metadatos de la Actividad*.

```
<organizations>
  <organization>
    <title>Introduction to the SCORM</title>
    <item>
      <title>SCORM 101</title>
      <metadata>
        <adlcp:location>activityMetadata.xml</adlcp:location>
      </metadata>
    </item>
  </organization>
</organizations>
```

Metadatos aplicados al elemento <item>.

A nivel de SCO:

Describen un SCO. Los metadatos en este nivel constituyen los *Metadatos de SCO*. Describen información sobre los recursos de aprendizaje independientemente de su contexto. Se aplica a los elementos <resource> que sean del tipo "sco".

```
<resources>
  <resource type="webcontent" adlcp:scormType="sco" href="sco1.htm">
    <metadata>
      <adlcp:location>SCOMetadata.xml</adlcp:location>
    </metadata>
  </resource>
</resources>
```

Metadatos aplicados a los SCO.

A nivel de Recurso:

Los metadatos en este nivel constituyen los *Metadatos del Recurso*. Estos metadatos están asociados a los recursos y suministran información sobre el Recurso independientemente del contexto en que se encuentre. Se aplica a los elementos <resource> que sean del tipo "asset" y a los elementos <file> que son hijos de <resource>.

```
<resources>
  <resource type="webcontent" adlcp:scormType="asset" href="asset.htm">
    <metadata>
      <adlcp:location>assetMetadata.xml</adlcp:location>
    </metadata>
  </resource>
</resources>
```

Metadatos de un Recurso.

```
<resources>
  <resource type="webcontent" adlcp:scormType="asset" href="asset.htm">
    <file href="asset.htm">
      <metadata>
        <adlcp:location>assetMetadata.xml</adlcp:location>
      </metadata>
    </file>
  </resource>
</resources>
```

Metadatos de un recurso dentro del elemento <file>. [18] [17]

1.7 Repositorios de Objetos de Aprendizaje

Los Repositorios de Objetos de Aprendizaje surgen de la necesidad de compartir recursos educativos que provienen de distintas fuentes y de organizar su almacenamiento de manera que permita potenciar su reutilización.

“Muchos autores plantean que la idea de Repositorio es intrínseca a los Objetos de Aprendizaje. No es posible pensar en objetos de aprendizaje si no se los concibe albergados en repositorios. Como objetos aislados no tienen ninguna relevancia ni significado real. Una manera de comprender los repositorios, es imaginar una combinación entre una biblioteca digital y un buscador como Yahoo o Google, pero mucho más sofisticado que ambos. Por una parte, los objetos son de naturaleza diversa (al contrario que en una biblioteca) y por otra parte, los criterios de búsqueda deben considerar bastante más que títulos, autores o palabras claves. El tipo de componentes albergados en un repositorio, que deben tener sus propias identidades y ser por lo tanto localizables, son tan variados como gráficos, imágenes, textos, “applets”, videos, documentos e integración de ellos como capítulos de un curso o hasta cursos completos. Un aspecto muy importante de los repositorios es que no necesariamente albergan físicamente los objetos que contienen; les basta con “apuntar” a ellos.” [19]

El funcionamiento de un repositorio de objetos es el siguiente: para que el objeto pueda

ser almacenado y localizado para cualquier uso posterior, ha de ser previamente etiquetado como se haría con cualquier material en una biblioteca ordinaria.

En la actualidad existe una tendencia a crear enormes redes de repositorios locales.

CAREO (Campus Alberta Repository of Educational Objects).

Es un proyecto que tiene como principal objetivo la creación de una colección de materiales educativos y multidisciplinarios basados en la Web. Está siendo llevado a cabo por las Universidades de Alberta, Calgary y Atabaca en cooperación con BELLE (Broadband Enabled Lifelong Learning Environment) y CANARIE (Canadian Network for the Advancement of Research in Industry and Education), y forma parte de una de las iniciativas del Campus Alberta Learning.

Ofrece la posibilidad de hacer búsquedas simples de objetos a través de palabras claves o búsquedas avanzadas a través de criterios como: título, descripción, disciplina, formato técnico, tipo de recurso, etc., en cambio, no aparece la búsqueda por nivel educativo. Los responsables hacen la advertencia en la página de que el repositorio de objetos de aprendizaje de CAREO es un prototipo en desarrollo y, por eso, puede estar fuera de servicio periódicamente para mejoras y actualizaciones del sistema.

El Repositorio brinda la posibilidad de registrarse para poder acceder a los objetos de enseñanza. De todas formas, antes de hacerlo se puede consultar cierta información sobre los distintos objetos que posee. En esta página se muestran los objetos más destacados y los cinco más visitados. La información que aparece consiste en una imagen del objeto más destacado, con su título, la fecha de creación, el autor, su origen, una breve descripción y el botón de acceso. En el caso de los más visitados la información se corresponde con el anterior salvo en su presentación, más resumida, sin foto y con una valoración además de la posibilidad de opinar sobre él y verlo.

Este repositorio es de fácil navegación, pero con la limitación de que al ser un proyecto en desarrollo, en ciertos momentos no es accesible (como en la misma página se advierte). Es de libre uso pero no de código abierto.

MarcoPolo

El Proyecto “*Open MarcoPolo*” se refiere a un software de gestión de bibliotecas que permite llevar la administración de sus tareas internas y también brindar servicios a los usuarios. Repositorio pensado para profesores de Primaria y de Secundaria. Tiene varias secciones: Recursos para el profesor, Desarrollo profesional, etc. En la primera sección, MarcoPolo ofrece a los docentes recursos diseñados para ayudar a integrar contenidos de Internet en la enseñanza.

También hay un buscador de contenidos a través de criterios como: tema, contenido, nivel y tipo de recurso. Este repositorio presenta muchos errores en tiempo de ejecución.

Con un enlace a Internet los programas pueden accederse desde cualquier punto de la red, característica que puede ser muy útil para consultar las bases bibliográficas desde puntos externos, lo que se convierte en una dificultad si se está trabajando dentro de una red local sin conexión a Internet.

Actualmente ofrece los módulos de circulación bibliográfica, consulta al catálogo y administración. Es de libre uso pero no de código abierto.

MERLOT (Multimedia Educational Resource for Learning and Online Teaching)

Se dirige principalmente a alumnos universitarios. Es de libre uso pero no de código abierto. Funciona mediante enlaces a materiales de aprendizaje en línea recogidos con anotaciones y evaluaciones.

Los objetos de aprendizaje se pueden buscar por áreas, que a su vez dan paso a una serie de categorías (generales y específicas) o la posibilidad de búsqueda por criterios un tanto escasos: título, autor, fecha de incorporación, valoración o tipo de objeto. No brinda, por ejemplo, la opción de buscar por idioma, por lo que parece dominar el inglés. Una vez seleccionado el objeto para la enseñanza, el esquema de información que presenta es similar al de CAREO, con el título, una posible imagen del objeto, opiniones, comentarios y evaluaciones realizadas por usuarios, con una puntuación que se representa por un número de estrellas.

Aleandría:

Se trata de un repositorio chileno de videos educativos y culturales en Internet, que corresponde a un prototipo de investigación resultante de un proyecto desarrollado por CAREO.

La mayoría de los vídeos son gratuitos. Cada vídeo se acompaña de una ficha descriptiva con información variada, sobre el año de creación, autor, título, duración, disciplina, tamaño, etc.

MOREA (Múltiples Objetos Reutilizables para la Enseñaza y el Aprendizaje)

Es un repositorio de materiales multimedia. Las secciones de este sitio advierten que cuenta con herramientas para desarrollar nuevos materiales de enseñanza reutilizando los que ya existen en el repositorio.

Los recursos están organizados por las grandes áreas de conocimiento, pero también pueden utilizarse otros criterios de búsqueda, como el tema, el autor, el idioma, etc. Ofrece servicios de consulta y búsqueda en repositorios de enseñanza con soporte Web basadas en diferentes estrategias didácticas: simulaciones, estudio de casos. Se basa en el estándar SCORM.

Estas funcionalidades, en su mayoría no son accesibles debido a fallas técnicas de la aplicación o del servidor donde se encuentra alojada. Es de libre uso pero no de código abierto.

COLOR (Colección de Objetos Reusables)

Permite reunir y consultar objetos de aprendizaje, así como editar, generar e incorporar los metadatos y archivos específicos que describen a los recursos informáticos y la combinación de los mismos. Se pueden explorar los objetos existentes y escogerlos. Brinda la posibilidad de hacer búsquedas por descripción, palabras claves, URL y clasificación. Para la implementación de estas funcionalidades tiene en cuenta la especificación IMS CP. Permite el registro libre de los usuarios pero los Objetos de Aprendizaje no son gratuitos.

APROA (Aprendiendo con Objetos de Aprendizaje)

En la página de Aproa se plantea la disponibilidad de una herramienta Web que permite construir Cursos con los Objetos de Aprendizaje bajo el estándar SCORM para diciembre del año 2005, aunque la misma no ha sido publicada aún. El repositorio permite incorporar todo tipo de recursos, basados o no en este estándar, pide clasificarlos y los almacena.

El catálogo APROA pone a disposición una colección de objetos de aprendizaje, muestra una imagen, el título y una descripción para que los desarrolladores de cursos puedan seleccionarlos e incorporarlos en sus actividades de formación. No permite hacer búsquedas avanzadas de los recursos. Es de libre uso pero no de código abierto.

1.8 Herramientas para el desarrollo de la aplicación.

El software libre gana cada día una gran batalla en el campo de la informática contra el software propietario, nuestro país no se ha quedado atrás y ha comenzado a apostar por él, comenzando la migración poco a poco de nuestros sistemas basados en Windows a Linux.

Las ventajas del software libre son muchas y entre ellas se encuentran la gran comunidad de usuarios y programadores que luchan cada día contra los errores encontrados en los sistemas y por llevar a cabo nuevas soluciones que ayuden a impulsar este movimiento. Dentro del mundo de la Web se encuentran productos de gran aceptación en su comunidad de programadores como el Apache, servidor de aplicaciones Web de código abierto, al igual que PHP, lenguaje de programación de creación de páginas Web dinámicas, en el ámbito de los sistemas gestores de base de datos se encuentra MySQL, el sistema gestor de bases de datos más popular del mundo y de altas prestaciones en cuanto a rapidez en las consultas.

Apache

Un servidor Web es un programa que se ejecuta de forma continua en un ordenador, manteniéndose a la espera de peticiones por parte de un cliente (un navegador) y que

contesta a estas peticiones de forma adecuada, sirviendo una página Web que será mostrada en el navegador respondiendo con el contenido que el cliente necesita o mostrando el mensaje correspondiente si se detectó algún error. [3]

Apache es un servidor Web de código abierto. Tiene capacidad para servir páginas tanto de contenido estático, como de contenido dinámico a través de otras herramientas soportadas que facilitan la actualización de los contenidos mediante bases de datos, ficheros u otras fuentes de información. [4]

Funciona en casi todas las plataformas actuales. Debido a esto se puede escoger la que más se adapte a las características de lo que se desea hacer y migrar hacia otra si en un momento determinado esta ofrece más ventajas.

Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado código cuando ocurra un error en concreto. Puede facilitar información en varios formatos para que un determinado cliente pueda interpretarla.

Las funciones de Apache pueden expandirse mediante módulos. Por ejemplo, Apache es capaz de ejecutar códigos CGI (Common Gateway Interface) en múltiples lenguajes de programación con ayuda de módulos. Además existen módulos que posibilitan, entre otras muchas cosas, la transmisión segura de los datos, la autenticación de usuarios, etc.

Su modularización le permite ser un servidor rápido y muy flexible. Soporte para servidores virtuales y un sistema de registro muy configurable lo convierten en la elección de los grandes sitios de Internet. [6]

PHP

PHP, acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje de código abierto interpretado de alto nivel, especialmente pensado para desarrollos Web, el cual puede ser embebido en páginas HTML (HyperText Markup Language). Es ejecutado en el servidor.

La meta de este lenguaje es permitir escribir a los creadores de páginas Web, páginas dinámicas de una manera rápida y fácil. Puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux, muchas variantes Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS. Es soportado por la mayoría de los servidores Web de hoy en día, incluyendo Apache, Microsoft Internet Information Server, Personal Web Server, Netscape e iPlanet, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd y muchos otros. De modo que, con PHP se tiene la libertad de elegir el sistema operativo y el servidor. También se tiene la posibilidad de usar programación procedimental u orientada a objetos.

Con PHP no se encuentra limitado a resultados en HTML. Entre sus habilidades se incluyen: creación de imágenes, archivos PDF y películas Flash. También puede presentar otros resultados, como archivos XML. Con PHP se pueden generar estos archivos y almacenarlos en el sistema de archivos en vez de presentarlos en la pantalla.

También cuenta con una extensión DBX (Data Base X, cualquier base de datos) de abstracción de base de datos que permite usar de forma transparente cualquier base de datos soportada por la extensión. Adicionalmente, PHP soporta ODBC (el Estándar Abierto de Conexión con Bases de Datos), así que puede conectarse a cualquier base de datos que soporte tal estándar.

PHP tiene unas características muy útiles para el procesamiento de texto, como documentos XML. [11]

MySQL

MySQL es un gestor de bases de datos relacionales, de código abierto, muy rápido, fiable, fácil de usar y robusto. El modelo relacional se caracteriza por disponer que toda la información esté contenida en tablas, y las relaciones entre datos deben ser representadas explícitamente en esos mismos datos, esto añade velocidad y flexibilidad.

Es el más indicado para aplicaciones que requieren muchas lecturas y pocas escrituras y no necesitan características muy avanzadas, como es el caso de las aplicaciones Web.

La meta de este lenguaje es permitir escribir a los creadores de páginas Web, páginas dinámicas de una manera rápida y fácil. Puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux, muchas variantes Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS. Es soportado por la mayoría de los servidores Web de hoy en día, incluyendo Apache, Microsoft Internet Information Server, Personal Web Server, Netscape e iPlanet, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd y muchos otros. De modo que, con PHP se tiene la libertad de elegir el sistema operativo y el servidor. También se tiene la posibilidad de usar programación procedimental u orientada a objetos.

Con PHP no se encuentra limitado a resultados en HTML. Entre sus habilidades se incluyen: creación de imágenes, archivos PDF y películas Flash. También puede presentar otros resultados, como archivos XML. Con PHP se pueden generar estos archivos y almacenarlos en el sistema de archivos en vez de presentarlos en la pantalla.

También cuenta con una extensión DBX (Data Base X, cualquier base de datos) de abstracción de base de datos que permite usar de forma transparente cualquier base de datos soportada por la extensión. Adicionalmente, PHP soporta ODBC (el Estándar Abierto de Conexión con Bases de Datos), así que puede conectarse a cualquier base de datos que soporte tal estándar.

PHP tiene unas características muy útiles para el procesamiento de texto, como documentos XML. [11]

MySQL

MySQL es un gestor de bases de datos relacionales, de código abierto, muy rápido, fiable, fácil de usar y robusto. El modelo relacional se caracteriza por disponer que toda la información esté contenida en tablas, y las relaciones entre datos deben ser representadas explícitamente en esos mismos datos, esto añade velocidad y flexibilidad.

Es el más indicado para aplicaciones que requieren muchas lecturas y pocas escrituras y no necesitan características muy avanzadas, como es el caso de las aplicaciones Web.

MySQL fue desarrollado para ocuparse de bases de datos mucho más rápido que las soluciones existentes.

Su conectividad, velocidad, seguridad hacen que este servidor sea uno de los preferidos, consume pocos recursos de la CPU (Central Processing Unit o Unidad Central de Procesamiento) y memoria, además de proporcionar muy buen rendimiento, tamaño del registro sin límite, buena integración con PHP, utilidades de administración y buen control de acceso usuarios-tablas-permisos.

MySQL puede usar varias CPU, es decir, es multiproceso. Puede trabajar en distintas plataformas y Sistemas Operativos. Cuenta con un sistema de contraseñas y privilegios muy flexible y seguro. Todas las columnas pueden tener valores por defecto.

Sin embargo este poderoso gestor de bases de datos tiene sus desventajas ya que no soporta triggers ni procedimientos en el servidor, además de que no soporta claves ajenas, no soporta vistas. Inconvenientes que no afectan en el desarrollo de este trabajo.

XML

XML (*eXtensible Markup Language, Lenguaje Extensible de Marcas*) fue desarrollado por un grupo de trabajo bajo los auspicios del consorcio World Wide Web (W3C) a partir de 1996.

XML es un lenguaje de marcas que ofrece un formato para la descripción de datos estructurados. Las construcciones como etiquetas, referencias de entidad y declaraciones se denominan "marcas". Éstas son las partes del documento que el procesador XML espera entender. El resto del documento que se encuentra entre las marcas, son los datos que resultan entendibles por las personas.

XML es un metalenguaje, dado que con él se puede definir un lenguaje propio para la creación y manipulación de documentos contenedores de datos descritos en sí mismos y, que no se centra en la representación de la información, XML se centra en la información en sí misma. La particularidad más importante del XML es que no posee

etiquetas prefijadas con anterioridad, ya que es el propio diseñador el que las crea a su antojo.

La tecnología XML pretende convertirse en la columna vertebral del intercambio de información hombre-máquina en pos de la consolidación de la sociedad del conocimiento. Esto se debe a que hoy en día los requerimientos funcionales de los sistemas de gestión implican un procesamiento cada vez más inteligente de los datos por parte de las máquinas, permitiendo obtener recuperaciones más ricas en información, en pos de la gestión del conocimiento en las más diversas esferas del desarrollo humano.

XML tiende a ser el lenguaje que avalará el intercambio de cualquier tipo de información, sin que ocasione problemas de tipo "contenido" o de tipo "presentación". Este garantiza que los datos estructurados sean uniformes e independientes de aplicaciones o fabricantes, lo que está originando una nueva generación de aplicaciones. [5]

1.9 Metodologías para el desarrollo de Sistemas Informáticos.

Influenciado por el gran desarrollo de las computadoras y el uso creciente de Internet se construyen software más grandes y más complejos. Se requiere de un software que esté mejor adaptado a las necesidades de todos, lo que hace que sea más complejo, pero no solo esto, sino que además, se requiere de un software construido en el menor tiempo posible y que sea eficiente.

“La presencia de un proceso bien definido y bien gestionado es una diferencia esencial entre los proyectos hiperproductivos y los que fracasan. El Proceso Unificado de Desarrollo es una solución al problema del software.” [10]

Proceso Unificado de Desarrollo de Software.

El Proceso Unificado es un proceso de desarrollo de software, es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. [8]

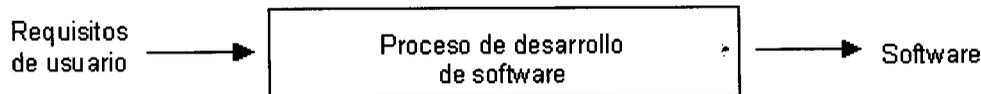


Figura 1. Un proceso de desarrollo de software

El Proceso Unificado está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces.

Sus características fundamentales, y que convierten en único al Proceso Unificado, son: está dirigido por casos de uso, está centrado en la arquitectura y es iterativo e incremental.

El Proceso Unificado propone un modelamiento visual de la estructura y los componentes, para ello utiliza el Lenguaje Unificado de Modelado (Unified Modeling Language, UML).

Un proceso efectivo proporciona normas para el desarrollo eficiente de software de calidad, y captura y presenta las mejores prácticas que la tecnología permite. Por tanto, reduce el riesgo y hace el proyecto más predecible.

Dirigido por casos de uso

Un caso de uso es un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante. Los casos de uso representan los requisitos funcionales de un sistema, guían su diseño, implementación y prueba; esto es, *guían el proceso de desarrollo*. Los casos de uso proporcionan un hilo conductor para el proceso de desarrollo. Dirigido por casos de uso significa que el proceso sigue un hilo, avanza a través de una serie de flujos de trabajo que parten de los casos de uso.

Centrado en la arquitectura

El concepto de arquitectura software incluye los aspectos estáticos y dinámicos más significativos del sistema y se refleja en los casos de uso. La arquitectura es una vista del diseño completo con las características más importantes, dejando a un lado los detalles.

Esta no solo incluye las necesidades de los usuarios e inversores, sino también otros aspectos técnicos como el hardware, sistema operativo, sistema de gestión de base de datos, protocolos de red; con los que debe coexistir el sistema.

Cada producto tiene tanto una función como una forma. La función corresponde a los casos de uso y la forma a la arquitectura. La arquitectura va a ir madurando en su interacción con los casos de uso hasta llegar a un equilibrio entre funcionalidad y características técnicas.

Iterativo e incremental

El trabajo se divide en partes más pequeñas o miniproyectos, cada uno de ellos es una iteración que representa un incremento en el grado de terminación del producto.

En cada iteración los desarrolladores identifican y especifican los casos de usos relevantes, los cuales se diseñan, implementan y prueban. La planificación de iteraciones hace que se reduzcan los riesgos de los costes de un solo incremento, de sacar al mercado un producto en el tiempo previsto, de mantener la motivación del equipo pues puede ver avances claros a corto plazo y que el desarrollo pueda adaptarse a los cambios en los requisitos. [10]

Rational Rose

Es una herramienta para el modelado visual, que forma parte de un conjunto más amplio de herramientas que juntas cubren todo el ciclo de vida de desarrollo de software. Permite completar gran parte de los flujos fundamentales del Proceso Unificado de Rational (RUP) e incluye un conjunto de herramientas de ingeniería inversa y generación de código.

Conclusiones del Capítulo

En este capítulo fueron expuestos los conceptos principales asociados al dominio del problema. Se hizo un análisis crítico de los repositorios existentes concluyendo que no existe una herramienta de código abierto que se ajuste a la solución del problema de

investigación planteado. Se ofreció una descripción de aquellos aspectos del estándar SCORM que serán de importancia para comprender la propuesta del sistema que se expondrá en el próximo capítulo.

Por otro lado, se expusieron las características principales de las herramientas empleadas en desarrollo de la aplicación, así como las ventajas y desventajas de las mismas, permitiendo fundamentar su uso.

De igual forma, las características descritas del Proceso Unificado de Desarrollo de software demuestran las ventajas de elegir esta metodología.

Capítulo 2. Descripción de la solución propuesta

En este capítulo se describirá el flujo de trabajo de los procesos llevados a cabo para desarrollar el sistema, dándole solución al problema de investigación planteado. Se detallarán los distintos pasos de la metodología propuesta en el capítulo anterior (RUP) para el desarrollo de la aplicación. Primero se realiza un modelo de dominio, donde se capturan las clases más importantes en el contexto del sistema. Se hace un levantamiento de los requisitos no funcionales y funcionales, estos últimos se estructuran mediante los Casos de Uso del sistema, de los cuales se ofrece una descripción textual y el Modelo del Diseño.

El diseño de la base de datos queda expuesto a través de una breve descripción de su estructura y el Diagrama Entidad-Relación (DER). La arquitectura del sistema propuesto se expresa a través del diagrama de Despliegue.

2.1 Modelo de Dominio

Hay por lo menos dos aproximaciones para expresar el contexto de un sistema en una forma utilizable para los desarrolladores de software: modelado del dominio y modelado del negocio. Un modelo del dominio describe los conceptos importantes del contexto como objetos del dominio y enlaza estos objetos unos con otros. La identificación y asignación de un nombre para estos objetos ayuda a desarrollar un glosario de términos que permitirán comunicarse mejor a los que trabajen en un sistema.

“Las entidades del negocio se derivan a partir de los clientes del negocio. Las clases del dominio se obtienen de la base del conocimiento de unos pocos expertos del dominio, o posiblemente del conocimiento (otras clases del dominio, especificaciones de requisitos, etc.) asociado con sistemas similares al que se está desarrollando.” [10]

Tal es el caso del presente trabajo donde se desarrolla este modelo a partir del estudio de las especificaciones propuestas por el estándar SCORM y del funcionamiento de algunos de los repositorios existentes.

2.1.1 Descripción del Problema de Dominio

El sistema *Repositorio de Objetos de Aprendizaje para la reutilización de contenidos en plataformas de teleformación* permite disponer de un lugar común donde profesores y estudiantes almacenen los recursos empleados en el proceso de enseñanza aprendizaje. Se desea que el sistema sea accesible a través de un entorno Intranet / Internet.

El sistema presenta en su pantalla principal un mensaje de bienvenida describiendo los servicios ofrecidos al usuario junto con la opción para registrarse, o si ya se está registrado, poder utilizar el sistema. Este acceso se da por medio de la inserción de un *nombre de usuario* y una *contraseña* previamente escogidos y que deben validarse.

Una vez registrado el usuario, y después de haberse validado su acceso, puede seleccionar las siguientes actividades:

Subir Paquete:

Permite a los usuarios subir Objetos de Aprendizaje empaquetados según el estándar SCORM, especificando la categoría en la que se incluye el paquete (PIF) que los contiene. El sistema verifica que el PIF se ajuste a las especificaciones propuestas por este estándar.

Actualizar Paquete:

Los usuarios pueden actualizar los PIFs que han aportado al Repositorio. El sistema brinda la posibilidad de cambiarle la categoría, especificar si estarán disponibles o no para el resto de los usuarios, así como borrarlos definitivamente.

Buscar recursos:

Asiste a los usuarios en la búsqueda de recursos, que puede restringirse especificando una serie de criterios con los que deben cumplir los objetos que espera encontrar. Ello es posible debido a que los componentes pueden describirse mediante el empleo de metadatos.

Explorar Paquetes:

Se muestran los PIFs almacenados, agrupados o no por categorías según seleccione el usuario. Además el usuario puede explorar el contenido de los paquetes. }

El sistema ofrecerá además a los usuarios la posibilidad de emitir juicios valorativos, ya sea cualitativa o cuantitativamente, acerca de los PIFs y los Objetos de Aprendizaje.

2.1.2 Glosario de Términos

Juicio cualitativo: Valoración de la utilidad del material de aprendizaje que emite el usuario a través opiniones.

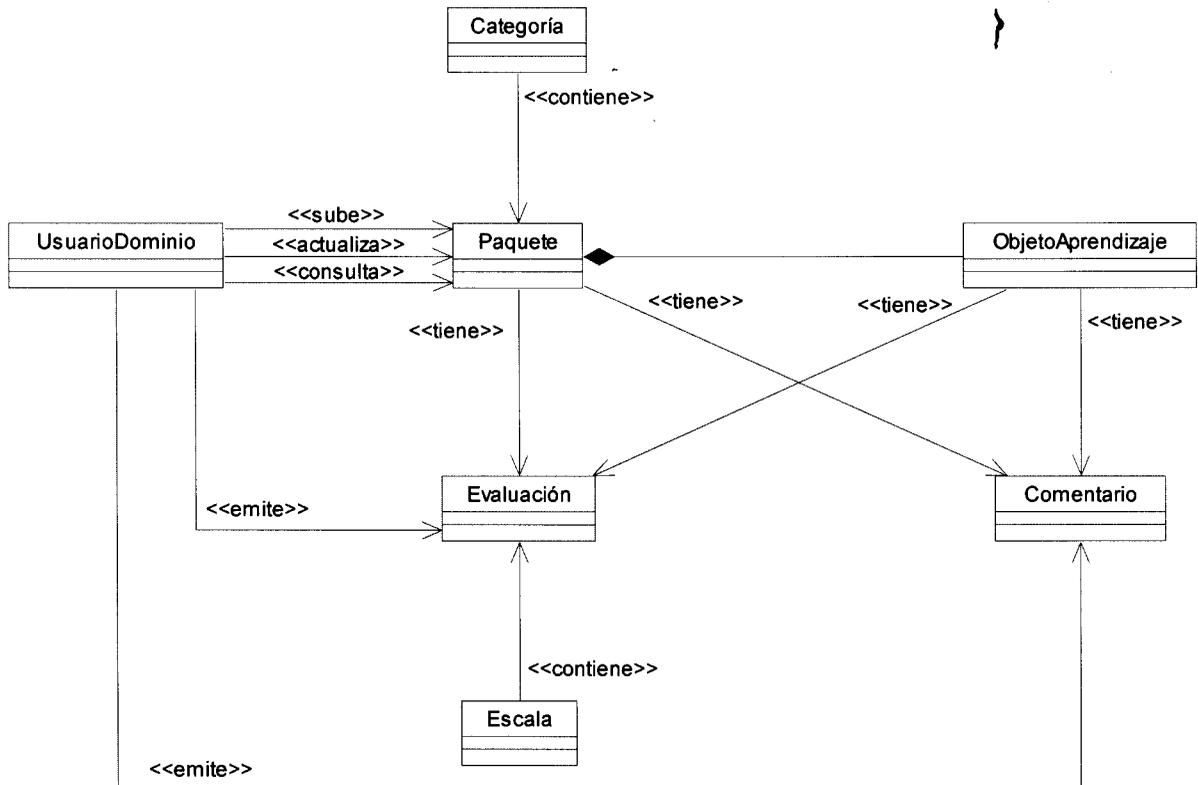
Juicio cuantitativo: Evaluación del material de aprendizaje que proporciona el usuario según una escala.

Objeto de Aprendizaje: Cualquier entidad digital que pueda ser usada en el aprendizaje. Representación digital de texto, imágenes, audio, video, etc.

PIF: Conjunto de Objetos de Aprendizaje empaquetados según el estándar SCORM.

SCORM: Shareable Content Object Reference Model (Modelo de Referencia de Objetos de Contenido Intercambiable).

2.1.3 Diagrama de clases del dominio.



2.2 Requisitos

2.2.1 Funcionales

1. Registrar Usuario.
2. Autenticar Usuario.
3. Actualizar Usuario
4. Subir Paquete al Repositorio.
5. Actualizar Paquete.
6. Buscar Recurso de aprendizaje.
7. Explorar Paquetes.
8. Visualizar contenido de Paquetes. Visualizar los OAs que están contenidos en los paquetes seleccionados con R8.
9. Evaluar utilidad de OA.
10. Evaluar utilidad de Paquete.
11. Visualizar opiniones de OA.
12. Visualizar opiniones de Paquete.

2.2.2 No Funcionales

Requerimientos de Hardware

- ✓ Conectividad. Red de Área Local (LAN)
- ✓ Respaldo eléctrico (UPS)
- ✓ Microprocesador 200 MHz.
- ✓ 32 MB de memoria RAM.
- ✓ 4 GB de disco duro.

Requerimientos de Software

- ✓ Funciona en cualquier plataforma que soporte apache+php+mysql. Preferentemente en los Sistemas Operativos Linux (Debian Sarge 3.1) y Windows XP.
- ✓ Servidor Web Apache.
- ✓ Sistema Gestor de Base de Datos MySQL versión 4.0.24 o superior.
- ✓ PHP 4.3 o superior compilado con soporte para MySQL, DOM XML y LDAP.

Requerimientos de Seguridad

- ✓ La información manejada por el sistema estará protegida de acceso no autorizado.

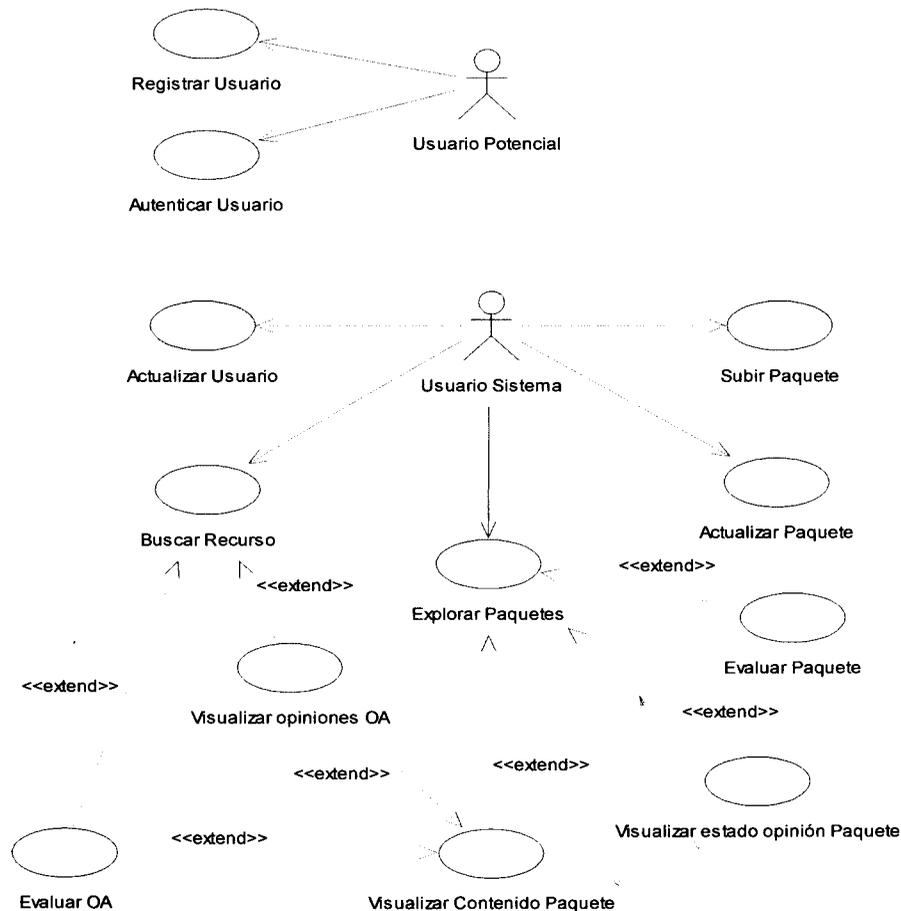
Requerimientos de Ayudas y Documentación en línea

- ✓ El directorio de instalación del sistema incluirá el archivo LEEME.txt donde se expondrán las instrucciones a seguir para realizar una correcta instalación.

2.3 Actores del Sistema

Usuario Potencial	Representa una persona que aún no forma parte del sistema y está a punto de registrarse o se ha registrado pero no se ha autenticado en el sistema.
Usuario Sistema	Representa una persona cuyos datos fueron registrados en el sistema, se ha autenticado y por tanto puede aportar nuevos recursos así como descargar y ofrecer criterios valorativos de los que ya se encuentran almacenados en el sistema. También puede actualizar sus datos personales y los de los paquetes que ha subido.

2.4 Diagrama de Casos de Uso del Sistema



2.5 Descripción de los Casos de Uso

Nombre del caso de uso	Registrar Usuario.
Actores	Usuario Potencial (Inicia).
Resumen	El caso de uso inicia cuando una persona decide crear una cuenta de acceso al sitio. El sistema solicita los datos de inscripción, que el usuario debe proporcionar y chequea que no existan colisiones. Una vez hecha la comprobación se crea la nueva cuenta, en caso contrario se retorna un mensaje error y se ofrece la posibilidad de intentarlo nuevamente. El caso de uso finaliza cuando el sistema ha creado la cuenta de usuario.
Referencias	R1
Precondiciones	No debe existir un usuario con el alias que el Usuario Potencial proporcionó.
Poscondiciones	La cuenta de usuario ha sido creada.

Nombre del caso de uso	Autenticar Usuario.
Actores	Usuario Potencial (Inicia).
Resumen	El caso de uso inicia cuando el usuario decide entrar al sistema y proporciona su nombre y contraseña. El sistema verifica que estos datos sean correctos y le da acceso, en caso contrario notifica un error y le brinda la posibilidad de acceder nuevamente. El caso de uso finaliza cuando el usuario se ha autenticado.
Referencias	R2
Precondiciones	El usuario debe estar registrado.
Poscondiciones	El usuario queda validado.

Nombre del caso de uso	Actualizar Usuario.
Actores	Usuario Sistema (Inicia).
Resumen	El caso de uso inicia cuando el usuario decide actualizar sus datos personales. El sistema edita los datos del usuario, el cual hace los cambios que desee. El caso de uso finaliza cuando se han actualizado los datos, en caso contrario muestra un mensaje de error.
Referencias	R3
Precondiciones	El usuario debe estar autenticado.
Poscondiciones	Los datos de usuario quedan actualizados.

Nombre del caso de uso	Subir Paquete.
Actores	Usuario Sistema (Inicia).
Resumen	El caso de uso inicia cuando el usuario decide subir un paquete. El sistema solicita el archivo que representa el paquete, lo sube al servidor, chequea que su formato sea válido y se actualizan sus datos, en caso contrario el paquete se desecha, se muestra un error y se brinda la posibilidad de subir otro fichero. El caso de uso finaliza cuando el paquete queda almacenado.
Referencias	R4
Precondiciones	El usuario debe estar autenticado.
Poscondiciones	El paquete queda almacenado y se actualizan sus datos.

Nombre del caso de uso	Actualizar Paquete.
Actores	Usuario Sistema (Inicia).
Resumen	El caso de uso inicia cuando el usuario decide actualizar un paquete. El sistema muestra sólo aquellos paquetes que han sido aportados por el usuario, quien selecciona uno y lo actualiza. El caso de uso finaliza cuando el paquete se ha actualizado.
Referencias	R5
Precondiciones	El usuario debe estar autenticado y debe haber aportado algún paquete.
Poscondiciones	Los datos del paquete quedan actualizados.

Nombre del caso de uso	Buscar recursos de aprendizaje.
Actores	Usuario Sistema (Inicia).
Resumen	El caso de uso inicia cuando el usuario decide buscar algún recurso. El sistema muestra una serie de criterios que el usuario puede utilizar para especificar las características del contenido que busca. El caso de uso termina cuando los contenidos que cumplen con los criterios de búsqueda se ponen disponibles para descargarlos, evaluarlos, opinar acerca de ellos y ver otras opiniones relacionadas.
Referencias	R6
Precondiciones	El usuario debe estar autenticado.
Poscondiciones	–

Nombre del caso de uso	Explorar Paquetes.
Actores	Usuario Sistema (Inicia).
Resumen	El caso de uso inicia cuando el usuario decide explorar los paquetes almacenados en el repositorio. El sistema presenta los paquetes agrupados o no por determinada categoría según seleccione el usuario. El caso de uso termina cuando los paquetes quedan disponibles para ser descargados, evaluados, opinar acerca de ellos, ver otras opiniones relacionadas y ver los recursos contenidos en ellos.
Referencias	R7
Precondiciones	El usuario debe estar autenticado.
Poscondiciones	–

Nombre del caso de uso	Visualizar contenido Paquete.
Actores	Usuario Sistema (Inicia).
Resumen	El caso de uso inicia si, una vez realizado el CU Explorar Paquetes, el usuario desea revisar el contenido del paquete. El sistema muestra cada recurso que se encuentra dentro del paquete, terminando así el caso de uso.
Referencias	R8
Precondiciones	El usuario debe estar autenticado.
Poscondiciones	–

Nombre del caso de uso	Evaluar OA.
Actores	Usuario Sistema (Inicia).
Resumen	El caso de uso inicia si, una vez realizado el CU Buscar recursos de aprendizaje o el CU Visualizar contenido de Paquete, el usuario decide evaluar un recurso. El usuario puede otorgar una evaluación y/o emitir su opinión acerca del recurso. El caso de uso termina cuando se registran estos criterios.
Referencias	R9
Precondiciones	El usuario debe estar autenticado.
Poscondiciones	Se actualizan la evaluación y/o las opiniones del recurso.

Nombre del caso de uso	Evaluar Paquete.
Actores	Usuario Sistema (Inicia).
Resumen	El caso de uso inicia si, una vez realizado el CU Explorar Paquetes, el usuario decide evaluar el paquete. El usuario puede otorgar una evaluación y/o emitir su opinión acerca de un paquete. El caso de uso termina cuando se registran estos criterios.
Referencias	R10
Precondiciones	El usuario debe estar autenticado.
Poscondiciones	Se actualizan la evaluación y/o las opiniones del paquete.

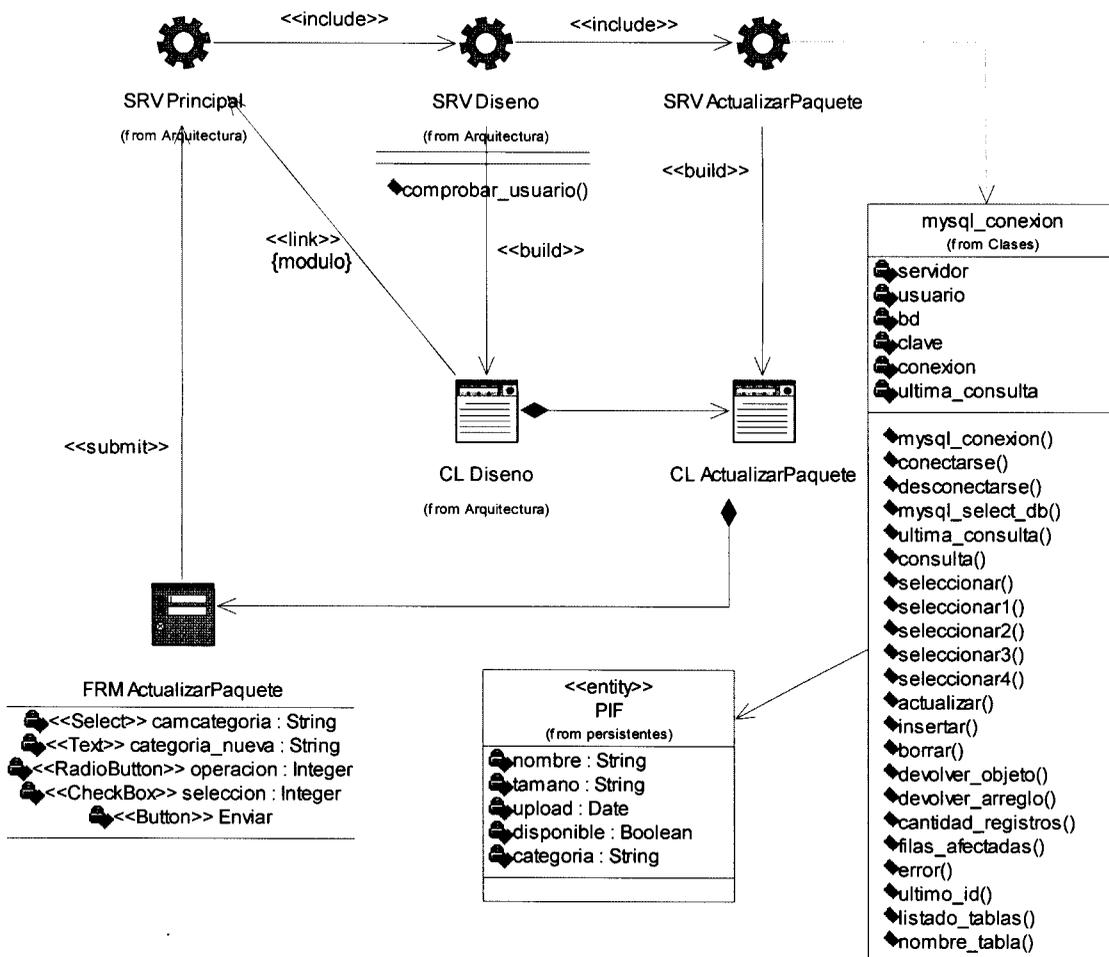
Nombre del caso de uso	Visualizar opiniones OÃ.
Actores	Usuario Sistema (Inicia).
Resumen	El caso de uso inicia si, una vez realizado el CU Buscar recursos de aprendizaje o el CU Visualizar contenido de Paquete, el usuario desea ver las opiniones de un recurso. El sistema muestra las opiniones recogidas acerca del recurso, terminando así el caso de uso.
Referencias	R11
Precondiciones	El usuario debe estar autenticado.
Poscondiciones	-

Nombre del caso de uso	Visualizar opiniones Paquete.
Actores	Usuario Sistema (Inicia).
Resumen	El caso de uso inicia si, una vez realizado el CU Explorar Paquetes, el usuario desea ver las opiniones de un paquete. El sistema muestra las opiniones recogidas acerca del Paquete, terminando así el caso de uso.
Referencias	R12
Precondiciones	El usuario debe estar autenticado.
Poscondiciones	-

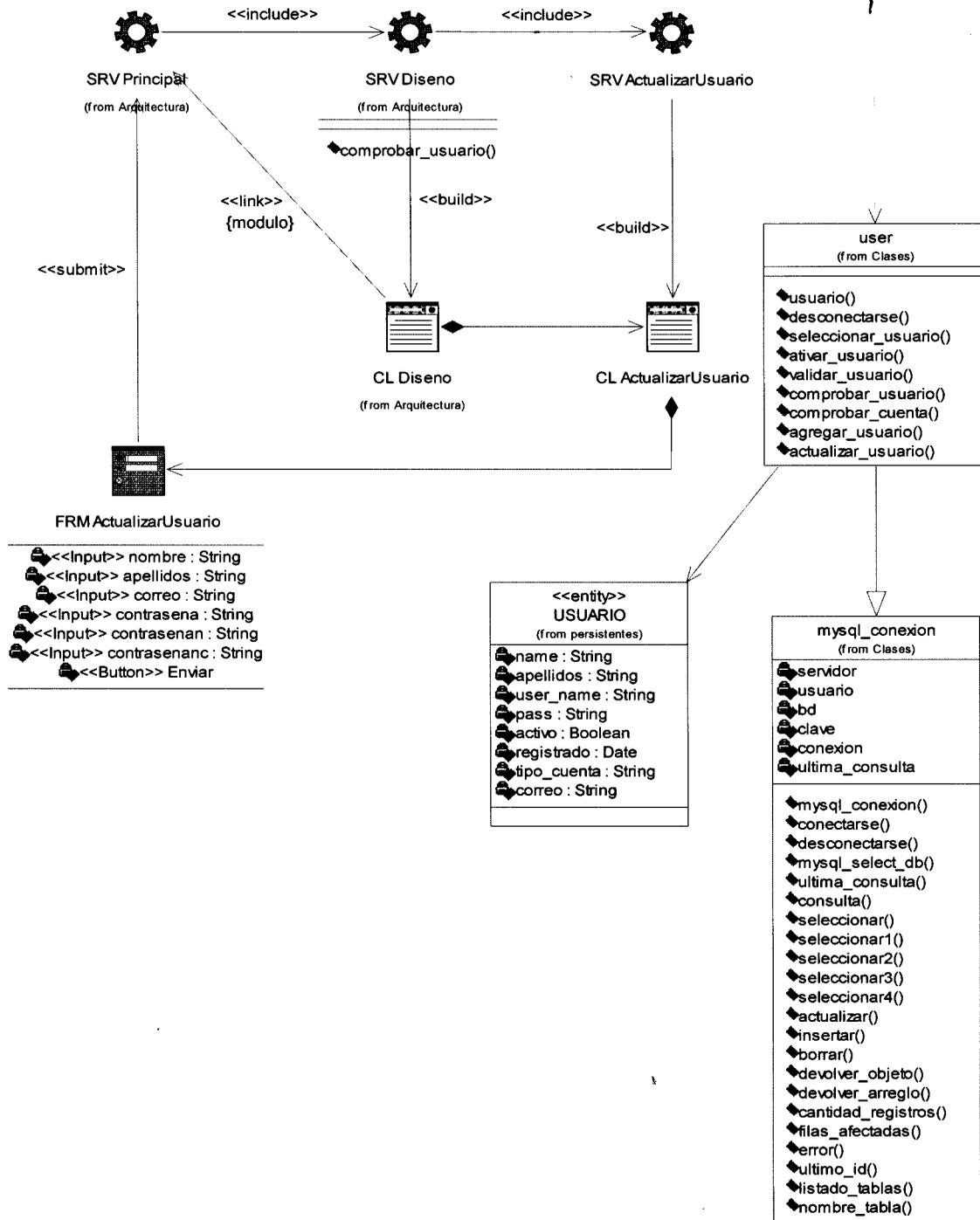
2.6 Diseño

2.6.1 Diagrama de clases del diseño

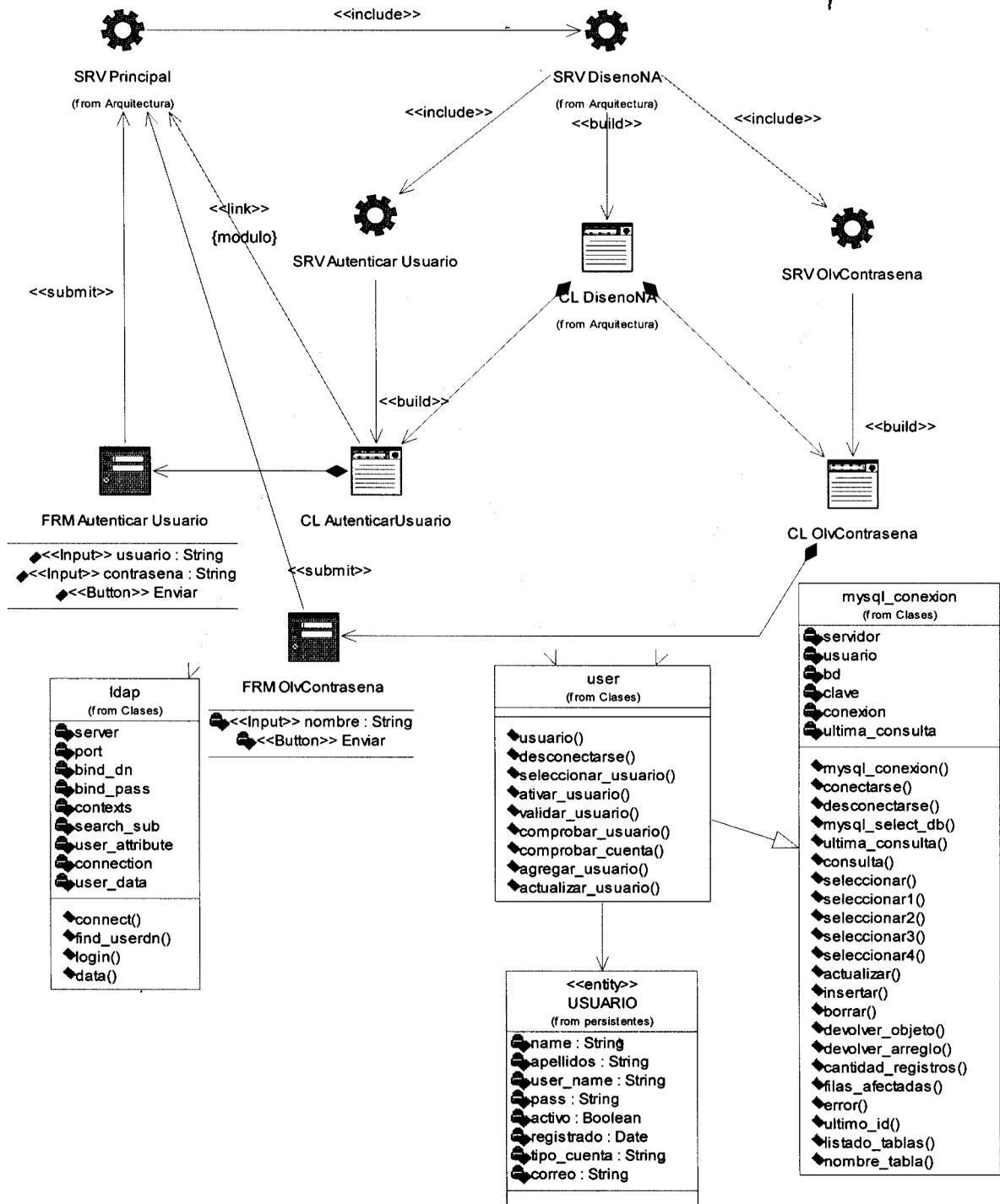
Caso de Uso Actualizar Paquete



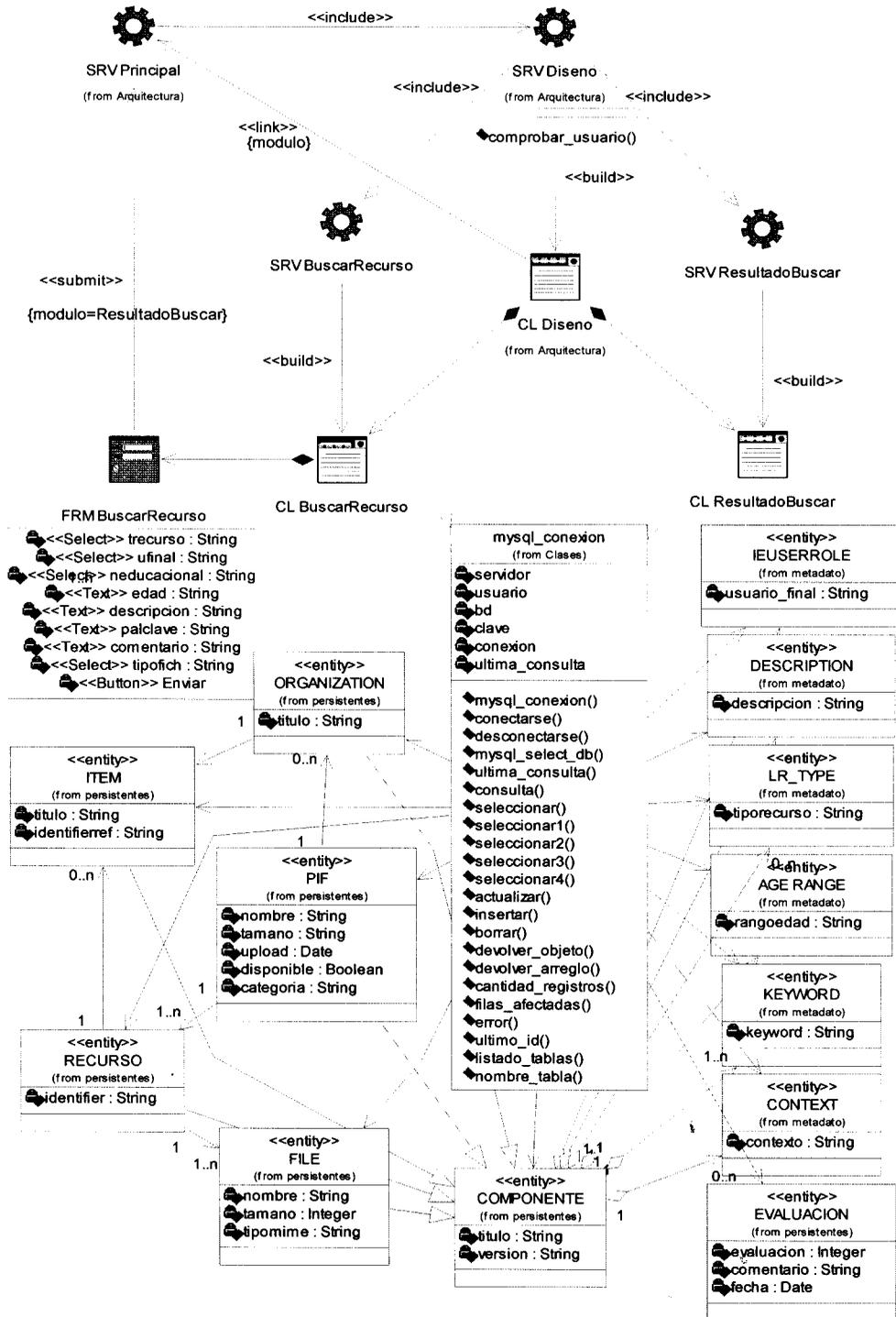
Caso de Uso Actualizar Usuario



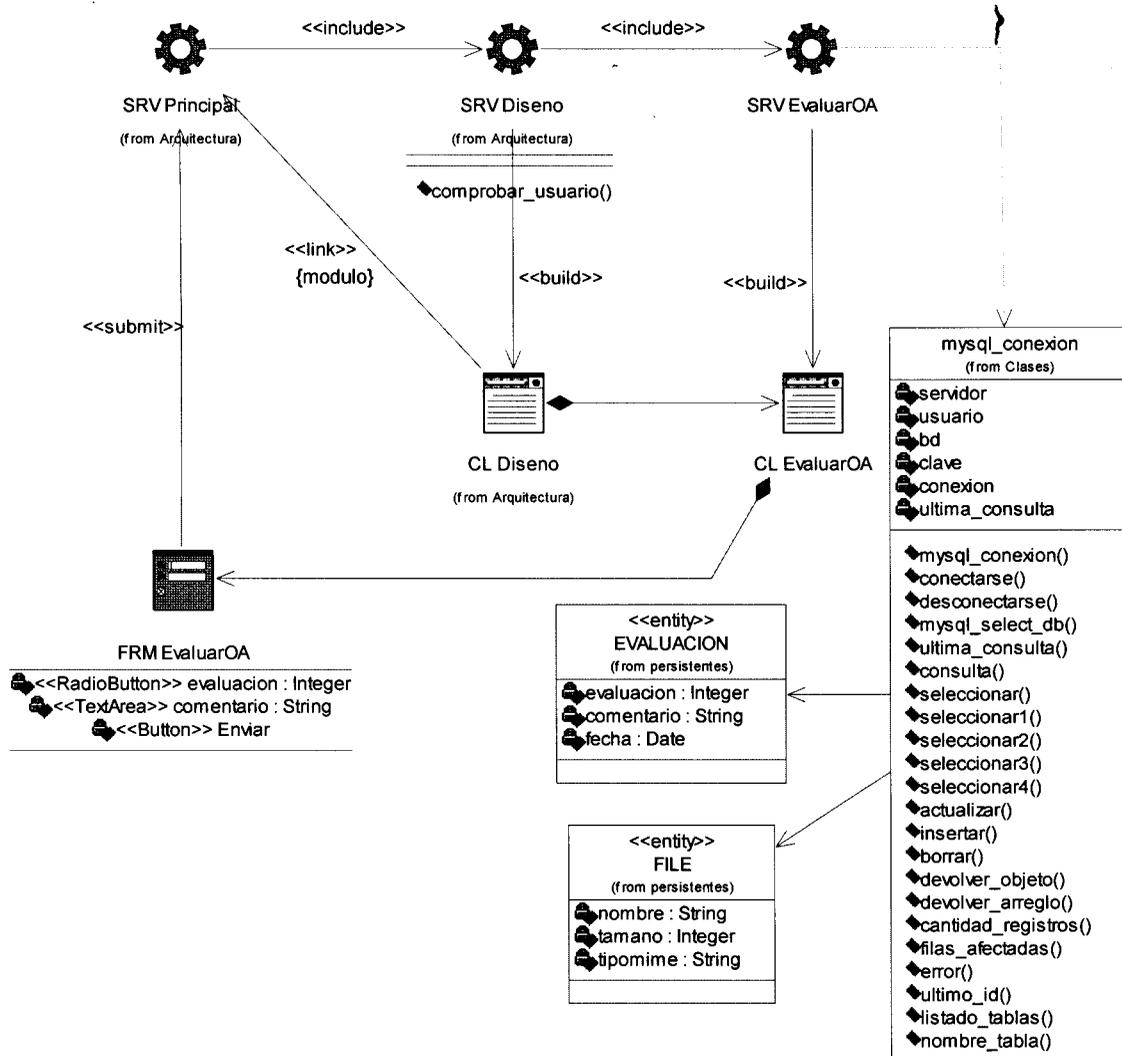
Caso de Uso Autenticar Usuario



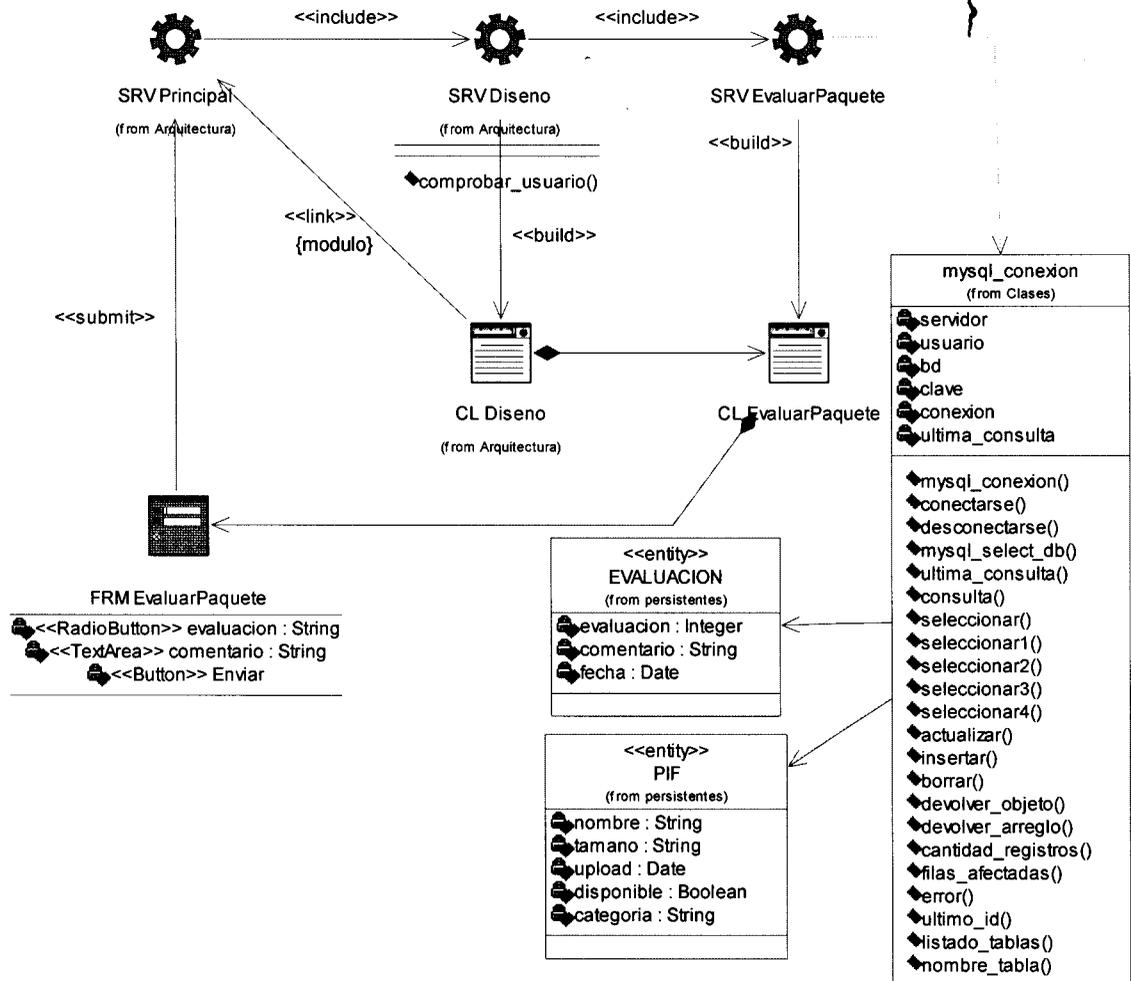
Caso de Uso Buscar Recurso



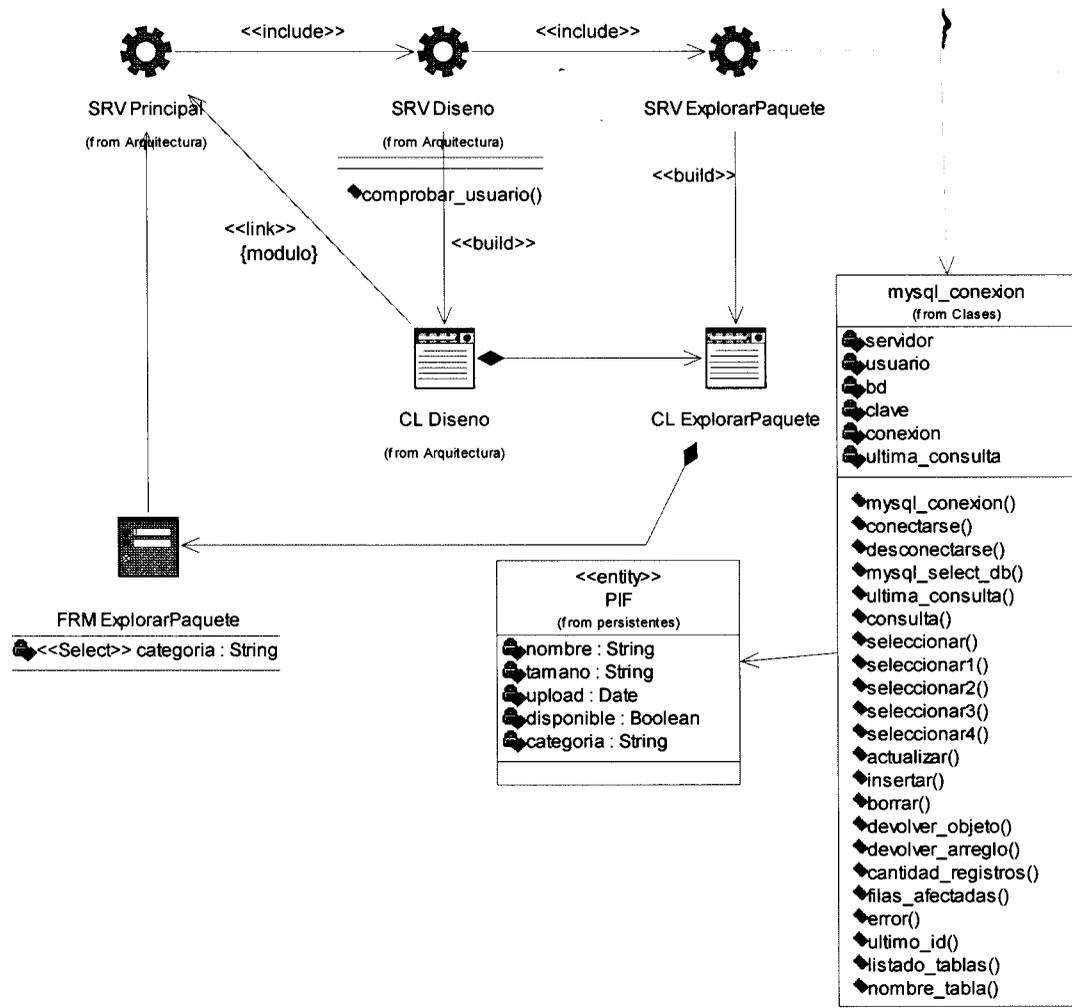
Caso de Uso Evaluar OA



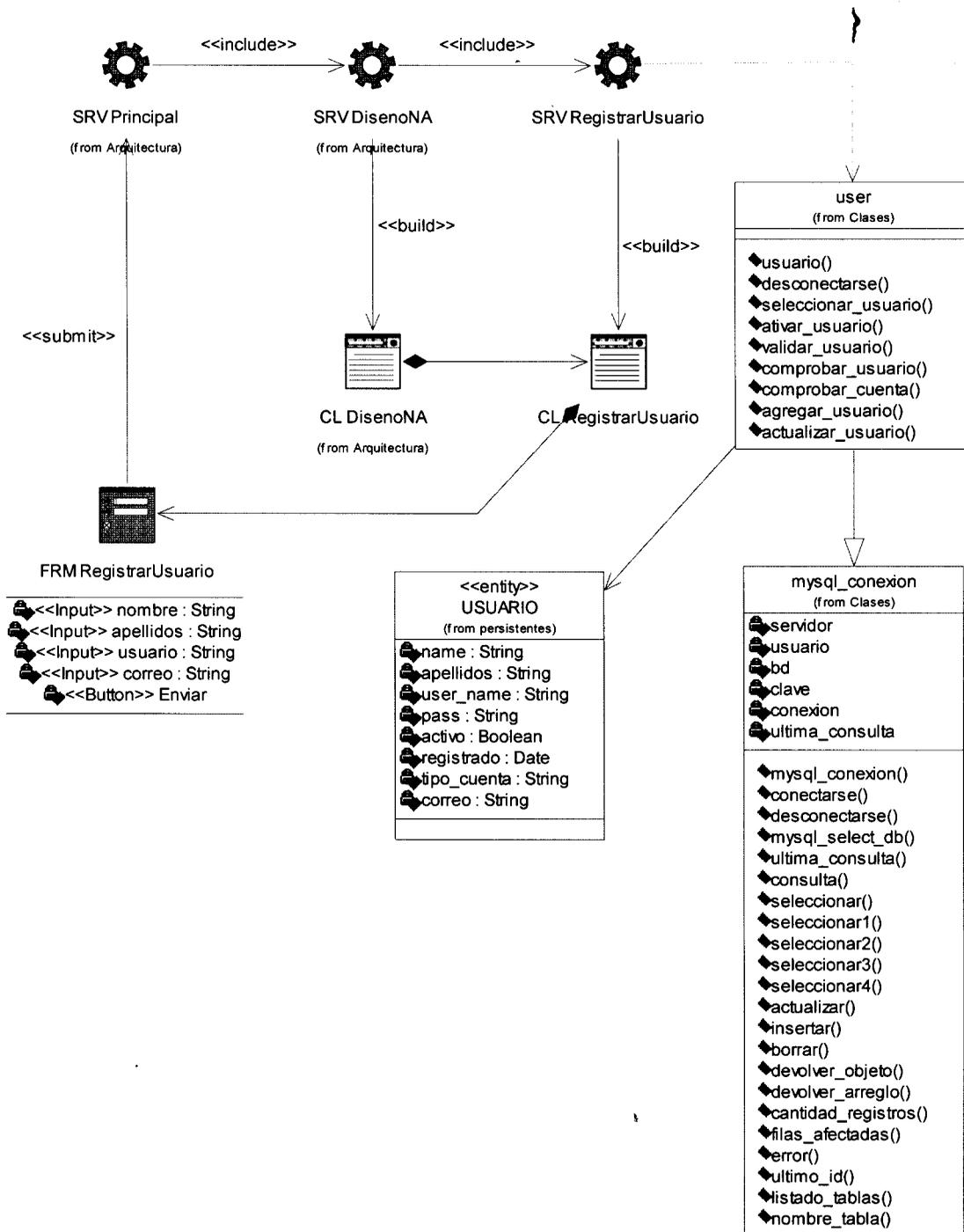
Caso de Uso Evaluar Paquete



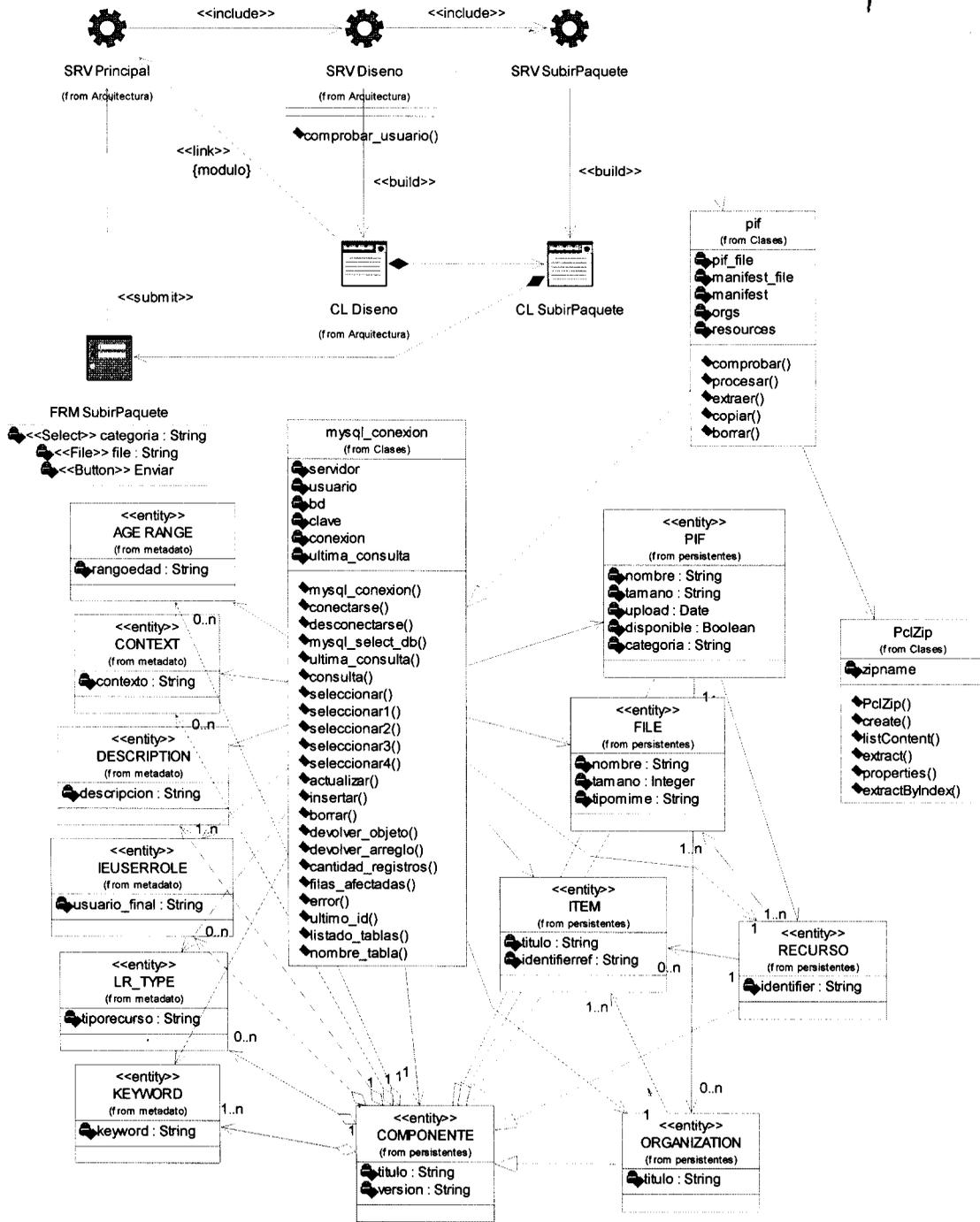
Caso de Uso Explorar Paquetes



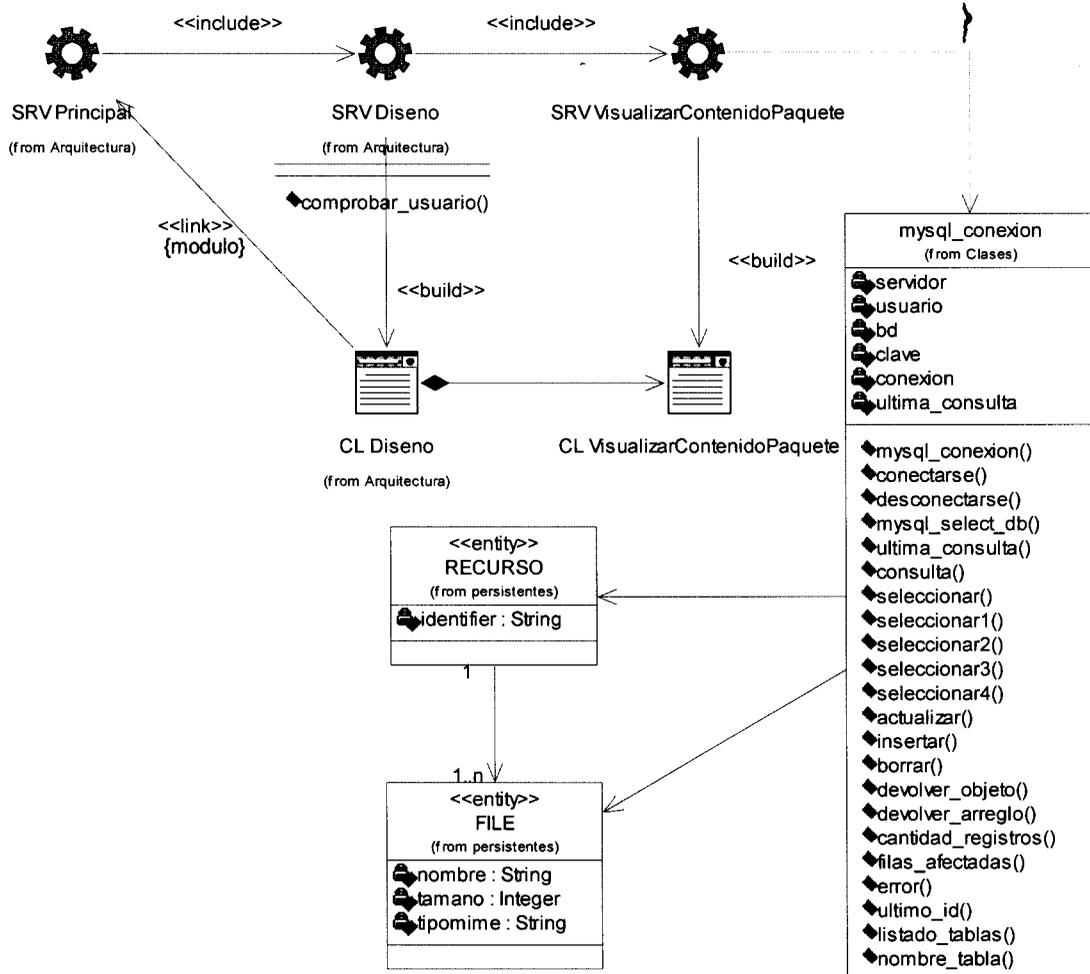
Caso de Uso Registrar Usuario



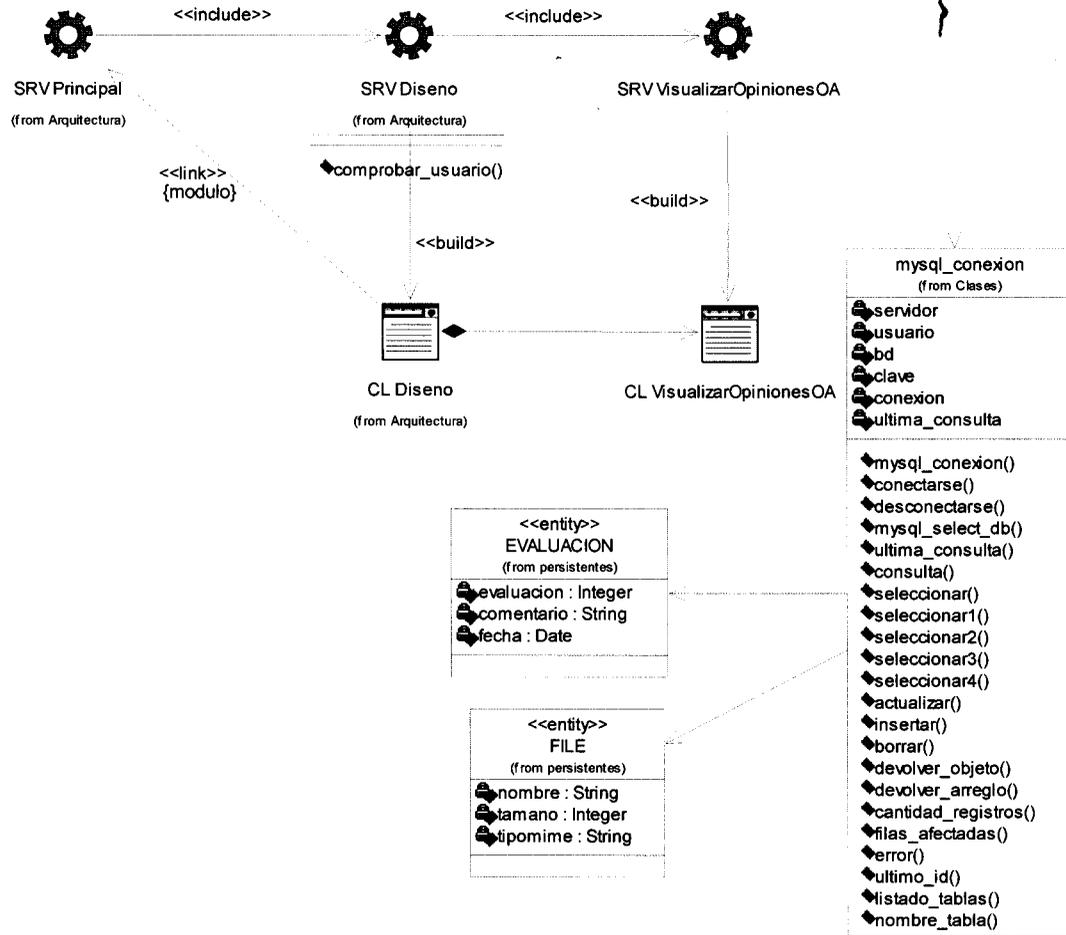
Caso de Uso Subir Paquete



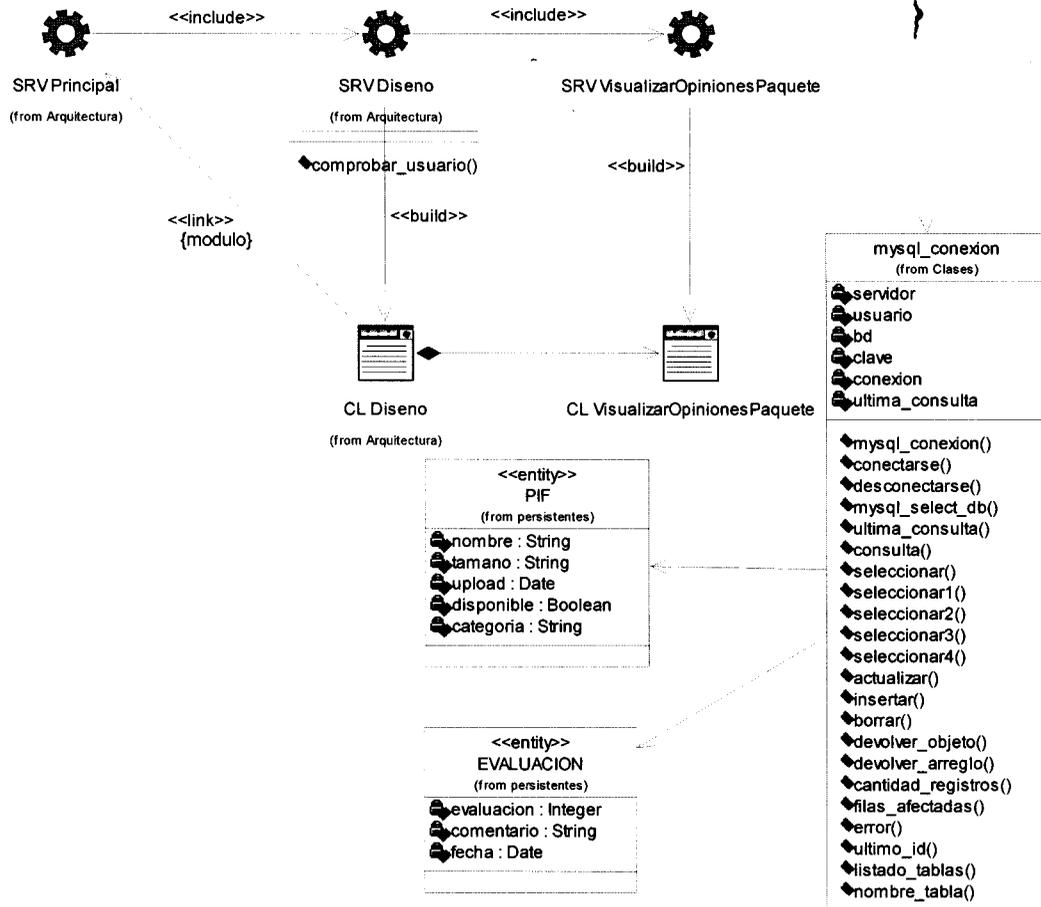
Caso de Uso Visualizar Contenido Paquete



Caso de Uso Visualizar Opiniones OA



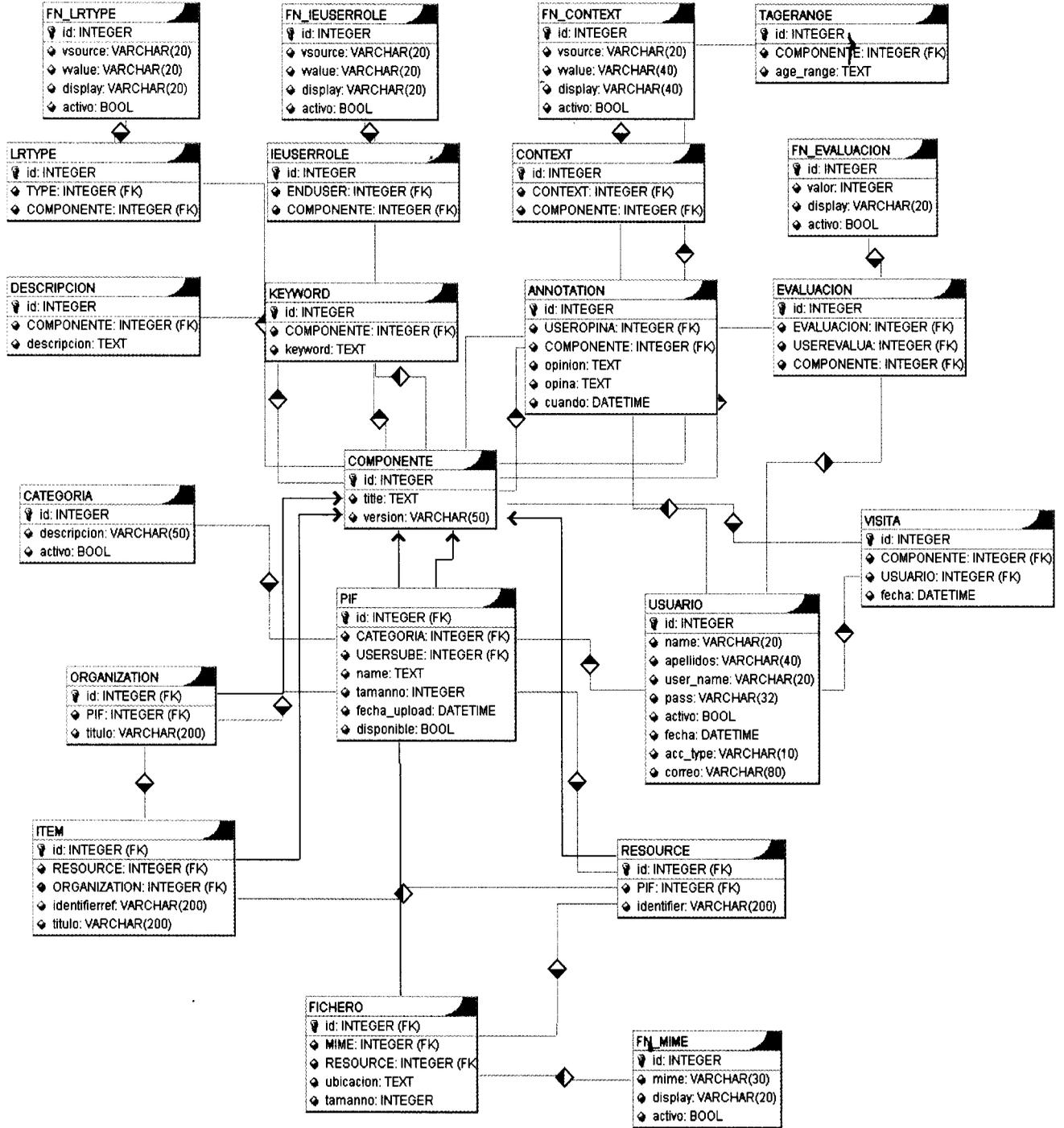
Caso de Uso Visualizar Opiniones Paquete



2.6.2 Diseño de la Base de Datos (DER)

El DER representa los 5 componentes del Modelo de Componentes de SCORM descritos en el Capítulo 1 y las relaciones entre ellos. Así un paquete de contenidos está formado por muchos recursos, que a su vez pueden estar compuestos de muchos ficheros. Por otra parte el contenido almacenado en el paquete puede ser estructurado de diversas formas para su entrega al usuario final a través del uso de organizaciones, las que están compuestas por actividades, que referencian a los recursos. Todos estos componentes pueden tener asociados metadatos, y una instancia de metadatos puede contener el título y la versión del mismo, muchas palabras claves, muchas descripciones, muchos usuarios finales y muchos contextos previstos para su uso, y muchos tipos de recursos educativos. Este diagrama expresa además que un usuario puede subir muchos paquetes, así como ofrecer criterios valorativos (cualitativos y cuantitativos) de los componentes.

DER



2.7 Tratamiento de errores

Un error en el sistema es un fallo que ocurre y que afecta al usuario. Para que los softwares tengan una mayor calidad estos errores deben tener algún tratamiento.

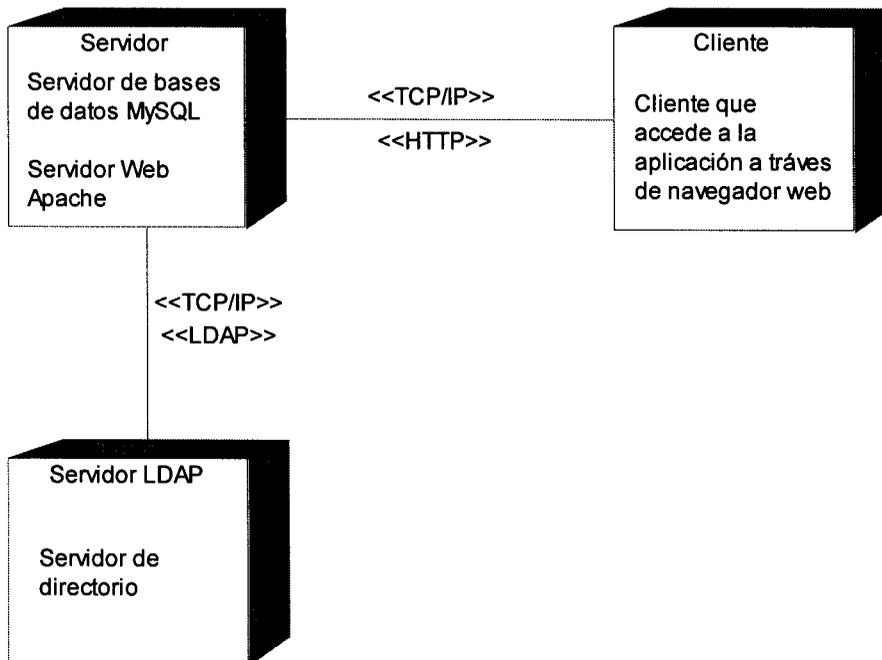
En esta aplicación pueden ocurrir dos tipos de errores. El primero, si hay un fallo en la funcionalidad del sistema, esto es, si ocurre algún error en las operaciones con la Base de Datos como en la inserción, actualización o eliminación de registros. El otro tipo de error ocurre cuando el usuario comete acciones que el software no espera.

Si hay algún error en las peticiones que se realizan a la Base de Datos y no se efectúan las acciones que el usuario desea, se capturan estos errores y se muestran al usuario para que tenga conocimiento del mismo. De la misma forma sucede si el usuario deja campos vacíos, si introduce datos no válidos, o si ejecuta alguna acción no válida en el sistema.

Todos estos errores son capturados y tratados fundamentalmente en el servidor, las validaciones en la parte del cliente son menos eficientes y seguras, incluso pudieran ser totalmente ignoradas si el navegador no es capaz de interpretar el código JavaScript, ya sea porque esta funcionalidad se encuentre deshabilitada o por razones de compatibilidad. La validación en el cliente implica además enviarle un volumen mayor de código que se traduce en un aumento del tiempo de carga de la página.

2.8 Implementación

2.8.1 Diagrama de Despliegue



Conclusiones del Capítulo

En este capítulo se mostraron las principales clases del dominio del problema para un mayor entendimiento del negocio y se describió la solución que se da al problema de investigación, se mostró una vista lógica y física de la aplicación así como los datos que se representan en ella.

Capítulo 3. Análisis de factibilidad y Resultados

La especificación de requisitos mediante Casos de Uso permite capturar la funcionalidad de un sistema. Este es una de las características fundamentales del Proceso Unificado de desarrollo, está dirigido por Casos de Uso. Sin embargo esto no permite efectuar una estimación del tamaño que tendrá el sistema o del esfuerzo que tomaría implementarlo. [14]

En este capítulo se expone un estudio de factibilidad mediante las normas establecidas en el modelo de estimación del esfuerzo basado en Casos de Uso, utilizándose el Análisis de Puntos de Casos de Uso. Se calcula el esfuerzo total del software, se exponen los beneficios tangibles e intangibles y se realiza un análisis costo-beneficio que demostrará si es factible o no este proyecto.

Se puede encontrar además una valoración de sostenibilidad del producto informático, donde se puede apreciar el impacto social y ambiental que tendrá la posterior implantación del sistema.

3.1 Estimación del esfuerzo.

“El método de estimación mediante el Análisis de Puntos de Casos de Uso es un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de pesos a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de estos factores.” [14]

Cálculo de Puntos de Caso de Uso sin Ajustar.

El primer paso para la estimación consiste en el cálculo de los Puntos de Casos de Uso sin ajustar. Este valor se calcula a partir de la siguiente ecuación:

$$UUCP = UAW + UUCW$$

Donde:

UUCP: Puntos de Casos de Uso sin ajustar

UAW: Factor de Peso de los Actores sin ajustar

UUCW: Factor de Peso de los Casos de Uso sin ajustar

Factor de Peso de los Actores sin ajustar (UAW).

Este valor se calcula mediante un análisis de la cantidad de Actores presentes en el sistema y la complejidad de cada uno de ellos.

Tipo Actor	Descripción	Peso	Actores	Total
Simple	Sistema con sistema a través de interfaz de programación.	1	0	0
Medio	Sistema con sistema mediante protocolo de interfaz basada en texto.	2	0	0
Complejo	Persona que interactúa con el sistema mediante interfaz gráfica.	3	2	6

Factor de Peso de los Casos de Uso sin ajustar (UUCW).

Este valor se calcula mediante un análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Casos de Uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo.

Tipo de CU	Descripción	Peso	Cantidad	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	10	50
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	2	20
Complejo	El caso de uso tiene más de 8 transacciones.	15	0	0

Entonces: **UUCP = UAW + UUCW = 6 + 70 = 76**

Cálculo de Puntos de Casos de Uso ajustados.

Una vez que se tienen los Puntos de Casos de Uso sin ajustar, se debe ajustar éste valor mediante la siguiente ecuación:

UCP = UUCP x TCF x EF

Donde:

UCP: Puntos de Casos de Uso ajustados

UUCP: Puntos de Casos de Uso sin ajustar

TCF: Factor de complejidad técnica

EF: Factor de ambiente

Factor de complejidad técnica (TCF).

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante.

El Factor de complejidad técnica se calcula mediante la siguiente ecuación:

$$TCF = 0.6 + 0.01 \times \Sigma (\text{Peso}_i \times \text{Valor asignado}_i)$$

Factor	Descripción	Peso	Valor	Total
T1	Sistema distribuido	2	0	0
T2	Tiempo de respuesta	1	1	1
T3	Eficiencia del usuario final	1	1	1
T4	Funcionamiento Interno complejo	1	3	3
T5	El código debe ser reutilizable	1	2	2
T6	Facilidad de instalación	0,5	0	0
T7	Facilidad de uso	0,5	5	2,5
T8	Portabilidad	2	1	2
T9	Facilidad de cambio	1	3	3
T10	Concurrencia	1	4	4
T11	Incluye objetivos especiales de seguridad	1	3	3
T12	Provee acceso directo a terceras partes	1	4	4
T13	Se requieren facilidades especiales de entrenamiento de usuarios	1	1	1

$$TCF = 0,865$$

Factor de ambiente (EF)

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo.

$$EF = 1.4 - 0.03 \times \sum (\text{Peso}_i \times \text{Valor asignado}_i)$$

Factor	Descripción	Peso	Valor	Total
E1	Familiaridad con el modelo de proyecto utilizado	1,5	4	6
E2	Experiencia en la aplicación	0,5	0	0
E3	Experiencia en la orientación a objetivos.	1	4	4
E4	Capacidad del analista líder.	0,5	4	2
E5	Motivación.	1	5	5
E6	Estabilidad de requerimientos	2	4	8
E7	Personal Part-Time	-1	0	0
E8	Dificultad del lenguaje de programación	-1	1	-1

$$EF = 0,68$$

$$\text{Entonces: UCP} = \text{UUCP} \times \text{TCF} \times \text{EF} = 76 \times 0,865 \times 0,68 = 47,7032$$

De los Puntos de Casos de Uso a la estimación del esfuerzo.

El esfuerzo en horas-hombre viene dado por:

$$E = \text{UCP} \times \text{CF}$$

Donde:

E: esfuerzo estimado en horas-hombre

UCP: Puntos de Casos de Uso ajustados

CF: factor de conversión

Para calcular CF

$$\text{CF} = 20 \text{ horas-hombre (si Total}_{EF} \leq 2)$$

$$\text{CF} = 28 \text{ horas-hombre (si Total}_{EF} = 3 \text{ ó Total}_{EF} = 4)$$

CF = abandonar o cambiar proyecto (si Total_{EF} ≥ 5)

Total_{EF} = Cant EF < 3 (entre E1 –E6) + Cant EF > 3 (entre E7, E8)

Total_{EF} = 1 + 0 = 1

CF = 20 horas-hombre

E = UCP x CF = 47,7032 x 20 = **894,064 horas-hombre**

Este método proporciona una estimación del esfuerzo en horas-hombre contemplando sólo el desarrollo de la funcionalidad especificada en los Casos de Uso. Para estimar la duración total del proyecto, se distribuye el esfuerzo entre las diferentes actividades del proyecto.

Actividad	Porcentaje %	Horas-Hombres
Análisis	10	223,516
Diseño	20	447,032
Implementación	40	894,064
Pruebas	15	335,274
Sobrecarga (otras actividades)	15	335,274
Total	100	2235,16

Teniendo en cuenta que el tiempo de trabajo planificado para un mes es de 240 horas, el tiempo de desarrollo se estima en aproximadamente 9 meses, como participan 2 personas, en aproximadamente 4 meses y medio se estima que finalice el desarrollo del proyecto.

Para la realización de este proyecto se requirió además de un extenso período de investigación, durante el cual se realizó una amplia revisión bibliográfica, debido a que fue necesario realizar un estudio de los estándares conforme a los cuales se implementaría el sistema. Además se hizo un estudio de algunas de las herramientas de autor para crear paquetes SCORM para su explotación, como Reload Editor y Weload.

3.2 Beneficios Tangibles e Intangibles.

Permitirá dotar a profesores y estudiantes de una herramienta que facilite el proceso de reutilización de los Objetos de Aprendizaje almacenados. De esta forma se disminuyen el esfuerzo, tiempo y costo para la generación de cursos y materiales didácticos en soporte electrónico, de buena calidad.

Al compartir y reutilizar se producen ahorros de recursos que se pueden redirigir hacia más desarrollo y se incrementa la disponibilidad de contenidos. La re-utilización del material, en diversos contextos, aumenta su valor.

Permite que la información esté centralizada en un lugar común donde se tiene un mejor acceso desde cualquier lugar donde se ponga en marcha el sistema.

Además, el uso efectivo de esta herramienta tendría gran repercusión como apoyo al proceso docente-educativo. Esto debido a que proporciona una serie de mecanismos que flexibilizan, en términos de tiempo y espacio, la interacción entre los implicados en el proceso.

3.3 Análisis Costo – Beneficio.

Los softwares que se utilizan para el desarrollo y puesta en práctica del sistema son de código abierto, por lo que no es necesario hacer gastos de licencia de software en su implementación ni para su implantación.

Para el desarrollo e implantación del sistema no es necesario adquirir equipamiento nuevo ya que todos están disponibles, se cuentan con los recursos técnicos necesarios de tipo tecnológico (hardware o software).

Si se tienen en cuenta los beneficios que se obtendrán al poner en marcha el sistema, en cuanto a calidad en la educación, ahorro de recursos, tiempo y esfuerzo el costo de desarrollar este software no es grande.

Se concluye que es factible el desarrollo del software.

3.4 Valoración de Sostenibilidad del Producto Informático.

Desde el punto de vista socio-cultural este proyecto aumenta el número de estudiantes y profesores que tienen acceso a los diferentes cursos, que tienen una mayor calidad, permitiendo una mejor preparación de ambos.

La reutilización de los contenidos de aprendizaje, contribuye a la construcción de materiales educativos de calidad con menos esfuerzo. Como los objetos de aprendizaje son fáciles de acceder y compartir, aumenta el valor del material educativo al ser reutilizado en diversos contextos.

Teniendo siempre presente el cuidado medioambiental, el sistema usa combinaciones de colores discretas, que no son dañinas a la vista del usuario y no produce ruidos que puedan dañar al ambiente, mostrando una interfaz agradable y fácil de usar por el usuario.

Todos los usuarios del sistema cuentan con todos los conocimientos necesarios para utilizarlo y se cuenta con toda la infraestructura electrónica necesaria para implantarlo.

3.5 Resultados

Se ha comprobado que la aplicación cumple con el estándar SCORM, pues es capaz de manipular paquetes creados con diferentes herramientas de autor como Reload Editor y Weload. Estos paquetes han sido incorporados a la plataforma Moodle y se comprobó que también son asumidos correctamente por ella.

Además la aplicación entiende los paquetes creados con la herramienta ROXS.

Conclusiones del Capítulo

En este Capítulo se estimó el esfuerzo total para la realización del sistema, concluyendo de este estudio el tiempo total de desarrollo, dado por la suma de los tiempos empleados en cada una de sus fases, teniendo en cuenta que participan dos personas.

Se señalaron los beneficios tangibles e intangibles de su puesta en marcha. Se realizó

un análisis costo-beneficio que permitió confirmar la factibilidad de la construcción del sistema propuesto.

Conclusiones

Con la realización de esta investigación se arriba a las siguientes conclusiones:

- ✓ El e-learning constituye un instrumento eficaz como forma de enseñanza que enriquece las capacidades de solución de problemas, autoaprendizaje y motivación de los estudiantes, rebasando límites geográficos y temporales.
- ✓ Existe en la actualidad la estandarización de los distintos elementos que componen al e-learning constituye una necesidad.
- ✓ Los repositorios de Objetos de Aprendizaje son herramientas que contribuyen a al desarrollo del e-learning.
- ✓ El Repositorio de Objetos de Aprendizaje facilita la reutilización de materiales educativos, aumentando el valor de estos al ser utilizados en distintos contextos, disminuyendo el costo de elaboración de los cursos.
- ✓ Es factible la realización de la aplicación.

Recomendaciones

Con vistas al desarrollo futuro de este proyecto se recomienda:

- ✓ Motivar a los profesores al uso de herramientas de autor para la creación de contenidos de calidad empaquetados acorde a estándares internacionales.
- ✓ Explicar a los creadores de estos contenidos la necesidad de describir los objetos de aprendizaje mediante el empleo de metadatos para facilitar su búsqueda y reutilización.
- ✓ Explicar a los creadores de estos contenidos la necesidad de granularlos para que puedan ser reutilizados en un mayor número de contextos.
- ✓ Utilizar el sistema propuesto como apoyo al proceso docente en la Universidad.
- ✓ Continuar desarrollando el sistema propuesto a partir de los nuevos requisitos que puedan surgir como resultado de su explotación.
- ✓ Ampliar el sistema utilizando los estándares de interoperabilidad entre repositorios.
- ✓ Añadir al sistema la posibilidad de syndicar noticias.
- ✓ Extender la funcionalidad del sistema para facilitar la búsqueda y uso de Objetos de Aprendizaje desde Moodle u otro LMS a través de una interfaz adaptada a sus características.
- ✓ Crear una biblioteca de clases útil para el desarrollo de nuevas aplicaciones para e-learning, basadas en tecnologías Web, o extensión de las existentes.

Referencias Bibliográficas

- [1] Badia, A.; Barberá, E.; Coll, C., et al. *La utilización de un material autosuficiente en un proceso de aprendizaje autodirigido*. Revista de Educación a Distancia (en línea) IV, No. III, 2005. [http://www.um.es/ead/red/1.\(02/03/2006\)](http://www.um.es/ead/red/1.(02/03/2006)).
- [2] Colectivo de Autores. *Estrategias para la docencia universitaria: La utilización pertinente de objetos de aprendizaje en el diseño de propuestas de enseñanza para la Web*. Universidad de Santiago de Compostela, 2005.
- [3] Colectivo de Autores. *Sirviendo Web desde la escuela*, 2003 (en línea) <http://es.tldp.org/Tutoriales/doc-servir-web-escuela/doc-servir-web-escuela-html/index.html> (03/04/2006).
- [4] De Bustos, J A.: *El servidor web Apache 1.3*, 2003 (en línea) http://www.augcyl.org/glol/old/N_1/apache.html (04/04/2006).
- [5] Deitel, H.: *XML. How to program*, Prentice Hall, 2001.
- [6] Del Castillo, A.: *El servidor de web Apache: Introducción práctica*, 2000 (en línea) <http://acs.barrapunto.org/articulos/trunk/LinuxActual/Apache/html/x31.html> (03/04/2006).
- [7] Learning Technology Standards Committee of IEEE, *Draft Standard for Learning Object Metadata*. USA, 2002.
- [8] Guerrero, L. A.: *Proceso Unificado de Desarrollo de Software*, 2005 (en línea) <http://www.dcc.uchile.cl/~luguerre/cc51h/clase23.html> (04/04/2006).
- [9] Hummel, H.; Manderveld, J.; Tattersall, C.; Koper, R.: *Educational modelling language and learning design: new opportunities for instructional reusability and personalized learning*. Int. J. Learning Technology 1, 2004
- [10] Jacobson, I.; Booch, G.; Rumbaugh J. *El proceso unificado de desarrollo de software I,*

[11] *Manual de PHP*, The PHP Documentation Group, 2005.

[12] Margalef, L. *Construcción de "objetos didácticos": buscando un marco de referencia desde la complejidad de entornos educativos*. Comunicación presentada en el SPDECE. I Simposio Pluridisciplinar sobre diseño, evaluación y descripción de contenidos educativos reutilizables, Guadalajara, 2004.

[13] Mauri, T.; Onrubia, J.; Coll, C. y Colomina, R. *La calidad de los contenidos educativos reutilizables: diseño, usabilidad y prácticas de uso*. Revista de Educación a Distancia, No. II, 2005. <http://www.um.es/ead/M2>.

[14] Peralta, M., Estimación del esfuerzo basado en casos de uso

[15] Rebollo, M. *El estándar SCORM para EaD*. Tesina del Máster en Enseñanza y Aprendizaje Abiertos y a Distancia de la Universidad Nacional de Educación a Distancia, España, diciembre, 2004.

[16] Rehak., D.; Mason, R.: *Keeping the Learning in Learning Objects'*. In Littlejohn, A. (ed.): *Journal of Interactive Media in Education, Special Issue on Reusing Online Resources 1*, 2003.

[17] Advanced Distributed Learning, *The SCORM Content Aggregation Model*, Advanced Distributed Learning Initiative, 2001. <http://coco.ccu.uniovi.es/e-learning/scorm/overview/organiza/ContentAM.pdf> (06/03/2006)

[18] Advanced Distributed Learning, *The SCORM Content Aggregation Model*, Advanced Distributed Learning Initiative, 2006, (en línea)
http://navyile.fedsun.navy.mil/netcile/content/SCORM_CAM.pdf, (03/06/2006)

[19] Varas, M. L., *Repositorio de objetos de aprendizaje*, 2005 (en línea)
http://www.alejandria.cl/recursos/documentos/documento_varas.doc (04/04/2006)

[20] Wiley, D. A., *Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy*, 2000.

Bibliografía

1. Advanced Distributed Learning. *The SCORM Overview*, 1 de octubre del 2001 (en línea) <http://coco.ccu.uniovi.es/e-learning/scorm/overview/organiza/Overview.pdf> (06/03/2006).
2. Álvarez, L. Gallardo, M. *Repositorio de Objetos de apoyo al aprendizaje colaborativo*, noviembre 2000. (en línea) http://www.tise.cl/archivos/tise2003/papers/repositorio_de_objetos.pdf (02/03/06).
3. Campbell, L. *UK Learning Object Metadata Core*, mayo del 2004 (en línea) http://www.cetis.ac.uk/profiles/uklomcore/uklomcore_v0p2_may04.doc (21/03/06).
4. Cid, A. Coss, J. *Herramienta de autor para la creación y gestión de objetos de aprendizaje reutilizable* Trabajo para optar por el título de ingeniero informático, Universidad de las Ciencias Informáticas, junio 2006.
5. Comité de Estandarización de Tecnologías Educativas de IEEE. *Estándares para Metadatos de Objetos Educativos*, 15 de julio del 2002 (en línea) <http://www.cenorm.be/cenorm/businessdomains/businessdomains/iss/iss/activity/loms/panish1.doc> (24/03/06).
6. Egea, J. *Estándares de e-Learning. Introducción a RELOAD*, noviembre del 2005 (en línea) <http://www.um.es/atiga/gat/tgm/reload/sesion1.pdf> (25/02/2006).
7. García, F. López, C., Pernías P., *Desarrollo de repositorios de objetos de aprendizaje a través de la reutilización de los metadatos de una colección digital: de Dublin Core a IMS*. RED. Revista de Educación a Distancia, número monográfico II, febrero del 2005. (en línea) http://spdece.uah.es/papers/Lopez_Final.pdf (02/03/06).
8. IMS Global Consortium. Inc, *IMS Content Packaging Best Practice and Implementation Guide*, octubre del 2004. (en línea),

- http://www.imsglobal.org/content/packaging/cpv1p1p4/imscp_bestv1p1p4.html
(25/02/2006).
9. IMS Global Consortium, Inc. *IMS Content Packaging XML Binding*, octubre del 2004, (en línea),
http://www.imsglobal.org/content/packaging/cpv1p1p4/imscp_bindv1p1p4.html
(25/02/2006).
10. IMS Global Consortium. Inc, *IMS Content Packaging Information Model*, octubre del 2004, (en línea),
http://www.imsglobal.org/content/packaging/cpv1p1p4/imscp_infov1p1p4.html
(25/02/2006).
11. IMS Global Consortium, Inc. *IMS Content Packaging Summary of Changes*, octubre del 2004, (en línea),
http://www.imsglobal.org/content/packaging/cpv1p1p4/imscp_sumcv1p1p4.html
(25/02/2006).
12. Iriarte, N., Marco, M., Morón, D., Pérez-Sacho, C., Pernías, P., *Diagramas Conceptuales y Objetos de Aprendizaje* (en línea),
http://spdece.uah.es/papers/Iriarte_1_Final.pdf (02/03/06)
13. Learning Technology Standards Committee of the IEEE. *Draft Standard for Learning Object Metadata*, 15 de julio del 2002, (en línea),
http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf (25/03/06).
14. Ostyn, C. *Cooking Up a SCORM*, (en línea),
http://www.elearningworkshops.com/docs/Autoria/CookingUpASCORM_v1_2_Draft_0_8.pdf, (15/03/06).
15. Solís, E. *Estándares en los sistemas de gestión de aprendizaje*, 2002
<http://spdece.uah.es/papers/estandares en los LMS.doc> (02/03/06).

16. Tamayo, D., Querejeta, A.,. *Plataforma para el desarrollo e-learning en la Universidad de Holguin "Oscar Lucero Moya"*. Trabajo para optar por el título de ingeniero informático, Universidad de Holguín, Holguín, junio 2005. }

Anexos

Anexo 1. 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 2. 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 3. Plataformas que siguen los estándares principales.

Plataforma	Descripción
ANGEL 6.1	Este software incorpora una certificación de estandarización con SCORM 1.2. Soporta la especificación IMS Enterprise 1.1. La empresa que lo proporciona soporta la migración entre los siguientes sistemas: BlackBoard o WebCT a ANGEL. Puede trabajar con la institución para migrar los cursos existentes en el nuevo sistema. LA propia aplicación incorpora herramientas que facilitan el paso de una versión a otra.
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.
Avilar WebMentor 4.0	Ha recibido una certificación de conformidad con SCORM 1.2.
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.
CentraOne 6.0	Es un software auto-testeado conforme con AICC AGR-010 (Web-based CMI Systems). EL fabricante afirma que también es compatible con SCORM 1.1 y 1.2 y con Microsoft LRN 2.0.
Click2learn Aspen 2.0	El fabricante garantiza el cumplimiento de SCORM 1.2
COSE 2.051	El sistema soporta los siguientes estándares: IMS Content Packaging 1.1.3 e IMS Metadata 1.2.2.
Desire2Learn 7.2	El fabricante garantiza que cumple la norma de SCORM 1.2 y soporta IMS Enterprise 1.1, IMS QTI 1.2, IMS Content Packaging Specification.
Desire2Learn 7.3	Ha recibido una certificación de conformidad con SCORM 1.2 RTE 3, y soporta IMS Enterprise 1.1, IMS QTI 1.2, e IMS Content Packaging Specification.

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.
Embanet hosting ANGEL	El fabricante soporta la migración entre las siguientes plataformas: Angel, BlackBoard, FirstClass, IntraLearn, Prometheus y WebCT. El sistema puede importar y exportar contenidos usando IMS Content Packaging standard.
Embanet hosting BlackBoard	El fabricante soporta la migración entre las siguientes plataformas: Angel, BlackBoard, FirstClass, IntraLearn, Prometheus, and WebCT. El sistema admite 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 los contenidos entre versiones diferentes del software.
Embanet hosting FirstClass	El fabricante soporta la migración entre las siguientes plataformas: Angel, BlackBoard, FirstClass, Prometheus IntraLearn, and WebCT.
Embanet hosting IntraLearn	El fabricante soporta la migración entre las siguientes plataformas: Angel, BlackBoard, FirstClass, Prometheus IntraLearn, and WebCT. El fabricante ha chequeado el software y garantiza que es compatible con IMS metadata specification 1.2.2, y IMS Content Packaging 1.1.3. Puede importar contenidos que cumplan AICC, IMS, o LRN.
Embanet hosting WebCT	El fabricante soporta la migración entre las siguientes plataformas: Angel, BlackBoard, FirstClass, Prometheus IntraLearn, and WebCT. El sistema puede importar y exportar contenidos usando IMS Content Packaging standard.
IntraLearn SME 3.1.2	El fabricante ha chequeado el software y garantiza que cumple la IMS metadata specification 1.2.2, y IMS Content Packaging 1.1.3. Puede importar contenidos de cursos que cumplan AICC, IMS, o LRN.
Janison Toolbox 6.2	Puede importar y exportar cursos y objetos utilizando manifiestos. Soporta la inclusión de objetos de SCORM.
Jones e-education V2004	Permite importar contenidos de cursos que satisfagan IMS.

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 1.4	Soporta la utilización de objetos SCORM. Incluye herramientas que facilitan la migración de cursos entre versiones distintas.
Teknical Virtual Campus	El fabricante garantiza que cumple IMS Metadata 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
TeleTop	Describe todos los contenidos de sus cursos usando especificaciones SCORM 1.1 y puede importar cursos y recursos que empleen manifiestos IMS.
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 4. Requerimientos de los elementos del manifiesto según el Perfil de Aplicación del paquete de contenido de SCORM.

Leyenda

M: Mandatory (Obligatorio)

O: Optional (Opcional)

NP: Not Permitted (No Permitido)

No.	Elementos/atributos	Paquete de recursos	Paquete de agregación de contenidos
1	<manifest>	M	M
1.1	identifier	M	M
1.2	version	O	O
1.3	xml:base	O	O
1.4	<metadata>	M	M
1.4.1	<schema>	M	M
1.4.2	<schemaversion>	M	M
1.4.3	{Metadata}	O	O
1.5	<organizations>	M	M
1.5.1	default	NP	M
1.5.2	<organization>	NP	M
1.5.2.1	identifier	NP	M
1.5.2.2	structure	NP	O
1.5.2.3	adlseq:objectivesGlobalToSystem	NP	O
1.5.2.4	<title>	NP	M
1.5.2.5	<item>	NP	M
1.5.2.5.1	identifier	NP	M
1.5.2.5.2	identifieref	NP	O
1.5.2.5.3	<title>	NP	M
1.5.2.5.4	isvisible	NP	O
1.5.2.5.5	parameters	NP	O
1.5.2.5.6	<item>	NP	O
1.5.2.5.7	<metadata>	NP	O
1.5.2.5.7.1	{Metadata}	NP	O
1.5.2.5.8	<adlcp:timeLimitAction>	NP	O
1.5.2.5.9	<adlcp:dataFromLMS>	NP	O
1.5.2.5.10	<adlcp:completionThreshold>	NP	O
1.5.2.5.11	<imsss:sequencing>	NP	O
1.5.2.5.12	<adlnav:presentation>	NP	O
1.5.2.6	<metadata>	NP	O
1.5.2.6.1	{Metadata}	NP	O
1.5.2.7	<imsss:sequencing>	NP	O
1.6	<resources>	M	M
1.6.1	xml:base	O	O
1.6.2	<resource>	O	O

1.6.2.1	identifier	M	M
1.6.2.2	type	M	M
1.6.2.3	href	O	O
1.6.2.4	adlcp:scormType	M	M
1.6.2.5	xml:base	O	Q
1.6.2.6	<metadata>	O	O
1.6.2.6.1	{Metadata}	O	O
1.6.2.7	<file>	O	O
1.6.2.7.1	href	M	M
1.6.2.7.2	<metadata>	O	O
1.6.2.7.2.1	{Metadata}	O	O
1.6.2.8	<dependency>	O	O
1.6.2.8.1	identifierref	M	M
1.7	<manifest>	O	O
1.8	<imsss:sequencingCollection>	NP	O

Anexo 5. Requerimientos de los elementos según el Perfil de Aplicación de los metadatos de SCORM.

Leyenda

M: Mandatory (Obligatorio)

O: Optional (Opcional)

No.	Elemento	Requerimiento
1	<general>	M
1.1	<identifier>	O
1.1.1	<catalog>	O
1.1.2	<entry>	O
1.2	<title>	M
1.3	<language>	O
1.4	<description>	M
1.5	<keyword>	M
1.6	<coverage>	O
1.7	<structure>	O
1.8	<aggregationLevel>	O
Elemento		Requerimiento
2	<lifeCycle>	M
2.1	<version>	M
2.2	<status>	O
2.3	<contribute>	O
2.3.1	<role>	O
2.3.2	<entity>	O
2.3.3	<date>	O
Elemento		Requerimiento
3	<metaMetadata>	M
3.1	<identifier>	O
3.1.1	<catalog>	O
3.1.2	<entry>	O
3.2	<contribute>	O
3.2.1	<role>	O
3.2.2	<entity>	O
3.2.3	<date>	O
3.3	<metadataSchema>	M
3.4	<language>	O
Elemento		Requerimiento
4	<technical>	M
4.1	<format>	M

4.2	<size>	O
4.3	<location>	O
4.4	<requirement>	O
4.4.1	<orComposite>	O
4.4.1.1	<type>	O
4.4.1.2	<name>	O
4.4.1.3	<minimumVersion>	O
4.4.1.4	<maximumVersion>	O
4.5	<installationRemarks>	O
4.6	<otherPlatformRequirements>	O
4.7	<duration>	O
Elemento		Requerimiento
5	<educational>	O
5.1	<interactivityType>	O
5.2	<learningResourceType>	O
5.3	<interactivityLevel>	O
5.4	<semanticDensity>	O
5.5	<intendedEndUserRole>	O
5.6	<context>	O
5.7	<typicalAgeRange>	O
5.8	<difficulty>	O
5.9	<typicalLearningTime>	O
5.10	<description>	O
5.11	<language>	O
Elemento		Requerimiento
6	<rights>	M
6.1	<cost>	O
6.2	<copyrightAndOtherRestrictions>	M
6.3	<description>	O
Elemento		Requerimiento
7	<relation>	O
7.1	<kind>	O
7.2	<resource>	O
7.2.1	<identifier>	O
7.2.1.1	<catalog>	O
7.2.1.2	<entry>	O
7.2.2	<description>	O
Elemento		Requerimiento
8	<annotation>	O
8.1	<entity>	O
8.2	<date>	O
8.3	<description>	O

Elemento		Requerimiento
9	<classification>	0
9.1	<purpose>	0
9.2	<taxonPath>	0
9.2.1	<source>	0
9.2.2	<taxon>	0
9.2.2.1	<id>	0
9.2.2.2	<entry>	0
9.3	<description>	0
9.4	<keyword>	0