

**PLATAFORMA PARA EL INTERCAMBIO DE MATERIALES
EDUCATIVOS REUTILIZABLES ENTRE AMBIENTES VIRTUALES
DE APRENDIZAJE**

Ing. ERWIN MEZA VEGA

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
MAESTRÍA EN INGENIERÍA
BUCARAMANGA
2005**

**PLATAFORMA PARA EL INTERCAMBIO DE MATERIALES
EDUCATIVOS REUTILIZABLES ENTRE AMBIENTES VIRTUALES
DE APRENDIZAJE**

Ing. ERWIN MEZA VEGA

**Investigación presentada como requisito parcial para optar al
título de Magíster en Ingeniería – Área Ciencias de la Computación**

Directora del proyecto:

**MARTHA VITALIA CORREDOR MONTAGUT
Doctora Ingeniera en Telecomunicaciones**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
MAESTRÍA EN INGENIERÍA
BUCARAMANGA
2005**

DEDICATORIA

A Carolina, por su amor, comprensión
... y paciencia.

AGRADECIMIENTOS

El desarrollo y la llegada a feliz término de este trabajo no hubieran sido posibles sin la valiosa colaboración de innumerables profesores, colegas y amigos, que de una u otra forma contribuyeron para que una idea inicial se transformara en un proyecto tangible. El autor desea expresar sus más sinceros agradecimientos a:

Martha Vitalia Corredor, por su sabio consejo, su apoyo y confianza incondicional, y por ser un ejemplo de obra y vida.

A los integrantes del Grupo GENTE de la Universidad Industrial de Santander, especialmente a Claude Ewert, Esperanza Aguilar y Jorge Ramón, por su apoyo y disposición a explorar las posibilidades que ofrece el uso de las nuevas tecnologías en la educación.

A los integrantes del Grupo de I+D en Tecnologías de la Información de la Universidad del Cauca, en especial a Carlos Cobos, Martha Eliana Mendoza y Miguel Niño, por sus aportes al desarrollo de este trabajo, y por su apoyo incondicional.

Y finalmente a todas las personas que ofrecieron su ayuda en el momento más oportuno, entre ellas a Carlos Rodríguez, Juan Carlos Escobar, María Eugenia, y a Carlos Jaime, por su apoyo desde la distancia.

CONTENIDO

INTRODUCCIÓN	19
1. PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN	25
2. OBJETIVOS	28
2.1. OBJETIVO GENERAL	28
2.2. OBJETIVOS ESPECÍFICOS	28
3. MARCO TEÓRICO	29
3.1. AMBIENTES VIRTUALES DE APRENDIZAJE	29
3.2. SISTEMAS DE GESTIÓN DE APRENDIZAJE – LMS	31
3.2.1. Los LMS desde el punto de vista de los servicios	33
3.2.2. Los LMS desde el punto de vista de los usuarios	36
3.2.3. Herramientas asociadas con los LMS	38
3.3. OBJETOS DE APRENDIZAJE	43
3.3.1. Definición de objeto de aprendizaje	44
3.3.2. Motivación por el estudio y uso de los objetos de aprendizaje	46
3.3.3. Clasificación de los objetos de aprendizaje	46
3.4. ESTÁNDARES PARA LA EDUCACIÓN EN LÍNEA	47
3.4.1. AICC, Aviation Industry CBT Committee	48
3.4.2. Institute for Electrical and Electronic Engineers Learning Technology Standards Committee (IEEE LTSC)	51
3.4.3. IMS Global Learning Consortium, Inc.	52
3.4.4. ADL SCORM	54
3.5. REPOSITORIOS DIGITALES DE OBJETOS DE APRENDIZAJE	70
3.5.1. Definición de repositorio digital	71

3.5.2. Iniciativas de repositorios digitales	72
3.5.3. Interoperabilidad de repositorios digitales	75
3.5.4. Modelos de interoperabilidad de repositorios digitales de contenidos	78
3.6. DESARROLLO DE APLICACIONES BASADAS EN SERVICIOS WEB XML	86
3.6.1. Arquitectura de un servicio Web	87
4. CREACIÓN DE MATERIALES EDUCATIVOS BASADOS EN OBJETOS DE APRENDIZAJE	96
4.1. SELECCIÓN DE UN ESTÁNDAR PARA LA CREACIÓN DE CONTENIDOS BASADOS EN OBJETOS DE APRENDIZAJE	97
4.2. ESTRATEGIA PARA LA CREACIÓN DE CONTENIDOS BASADOS EN OBJETOS DE APRENDIZAJE	99
4.2.1. Fase 0. Conformación del equipo de trabajo	100
4.2.2. Fase 1. Análisis del contexto de la experiencia de aprendizaje	101
4.2.3. Fase 2. Diseño de la estructura de contenidos y caracterización de los objetos de aprendizaje	102
4.2.4. Fase 3. Reutilización y construcción de los objetos de aprendizaje	107
4.2.5. Fase 4. Catalogación y almacenamiento de los objetos de aprendizaje	110
4.2.6. Fase 5. Agregación de los objetos de aprendizaje	112
4.2.7. Fase 6. Catalogación, almacenamiento y distribución de los paquetes de contenido	114
4.2.8. Fase 7. Evaluación del proceso	116
5. DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA PARA EL ALMACENAMIENTO, LA ADMINISTRACIÓN, LA BÚSQUEDA Y LA DISTRIBUCIÓN DE OBJETOS DE APRENDIZAJE A TRAVÉS DE INTERNET	118
5.1. ESPECIFICACIÓN DE CASOS DE USO PARA LA PLATAFORMA DE INTERCAMBIO DE OBJETOS DE APRENDIZAJE	119

5.1.1. Requerimientos y restricciones de la plataforma	119
5.1.2. Modelos de casos de uso	120
5.1.3. Modelo general de los paquetes del sistema	125
5.2. PROTOCOLO DE COMUNICACIÓN PARA EL ALMACENAMIENTO Y LA GESTIÓN DE CONTENIDOS	130
5.2.1. Modelo de comunicación	130
5.2.2. Estructura de los mensajes	131
5.2.3. Identificación de usuarios	132
5.2.4. Definición de las funciones que ofrece el protocolo de comunicación	133
5.3. ARQUITECTURA DE LA PLATAFORMA DE INTERCAMBIO DE OBJETOS DE APRENDIZAJE	150
5.3.1. Roles	151
5.3.2. Interfaz de servicios de la plataforma	153
5.3.3. Componentes de LMS	165
5.3.4. Componentes de repositorio	178
CONCLUSIONES Y RECOMENDACIONES	186
BIBLIOGRAFÍA	188
ANEXOS	191
ANEXO 1. ESQUEMA DE METADATOS LOM	192
ANEXO 2. MODELO DE DATOS DE AULA VIRTUAL 2.0	200
ANEXO 3. MODELO DE DATOS DEL REPOSITORIO DE OBJETOS DE APRENDIZAJE	214
ANEXO 4. DOCUMENTO DE DEFINICIÓN DE LOS SERVICIOS DE INTERCAMBIO DE LA PLATAFORMA	220

LISTA DE TABLAS

Tabla 1. Grupos de trabajo que conforman el LTSC	52
Tabla 2. Iniciativas del IMS Global Learning Consortium	53
Tabla 3. Modos de secuenciación definidos por SCORM SN	67
Tabla 4. Condiciones de una regla de secuenciación	68
Tabla 5. Acciones para una regla de secuenciación	70
Tabla 6. Servicios ofrecidos por la implementación de ECL	83
Tabla 7. Roles que conforman del equipo de trabajo	100
Tabla 8. Estructura de un concepto	104
Tabla 9. Estructura de un hecho.	104
Tabla 10. Estructura de un proceso.	105
Tabla 11. Estructura de un procedimiento.	105
Tabla 12. Estructura de un principio.	106
Tabla 13. Formato para la recolección de los metadatos LOM de un objeto de aprendizaje	111
Tabla 14. Errores que se pueden presentar en la adición de un objeto de aprendizaje	140
Tabla 15. Códigos de error para la función getMetadata	142

LISTA DE FIGURAS

Figura 1. Estructura de un paquete de contenidos SCORM	57
Figura 2. Estructura de una organización de contenidos	58
Figura 3. Modelo del ambiente de ejecución de SCORM	62
Figura 4. Esquema de una actividad de aprendizaje	64
Figura 5. Estructura de un árbol de actividades	65
Figura 6. Estructura de clusters de un árbol de actividades	66
Figura 7. Condiciones de una regla de secuenciación	68
Figura 8. Acciones definidas para una regla de secuenciación	69
Figura 9. Arquitectura funcional definida en DRI	79
Figura 10. IMS DRI - Modelo Search/Expose	80
Figura 11. IMS DRI - Modelo Gather/Expose	80
Figura 12. IMS DRI - Modelo Submit/Store	81
Figura 13. Modelo de federación de repositorios de CORDRA	85
Figura 14. Arquitectura de un servicio Web	88
Figura 15. Modelo extendido de un servicio Web	89
Figura 16. Estructura de un mensaje SOAP	91
Figura 17. Estructura de un documento WSDL	93
Figura 18. Estructura resumida del documento WSDL	95
Figura 19. Evolución de los estándares y especificaciones de educación en línea.	97
Figura 20. Jerarquía de contenidos propuesta por Cisco	103
Figura 21. Niveles cognitivos de acuerdo con Merrill y Bloom	107
Figura 22. Interfaz de búsqueda de objetos de aprendizaje	108
Figura 23. RELOAD: Herramienta para la creación de estructuras de contenidos	113

Figura 24. Módulo de Aula Virtual para crear materiales que cumplen con SCORM	114
Figura 25. Interfaz que permite agregar un recurso a un repositorio	115
Figura 26. Interfaz ofrecida por el repositorio para la edición de metadatos	116
Figura 27. Modelo general de actores la plataforma	120
Figura 28. Diagrama general de casos de uso de la plataforma	122
Figura 29. Diagrama de casos de uso crear de objetos de aprendizaje	123
Figura 30. Diagrama de secuencia del caso de uso buscar objeto de aprendizaje	124
Figura 31. Diagrama de secuencia para la adición de objetos y/o metadatos	125
Figura 32. Modelo general de los paquetes de la plataforma	126
Figura 33. Estructura de almacenamiento de los recursos	127
Figura 34. Modelo de comunicación entre los usuarios y los proveedores de servicios	131
Figura 35. Arquitectura de la plataforma de intercambio de objetos de aprendizaje	151
Figura 36. Interfaz de edición de una estructura de contenidos	167
Figura 37. Selección de los criterios de búsqueda de recursos	168
Figura 38. Interfaz para buscar y localizar objetos de aprendizaje	168
Figura 39. Interfaz de administración de repositorios de Aula Virtual	169
Figura 40. Interfaz para la suscripción a los repositorios	170
Figura 41. Interfaz de registro en un repositorio	171
Figura 42. Interfaz de búsqueda de objetos de aprendizaje almacenados en diferentes repositorios	171
Figura 43. Interfaz para adicionar un nuevo recurso a un repositorio	172
Figura 44. Ambiente de ejecución SCORM	173

Figura 45. Ejecución de un SCO	174
Figura 46. Mensajes emitidos por la implementación del API	176
Figura 47. Envío de datos desde SCO hacia el LMS	177
Figura 48. Fin de la ejecución de un SCO	177
Figura 49. Página de entrada a la interfaz web del repositorio	178
Figura 50. Interfaz de suscripción al repositorio	179
Figura 51. Interfaz de gestión de recursos	180
Figura 52. Interfaz de adición de un nuevo recurso	181
Figura 53. Interfaz para la edición de metadatos de un objeto de aprendizaje	182
Figura 54. Interfaz de búsqueda de objetos de aprendizaje	183
Figura 55. Resultados de la búsqueda de recursos	184
Figura 56. Herramienta para visualizar agregaciones de contenidos SCORM	185

LISTA DE FRAGMENTOS DE CÓDIGO

Fragmento de código 1. Esquema de una solicitud SOAP	90
Fragmento de código 2. Mensaje de respuesta a la solicitud SOAP	91
Fragmento de código 3. Ejemplo de un documento WSDL	94
Fragmento de código 4. Mensaje SOAP que incluye la llave de usuario	133
Fragmento de código 5. Resultado de la llamada a la función getReferenceModelList	136
Fragmento de código 6. Resultado del llamado a la función GetComponentList	137
Fragmento de código 7. Resultado de la función getMetadata	141
Fragmento de código 8. Resultado de la función metadataTextQuery	144
Fragmento de código 9. Resultado de la función metadataQueryXML	146
Fragmento de código 10. Llamada al servicio web de suscripción	154
Fragmento de código 11. Verificación de la conexión con un repositorio	155
Fragmento de código 12. Obtener el listado de los modelos de referencia disponibles en un repositorio	156
Fragmento de código 13. Resultados de la llamada obtener modelos de referencia	156
Fragmento de código 14. Obtener el listado de componentes para un modelo de referencia dado	157
Fragmento de código 15. Listado de los componentes de un modelo de referencia	157
Fragmento de código 16. Validación de un componente del modelo de referencia SCORM	158
Fragmento de código 17. Llamada al método para cargar los metadatos de un objeto de aprendizaje	159

Fragmento de código 18. Llamada al método para descargar los metadatos de un recurso	160
Fragmento de código 19. Invocación al servicio de búsqueda (método 1)	161
Fragmento de código 20. Formato de objetos del resultado del método 1 de búsqueda	161
Fragmento de código 21. Invocación al servicio de búsqueda (método 2)	162
Fragmento de código 22. Resultados de la búsqueda de recursos (método 2)	162
Fragmento de código 23. Consulta de los registros de metadatos para recursos propios	163
Fragmento de código 24. Llamada al método para adicionar un nuevo recurso	163
Fragmento de código 25. Obtener la información general de un recurso	164
Fragmento de código 26. Llamada al método para actualizar la información general de un recurso	165
Fragmento de código 27. Rutina de inicialización del SCO	175

LISTA DE ANEXOS

Anexo 1. Esquema de metadatos LOM	192
Anexo 2. Modelo de datos de Aula Virtual 2.0	200
Anexo 3. Modelo de datos del Repositorio de Objetos de Aprendizaje	214
Anexo 4. Documento de definición de los servicios de intercambio de la plataforma	220

RESUMEN

TITULO: PLATAFORMA PARA EL INTERCAMBIO DE MATERIALES EDUCATIVOS REUTILIZABLES ENTRE AMBIENTES VIRTUALES DE APRENDIZAJE*

AUTOR: ERWIN MEZA VEGA**

PALABRAS CLAVE: Ambientes virtuales de aprendizaje, repositorios de contenidos, objetos de aprendizaje

DESCRIPCIÓN:

En el presente proyecto se propone una plataforma para el intercambio de materiales educativos, que permite a los Ambientes Virtuales de Aprendizaje empaquetar y enviar objetos de contenido y agregaciones de contenidos a un repositorio digital, y además ofrece los medios para buscar, localizar y visualizar los recursos almacenados en diferentes repositorios, sin importar la versión del esquema de metadatos utilizado para la descripción de sus recursos.

Igualmente, se llevó a cabo el desarrollo de un repositorio de objetos de contenido, con base en la Especificación para la Interoperabilidad de Repositorios Digitales (IMS DRI). Este repositorio fue diseñado inicialmente para soportar el esquema de metadatos IEEE LOM, pero el modelo de datos subyacente puede ser utilizado para almacenar cualquier esquema de metadatos basado en XML. Con el fin de permitir la comunicación entre diferentes implementaciones de Ambientes Virtuales de aprendizaje y repositorios digitales, se diseñó e implementó un protocolo de comunicación, que consiste en una serie de Servicios Web basados en el Protocolo de Acceso Simple a Objetos (SOAP).

Además de las herramientas desarrolladas, se propone una estrategia para la creación de materiales educativos basados en objetos de aprendizaje, la cual presenta los pasos generales que se deben seguir en el proceso de desarrollar materiales que cumplan con el modelo de referencia SCORM. La implementación de esta estrategia y el uso de las herramientas desarrolladas permitirán obtener un mayor conocimiento de los estándares, las especificaciones y las herramientas tecnológicas que se utilizan en la educación en línea.

* Trabajo de investigación

** Facultad: Facultad de Ingenierías Físico – Mecánicas

Programa de postgrado: Maestría en Ingeniería, Área de Informática y Ciencias de la Computación.

Director: Dra. Martha Vitalia Corredor Montagut

SUMMARY

TITLE: PLATFORM FOR THE EXCHANGE OF REUSABLE EDUCATIONAL CONTENT AMONG VIRTUAL LEARNING ENVIRONMENTS*

AUTHOR: ERWIN MEZA VEGA**

KEYWORDS: Virtual Learning Environments, Digital Repositories, Learning Objects

DESCRIPTION:

In this Project a platform for the exchange of educational content is proposed, which allows to the Virtual Learning Environments to pack and to submit content objects and content aggregations to a digital repository, and it also offers the means to search, to locate and to visualize the resources stored in different repositories, regardless of the metadata schema used for the description of their resources.

Likewise, the development of a content object repository which is based on the Digital Repositories Interoperability Specification (IMS DRI) was carried out. This repository was initially designed to support the IEEE LOM metadata schema, but the underlying data model can be used to store any XML metadata schema. In order to allow the communication among different implementations of Virtual Learning Environments and digital repositories, a communication protocol was designed and developed. This protocol consists of a series of Web Services, and it is based on the Simple Object Access Protocol (SOAP).

Besides the developed tools, a strategy for the creation of educational materials based on learning objects is proposed. This strategy presents the general steps to follow in the process of developing educational materials that comply with the SCORM reference model. The implementation of this strategy and the use of the developed tools will allow obtaining a better knowledge about the standards, the specifications and the technological tools that are used in on-line education.

* Research work

** Faculty: Physical and Mechanical Engineering
Postgraduate program: Master in Engineering, Systems and Computer Science
Director: Dr. Martha Vitalia Corredor Montagut

INTRODUCCIÓN

Es reconocido que las Tecnologías de la Información y la Comunicación (TIC) están influyendo en todos los campos, y la educación no es ajena a esta influencia, que amplía la gama de las estrategias y moderniza los procesos de enseñanza y aprendizaje. Los sistemas públicos de correo domiciliario hicieron posible el nacimiento de los programas de aprendizaje a distancia desde comienzos del siglo XX en Estados Unidos y la Unión Soviética, mientras que la radio y la televisión hicieron posible a mediados del siglo XX extender estas experiencias de manera casi interactiva. La masificación del acceso a la educación recurrió en todos los continentes al uso de los medios masivos de (radio, cine, televisión) para favorecer la enseñanza a distancia y el autoaprendizaje desde hace unos cincuenta años.

La difusión de las computadoras personales a partir de los años 80 y de Internet a partir de los 90, junto con la aparición de poderosas redes de información a escala mundial vino a modificar sustancialmente esta evolución. En la última década Colombia ha estado conectada al mundo a través de Internet. Las primeras comunidades beneficiadas fueron los profesores y estudiantes de las universidades colombianas, quienes a partir de esta conexión empezaron a fortalecer sus posibilidades de información y de intercambio con los pares académicos, así como con los grupos que dinamizan y realizan una actividad de investigación en los temas propios de cada una de las disciplinas que se trabajan en las universidades. Estas posibilidades de las telecomunicaciones han permitido que el rango de acción de los académicos sea el mundo; sin embargo, no han generado una dinámica de uso de los servicios de las redes como apoyo a los procesos de aprendizaje que se dan principalmente en los programas de formación profesional.

La dinámica que generó la conexión de universidades a la gran red internacional se extendió a diferentes empresas, instituciones gubernamentales, escuelas, sitios de diversión y en los últimos años a los hogares. Este fenómeno obedece al desarrollo de la informática en combinación con las telecomunicaciones y a la explosión de servicios ofrecidos a través de Internet, un acontecimiento irreversible a pesar de las

diferentes críticas de los guardianes de la tradición. La decisión de no trabajar con las nuevas tecnologías puede llevar a la desaparición de algunas universidades, o, por lo menos, a la pérdida de algún “buen nombre y calidad educativa” adquiridos, ya que muchas instituciones extranjeras están entrando en el entorno colombiano de la educación, ofreciendo cursos y programas de mejor calidad, de manera más interactiva, sin necesidad para el estudiante de dejar su empleo durante dos o más años. Ya se puede apreciar el impacto que tiene el Instituto Tecnológico de Monterrey de México (ITESM), con sus maestrías satelitales ofrecidas en varios sitios. Las universidades de Harvard, Stanford, Berkeley, Penn State, el Instituto Tecnológico de Massachussets están ofreciendo numerosos programas de maestrías a distancia con el uso de tecnologías en países como Rusia, China, Tailandia, Indonesia, y están explorando posibilidades en Latinoamérica.

Por otra parte, las dos últimas décadas se han caracterizado por grandes y rápidos cambios en lo referente a la Informática y las Telecomunicaciones, lo que ha generado la necesidad del planteamiento de diseños nuevos y flexibles en los entornos que se ofrezcan para apoyar los procesos de enseñanza y aprendizaje. En los países con un desarrollo similar al de Colombia, existen aún resistencias al uso de las TIC como herramientas de apoyo en procesos educativos, motivadas principalmente por una carencia de infraestructura física y tecnológica, así como por el desconocimiento, la falta de capacitación y calificación de los docentes en el uso de estas modernas herramientas y sus implicaciones en educación.

En países como Estados Unidos, Canadá, Francia, Alemania, Gran Bretaña, Japón, a pesar de presentarse también resistencia, el uso de las TIC como apoyo a la educación se ha ido incrementando principalmente en cursos de Educación Abierta y a Distancia, para ofrecer una alternativa que permita llevar más estudiantes a los ambientes universitarios, en los que falten salones, laboratorios y espacios para bienestar estudiantil. El MIT (Massachussets Institute of Technology) señaló en la conferencia TELECON XVI (Anaheim, CA, Octubre 1996) que hacia los años 2000-2002, el MIT tendría 50% de su matrícula con estudiantes a distancia de diferentes países conectados a través de los servicios de redes de telecomunicación; Penn

State University tiene estudiantes en una localidad situada a mas de 20 horas en tren de Moscú, utilizando para apoyar el aprendizaje el video y la internet, con algunas interacciones por audio conferencia; en el ámbito local, la Universidad Autónoma de Bucaramanga (UNAB) está trabajando por satélite con el ITESM (Instituto Tecnológico y de Educación Superior de Monterrey), diferentes programas de maestría, con interacciones mediante videoconferencias, fax, teléfono e internet.

A nivel de grandes empresas, para apoyar el entrenamiento y la capacitación, existen diferentes programas ofrecidos mediante el uso de las TIC. Como ejemplos tenemos BOEING, la NASA, IBM, la FORD. La NTU (National Technology University) está ofreciendo los cursos de 43 universidades importantes a más de 207 empresas utilizando la televisión por satélite, actualmente ofrece 13 maestrías en ingeniería totalmente a distancia. La universidad de los Andes de Santa fe de Bogotá ha estado en contacto con la NTU para traer algunos programas a Colombia.

También es necesario reconocer en Colombia las investigaciones y los desarrollos realizados sobre el uso de las TIC para apoyar procesos de enseñanza y aprendizaje en los niveles de Educación Básica, Media y Superior. Es importante destacar algunos de los grupos de Investigación en Informática Educativa en Colombia: el de la Universidad de los Andes que desarrolla trabajos en Uniandes, con la Secretaría de Educación del Distrito en Santafé de Bogotá y con el Ministerio de Educación Nacional; el de la Universidad EAFIT de Medellín que desarrolla trabajos en la universidad y con la Secretaría de Educación Departamental, con su proyecto bandera Conexiones; el de la Universidad del Norte de Barranquilla que apoya trabajos de informática educativa en Uninorte y en colegios de educación Básica de su ciudad; pueden mencionarse también los grupos de la Universidad Pedagógica Nacional, el de la Universidad Nacional, la Universidad de Manizales, la Universidad Autónoma de Bucaramanga y el de la Universidad de Antioquia que adelantan investigaciones sobre las posibilidades de las TIC como herramientas de apoyo en los procesos educativos. Igualmente la Universidad del Cauca con el desarrollo de los proyectos “Unicauca Virtual – Fase 1” y “Unicauca Virtual – Fase 2”

busca analizar y estudiar diversos modelos educativos utilizados en la educación en línea, con el fin de obtener un *meta-modelo educativo*, sobre el cual se está desarrollando un sistema de gestión de aprendizaje que podrá ser utilizado por educaciones de educación media y superior. Finalmente la Universidad Industrial de Santander a través del proyecto “Escenario Pedagógico para la Enseñanza Virtual en la Universidad Industrial de Santander” busca construir un escenario que permita ofrecer experiencias de enseñanza en línea en la UIS, con el fin de enriquecer los procesos de aprendizaje en el desarrollo de cursos presenciales, utilizando la plataforma Aula Virtual, que se desarrolló dentro del trabajo de investigación “Ambiente Virtual para el apoyo de aprendizajes colaborativos en un modelo pedagógico fundamentado en la resolución de problemas”, desarrollado en el 2002 por el Grupo GENTE, y cofinanciado por COLCIENCIAS, y que se ha utilizado con éxito en el acompañamiento de cursos de pregrado, especialización y educación a distancia.

De los trabajos mencionados anteriormente se puede destacar que el uso de las TIC en el diseño y la creación de herramientas de apoyo a los procesos de enseñanza y aprendizaje (Ambientes Virtuales de Aprendizaje - AVA), ha llevado al surgimiento de una amplia gama de soluciones que buscan soportar experiencias educativas en diferentes áreas del conocimiento, tales como la ingeniería, las ciencias básicas o la salud. Tomando a Internet como medio de comunicación, se han desarrollado Ambientes Virtuales de Aprendizaje que permiten a los estudiantes y docentes eliminar las barreras de tiempo y espacio en sus actividades académicas, lo que ofrece la posibilidad de mejorar la asimilación de conceptos para la construcción de nuevo conocimiento, el trabajo colaborativo y el aprendizaje de acuerdo con el ritmo de cada individuo o de un grupo.

Independientemente a que cada AVA se desarrolla teniendo en mente una experiencia de aprendizaje específica, es posible identificar algunos elementos presentes en la mayoría de estos ambientes, tales como herramientas de comunicación y colaboración, manejo de contenidos, programación de actividades, evaluación, seguimiento y acceso a recursos adicionales, entre otros. Estos

elementos conforman la base sobre la cual se establece una estrategia para la transmisión y generación de conocimiento, que es el fin último de todo proceso educativo.

El desarrollo de una plataforma robusta para la creación de estos AVA, exige tener en cuenta la creación, el manejo y despliegue de materiales educativos¹ que sirven de apoyo a la experiencia de aprendizaje y que en la mayoría de los casos, se basan en texto, imágenes, sonido o video. Al respecto se han adoptado varias estrategias para el manejo de dichos materiales, que van desde la inclusión de contenidos estáticos que no pueden ser editados posteriormente, hasta posibilidad de generarlos, almacenarlos y editarlos de manera dinámica, pasando por la carga y descarga de archivos de diferente formato, que contienen el material educativo. Cada estrategia tiene sus ventajas y desventajas, y es necesario realizar un análisis de las necesidades de aprendizaje para saber cuál es la estrategia adecuada para los objetivos que se buscan, si se requiere una combinación de algunas de ellas o, en última instancia, se debe contemplar la creación de una nueva estrategia para el manejo de materiales educativos dentro del ambiente.

Un aspecto estrechamente relacionado a las funciones de los AVA es la reutilización de materiales educativos en posteriores experiencias de aprendizaje. En esta dirección, para lograr cumplir con estándares de reusabilidad, accesibilidad e interactividad entre contenidos y con el ambiente en el que están integrados, se han creado especificaciones que reglamentan cómo se deben organizar los contenidos y construir los ambientes virtuales que los soportan. A pesar de los esfuerzos realizados para lograr que los contenidos sean reutilizables entre diferentes ambientes virtuales de aprendizaje, surge el problema de la distribución de contenidos, ya que no se cuenta con una plataforma que permita la creación de un

¹ Conjunto de recursos que realizan un aporte en la adquisición y construcción de conocimiento dentro de un proceso educativo. Con frecuencia reciben el nombre de *contenidos temáticos*, o simplemente *contenidos*.

ambiente distribuido donde sea posible el intercambio de contenidos entre estos ambientes.

Con el objetivo de ofrecer una solución al problema mencionado, el trabajo de investigación que se desarrolló propone el uso de una plataforma de intercambio de materiales educativos, basada en los estándares y las especificaciones reconocidas en esta área de estudio, y una infraestructura fundamentada en el uso de servicios Web XML, para permitir el acceso a los materiales, de forma transparente a los mecanismos utilizados para su almacenamiento y administración. En el capítulo 1 del presente documento se establece el planteamiento del problema, que define el área de estudio en la cual se desarrolló esta propuesta. Por su parte, en el capítulo 2 se presenta el objetivo general y los objetivos específicos del trabajo de investigación, que delimitaron el trabajo a realizar.

En el capítulo 3 se define el marco teórico sobre el cual se sustentó el desarrollo de la propuesta de investigación, representado en el estudio de los diferentes conceptos de ambiente virtual de aprendizaje, los materiales educativos reutilizables como medio para fortalecer la creación de experiencias de aprendizaje en línea, los repositorios de contenidos que permiten gestionar los recursos digitales, y el desarrollo de aplicaciones basadas en servicios Web XML, que permite ofrecer un alto nivel de interoperabilidad entre diversas plataformas.

El capítulo 4 presenta una estrategia para la creación de materiales educativos basados en objetos de aprendizaje, fundamentada en los trabajos de diferentes organizaciones internacionales, y soportado por algunos estándares para el desarrollo de contenidos. Posteriormente, el capítulo 5 muestra los aspectos conceptuales y técnicos relacionados con el diseño y la implementación de la plataforma de gestión e intercambio de objetos de aprendizaje. Finalmente se exponen algunas conclusiones y recomendaciones, que permiten identificar algunas áreas de estudio en las cuales se pueden desarrollar trabajos futuros.

1. PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN

Debido a que se ha puesto en evidencia la importancia del uso de herramientas que apoyen los procesos de enseñanza y aprendizaje, las instituciones de educación superior han aceptado el reto de crear herramientas para tal fin, que adoptan diferentes metodologías o estilos de aprendizaje. La utilización de las TIC para la creación de ambientes virtuales de aprendizaje basados en Internet, así como la posibilidad de desarrollar contenidos que puedan ser utilizados en varios cursos o ambientes plantea una serie de interrogantes desde diversas áreas del conocimiento, que deben ser estudiados para permitir el desarrollo de competencias en el diseño y la construcción de este tipo de herramientas, que brinden la posibilidad de mejorar la adquisición y construcción de conocimiento en los procesos educativos desarrollados en el interior de las instituciones.

Un problema que se observa con frecuencia en los ambientes virtuales existentes en el ámbito local es la carencia de medios para reutilizar los contenidos creados en un curso, ya sea dentro del mismo ambiente, o en otro ambiente virtual, situación generada por la falta de aplicación de estándares para la organización de contenidos. De igual manera, no se posee una infraestructura adecuada que permita la comunicación entre ambientes virtuales de aprendizaje, y que facilite la interacción entre éstos para el intercambio de contenidos. Esta situación conlleva a que **cada ambiente de aprendizaje se convierte en un ente aislado de información, sin posibilidades de comunicarse con otros ambientes y lograr una integración entre ellos.**

Por otra parte, es frecuente que los docentes o las personas encargadas de desarrollar y mantener contenidos deban realizar la labor de creación de los mismos contenidos en repetidas ocasiones para cursos diferentes, utilizando tiempo y recursos que pueden ser invertidos en otras labores, tales como la preparación de nuevos materiales, la programación de actividades de refuerzo, el trabajo colaborativo, las evaluaciones, la comunicación con los estudiantes o su seguimiento personalizado o grupal.

Una causa de la carencia de medios de distribución de contenidos utilizados en AVA radica en **que no se utilizan estándares para el diseño, la creación, catalogación y la distribución de materiales educativos**, o de los ambientes que los soporten, sea por su inexistencia o desconocimiento, lo que impide la posibilidad de utilización de recursos que podrían ser valiosos en otras experiencias educativas. Adicionalmente, en algunos casos las personas interesadas en el desarrollo y el uso de un ambiente virtual desean que la información que colocan a disposición en la herramienta no esté disponible para que otros ambientes la utilicen para su beneficio, poniendo en evidencia una posición que va en contra de la universalidad del saber y la necesidad de compartir el conocimiento para permitir su desarrollo.

Los hechos presentados anteriormente dan espacio para plantear interrogantes en torno a la necesidad de aunar esfuerzos en la identificación y utilización de estándares para la creación y distribución de materiales educativos y de las herramientas de desarrollo de contenidos que los soportan, para prevenir una inversión innecesaria en el desarrollo de materiales que pueden ser aportados por otras fuentes; de igual manera, se deben considerar estándares o guías de trabajo que permitan la creación de ambientes virtuales que soporten reutilización de contenidos y que proporcionen servicios para su almacenamiento, catalogación, búsqueda y distribución. Esto requiere la definición de protocolos de comunicación entre ambientes virtuales, políticas de identificación, nombramiento², seguridad, control de versiones y actualización automática contenidos reutilizables.

En la actualidad se están llevando a cabo esfuerzos que buscan la estandarización del proceso de creación de contenidos, de manera que cumplan con ciertos requerimientos de accesibilidad y reutilización, así como de interacción entre ellos y los ambientes de aprendizaje que los contienen. Sin embargo, no se ha establecido de forma definitiva una plataforma que facilite la interacción al nivel de ambientes

² Proceso por el cual se asigna un nombre único a cada elemento de una colección, lo cual permite su posterior identificación sin ambigüedades.

virtuales de aprendizaje, y que permita establecer una base para el desarrollo de redes de conocimiento entre instituciones educativas. En este sentido, la investigación que llevó a cabo tuvo como fin ofrecer respuestas a algunos de los interrogantes planteados anteriormente, al obtener una respuesta a la pregunta **¿Cómo crear una plataforma basada en estándares que permita la administración y la distribución de materiales educativos entre ambientes virtuales de aprendizaje?**

Los resultados de este proyecto beneficiarán al grupo GENTE y a la comunidad de la Universidad Industrial de Santander, al aportar bases conceptuales no disponibles en la actualidad, sobre las cuales se podrá construir ambientes virtuales que soporten contenidos reutilizables. En este sentido, la Universidad Industrial de Santander estará en capacidad de liderar en el ámbito local y regional el proceso de creación y distribución de este tipo de materiales, basados en estándares de reusabilidad.

Por otra parte, la realización de este proyecto aportará una herramienta integrada con la plataforma Aula Virtual, que facilitará la interacción entre los ambientes virtuales existentes y los nuevos ambientes que se desarrollen para satisfacer necesidades específicas, y que permitirá que las herramientas que fueron desarrolladas sin tener en cuenta los estándares de reutilización no queden obsoletas; esta posibilidad de integración generará un movimiento de estandarización de las herramientas existentes, para ajustarlas al modelo de contenidos reutilizables.

Una posibilidad a largo plazo que ofrece esta investigación es la conformación de un ambiente de intercambio y/o distribución de contenidos entre instituciones educativas, el sector público y privado, en el cual cada fabricante de materiales educativos pueda obtener algún tipo de reconocimiento o retribución por la labor realizada en la construcción de los mismos. Con la creación de este ambiente, se tendrá una base relativamente económica para la construcción de conocimiento utilizando ambientes virtuales de aprendizaje.

2. OBJETIVOS

2.1. OBJETIVO GENERAL

Proponer una plataforma que permita la integración entre ambientes virtuales de aprendizaje, a través del uso y la distribución de materiales educativos reutilizables

2.2. OBJETIVOS ESPECÍFICOS

- Identificar la aplicabilidad de los estándares existentes para la creación de contenidos reutilizables en el uso de estos en ambientes virtuales de aprendizaje para diferentes áreas de conocimiento.
- Diseñar una estrategia que permita la creación de materiales educativos reutilizables para su uso en ambientes virtuales de aprendizaje.
- Diseñar un protocolo de comunicación que permita proveer servicios de almacenamiento, administración, búsqueda y distribución de contenidos reutilizables a través de Internet.
- Implementar las herramientas software basadas en el diseño propuesto que permitan el almacenamiento, la administración, búsqueda y distribución de contenidos reutilizables, mediante el desarrollo de experiencias de utilización del protocolo diseñado.

3. MARCO TEÓRICO

Los avances en las tecnologías de la información y la comunicación han abierto una gama de posibilidades para la creación y el desarrollo de proyectos educativos bajo el modelo de la virtualidad. Debido a las ventajas que ofrece este tipo de educación, se ha generado una dinámica que permitirá la renovación de los estilos pedagógicos, el enriquecimiento de las posibilidades de aprendizaje y el acceso a fuentes globales de información. Esta motivación viene acompañada de un proceso de reflexión en torno al uso de TIC y de ambientes virtuales de aprendizaje para el desarrollo de nuevos programas educativos y mejoramiento de los existentes, y que conducirá a la identificación de problemas educativos frente a los retos planteados por los desarrollos regionales y globales, y la construcción de hipótesis explicativas y posibles alternativas de solución.

El marco teórico se organiza en torno las temáticas que conformaron el objeto de estudio del presente trabajo de. Entre ellas se puede destacar los ambientes virtuales de aprendizaje, los materiales educativos reutilizables, los repositorios digitales de contenidos y el desarrollo de aplicaciones basadas en servicios web XML.

3.1. AMBIENTES VIRTUALES DE APRENDIZAJE

La definición de ambiente virtual de aprendizaje parte del concepto de ambiente educativo, que es el conjunto de condiciones o circunstancias de una institución, orientados a favorecer el logro de los fines planteados por ésta. En este sentido, el ambiente tiene que ver con la filosofía que inspira las acciones, la razón de ser de los procesos, principios y lineamientos que orientan las decisiones sobre el para qué enseñar (fines educativos), el qué enseñar (contenidos), el cómo enseñar (métodos, estrategias), el qué y cómo evaluar; así mismo tiene que ver con el conjunto de

medios, experiencias y materiales educativos que son puestos al servicio de los docentes y estudiantes para lograr los fines educativos³.

De otro lado, el aprendizaje se entiende como un proceso permanente cuyo protagonista central es el estudiante, orientado a la construcción de quien aprende y a la apropiación por parte de éste del mundo y la cultura. La construcción de sí mismo la realiza cada estudiante si puede participar en experiencias que le permitan construir conceptos, competencias, actitudes y valores, con miras a mejorar su calidad de la vida. Esta construcción requiere que el estudiante desarrolle capacidades comunicativas para relacionarse fluidamente, desenvolverse en diversas situaciones, desarrollar habilidades sociales para el trabajo en equipo, resolver problemas de distinto tipo, e identificar y asumir las consecuencias de sus acciones y las de los demás.

Un ambiente de aprendizaje hace relación al ambiente educativo en el que docentes y estudiantes planifican, acuerdan, desarrollan y participan activamente en experiencias orientadas al logro del aprendizaje de los estudiantes y de los fines educativos de la institución. El término “virtual” o “en línea” se refiere a que estas experiencias se pueden llevar a cabo sin la necesidad de que personas, tiempo y espacio confluyan. Esto es posible gracias al avance de las Tecnologías de la Información y la Comunicación, ya que se puede establecer una relación interpersonal de carácter educativo sin tener un encuentro cara a cara entre el docente y el estudiante; puede haber un encuentro en el mismo lugar (un tablón de anuncios, por ejemplo), pero en tiempos distintos; por otra parte, se puede establecer un encuentro al mismo tiempo, pero en lugares diferentes (Chat, tele conferencia) o en lugares y tiempos diferentes (correo electrónico, videos).

Un ambiente virtual de aprendizaje es un ambiente que plantea el uso de las tecnologías de la información y la comunicación para *almacenar digitalmente*

³ Tomado de Aula Virtual: una alternativa en educación superior. Martha Vitalia Corredor y otros. Bucaramanga: Ediciones UIS. 2003.

algunas de las experiencias, los servicios y las actividades planificadas y acordadas, a las cuales los estudiantes podrán acceder en línea, mediante el computador, el software y la arquitectura de comunicación adecuados. Al decir “*almacenar digitalmente algunas*”, se habla de ambientes donde el grado de virtualización del proceso de aprendizaje no es total, es decir, en este caso los estudiantes participan en actividades presenciales y pueden acceder a través de un entorno virtual a un conjunto de actividades, información, servicios, enlaces, para apoyar el trabajo que realiza sin la presencia física del docente, pero sí bajo su acompañamiento. En estos casos, la responsabilidad de decidir qué ofrecer de manera virtual para apoyar a los estudiantes corresponde a los docentes, y dependerá del análisis de diversos factores educativos, tecnológicos e institucionales, pero en todos los casos es fundamental plantear el uso de las TIC de una manera creativa, inteligente y productiva para los fines de la educación. Finalmente, en un ambiente virtual de aprendizaje, lo realmente importante es la comunidad académica, que se une y se comunica utilizando las TIC para aprender de manera permanente y colaborativa, e implementa así nuevas estrategias de trabajo académico.

3.2. SISTEMAS DE GESTIÓN DE APRENDIZAJE – LMS

De acuerdo con Paulsen (2002), una gran parte del éxito de la educación en línea se debe a la disponibilidad de los Ambientes Virtuales de Aprendizaje, también conocidos Sistemas de Gestión de Aprendizaje (Learning Management Systems, LMS), o plataformas de tele-educación⁴. Un sistema de gestión de aprendizaje permite que una entidad ofrezca una serie de servicios que apoyan las experiencias de enseñanza y aprendizaje, tales como herramientas de comunicación sincrónica o asincrónica, medios de acceso a materiales educativos, administración de cursos, la creación, el manejo y la ejecución de evaluaciones, y el seguimiento y acompañamiento a los participantes de las experiencias. Técnicamente un LMS es

⁴ Aunque existe una leve diferencia entre la concepción de ambiente virtual de aprendizaje y la de LMS o plataforma de tele-formación, en lo restante del documento se dará preferencia al término LMS, debido a que se encuentra con mayor frecuencia en la literatura actual.

un software implementado como una aplicación Web, que se encarga de facilitar las labores de planeación, organización, desarrollo y la evaluación de un curso mediado por Internet. Zapata (2003) identifica algunas de las características más importantes de los LMS, que se describen a continuación.

Acceso remoto

Los LMS ofrecen la posibilidad de acceso remoto a las experiencias educativas, desde cualquier ubicación geográfica y en cualquier momento, siempre y cuando se cuente con los medios telemáticos y el software adecuados. El acceso a los LMS realiza generalmente a través del protocolo HTTP, aunque es posible encontrar implementaciones de LMS que utilizan otros modelos de red, tales como las redes Peer-to-Peer (P2P).

Independencia de plataforma

La información y los servicios que se administran en un LMS pueden ser accedidos por medio de navegadores web, que son herramientas basadas en estándares y que brindan una interfaz simple de acceso a la información, independiente a la plataforma utilizada por quienes acceden al LMS⁵.

Acceso selectivo

Todos los LMS implementan algún nivel de autenticación y autorización, que permite identificar a los usuarios de acuerdo con el rol que desempeñan dentro de la experiencia de enseñanza y aprendizaje. Estos roles se pueden asignar de forma dinámica, lo que permite que algunos usuarios puedan actuar como docentes o estudiantes en cursos diferentes. Al establecer diferentes niveles de usuarios con distintos privilegios de acceso se tienen usuarios administradores, encargados del mantenimiento de la herramienta y del servidor en el cual se encuentra instalada, los

⁵ Se debe tener en cuenta que aunque los navegadores son independientes de la plataforma a la cual tiene acceso, los materiales que son accedidos por medio de éstos puede tener un formato propietario.

usuarios, sus privilegios, y los cursos. También se tienen usuarios docentes, encargados de coordinar las experiencias educativas, y los estudiantes, como protagonistas de las experiencias de aprendizaje.

Interfaz común

La mayoría de los LMS implementan una interfaz común, que se adapta al rol de usuario y a las funcionalidades que éste posee dentro del sistema. Sobre esta interfaz de usuario se agrupan los elementos que apoyan la experiencia educativa, tales como materiales de texto, gráficos, animaciones.

Enlaces a fuentes externas de información

Los usuarios de un LMS pueden tener acceso a los recursos existentes en Internet, ya sea por medio de los enlaces adicionados o por las herramientas de búsqueda y navegación con que disponga el sistema.

Medios para la edición y actualización de la información

Los LMS proporcionan las herramientas necesarias para agregar o editar la información relevante para las experiencias de aprendizaje que se desean implementar. Generalmente la estructuración de la información se realiza de forma jerárquica, basada en una estructura de hiperenlaces, que forman conexiones de tipo conceptual o funcional

3.2.1. Los LMS desde el punto de vista de los servicios

Los servicios ofrecidos por los LMS dependen de la finalidad para la cual fueron creados. Sin embargo se pueden identificar algunos servicios básicos que conforman su núcleo, sobre el cual se genera el ambiente propicio para desarrollar

las experiencias de enseñanza y aprendizaje. De Benito⁶ propone una serie de funcionalidades básicas de los LMS, que se describen a continuación.

Gestión y administración de alumnos y profesores

Los LMS ofrecen funciones que permiten organizar y gestionar los usuarios del sistema, y clasificarlos de acuerdo con su rol dentro de las experiencias educativas. Esta labor involucra la creación de usuarios, la asignación de roles y privilegios de acceso, y la adición o edición de la información sobre los alumnos y profesores.

Creación, edición y acceso de contenidos de aprendizaje y recursos

Los contenidos de aprendizaje se refieren a cualquier recurso o bien que por sí mismo o combinado con otros proporciona un objetivo específico de información, apropiación o construcción de conocimiento dentro de una experiencia de aprendizaje. Físicamente estos recursos están conformados por elementos textuales, multimedia (esquemas, imágenes, sonidos, video o animaciones) o interactivos (simuladores, herramientas de cálculo o análisis). De igual forma se pueden ofrecer enlaces a recursos adicionales, como prácticas, tutoriales, ejercicios dirigidos y en un sentido más amplio a repositorios de información como bibliotecas digitales, sistemas de información, publicaciones electrónicas, buscadores temáticos, diccionarios electrónicos o enciclopedias digitales. Los LMS proporcionan los medios necesarios para la creación, la manipulación, la organización y el acceso a estos recursos, tales como editores, herramientas de edición, organización y catalogación, visualizadores y utilidades de búsqueda.

Comunicación e interacción

La comunicación es el elemento fundamental que define un LMS, ya que es a través del intercambio de información, el diálogo y la discusión que los participantes de la experiencia de aprendizaje se relacionan entre sí. Las herramientas de

⁶ Citado en ZAPATA (2003)

comunicación se pueden clasificar de acuerdo con la concurrencia en el tiempo en sincrónicas, que exigen que todos los participantes interactúen en determinado espacio de tiempo establecido anteriormente, o asincrónicas, que permiten desarrollar experiencias más extensas, sin la necesidad de la convergencia de todos los participantes a una hora determinada. Algunos ejemplos de herramientas de comunicación sincrónica son los salones de charla (*Chat*), y audio o video conferencia. Por su parte algunos ejemplos de comunicación asincrónica son el correo electrónico, las listas de discusión y los foros electrónicos.

Acompañamiento, control y evaluación

Estas herramientas se utilizan para facilitar el desarrollo de la experiencia, o para cuantificar en alguna medida el nivel de apropiación de conocimiento y/o adquisición de competencias de los estudiantes. Los LMS pueden ofrecer algunos mecanismos para realizar evaluaciones, tales como simulaciones o escenarios de aplicación de conceptos, o de *tests* que combinan preguntas de selección, de relación, de respuestas abiertas o de respuestas de tipo falso / verdadero.

La aplicación de la evaluación se puede dar en varias formas: auto evaluación, como el mecanismo para que el estudiante aprecie su progreso en la apropiación del conocimiento, la coevaluación, en la que el estudiante evalúa y es evaluado a su vez, y la heteroevaluación, en la cual el concepto de un agente externo puede complementar o contrastar los resultados obtenidos en la experiencia.

Trabajo Colaborativo (Collaborative Work – CW)

Algunos LMS incorporan elementos particulares que facilitan el trabajo colaborativo entre los participantes de la experiencia educativa. Estos elementos proporcionan mecanismos para compartir información, crear y modificar documentos de forma simultánea, programar tareas, construir mapas conceptuales, desarrollar lluvias de ideas o tener acceso a pizarras compartidas. De acuerdo con Zapata (2003), el objetivo de las herramientas de trabajo colaborativo es preparar a los estudiantes en la resolución de problemas, la toma de decisiones y el trabajo en grupo.

3.2.2. Los LMS desde el punto de vista de los usuarios

A pesar que las experiencias educativas en línea pueden ser desarrolladas por dos actores principales (docentes y estudiantes), es necesario realizar una caracterización más amplia que considere los roles de usuario que podrían participar en el diseño, la planeación, la construcción y el desarrollo de este tipo de experiencias. Zapata (2002) propone una serie de roles que se describen a continuación.

Estudiantes

Los estudiantes⁷ son el centro de mayoría de las experiencias llevadas a cabo en el contexto de un LMS⁸. Esta situación privilegiada en cuanto al aprendizaje les permite interactuar con los demás roles del LMS en diversas formas: Con los tutores en las consultas por diversos medios, y con los demás estudiantes o los profesores por medio de las herramientas de comunicación e interacción. Al ser el rol principal dentro de las experiencias, los estudiantes deben contar con la motivación adecuada para desarrollarlas a plenitud.

Profesores

Los profesores son los usuarios especialistas en el área de estudio en la cual se está desarrollando la experiencia de aprendizaje. Son los encargados de definir la estructura, el objetivo y el alcance de las experiencias educativas, así como las metodologías a utilizar y los medios de evaluación.

Tutores

Los tutores son los encargados de supervisar la ejecución de las tareas asignadas a los alumnos, de participar en los espacios de interacción como foros y debates,

⁷ Análogos al término estudiante son alumno, aprendiz, discente (Zapata, 2002).

⁸ Se deben considerar además los casos en los cuales el aprendizaje se da fuera del contexto de profesor-estudiante, tales como profesor – tutor, profesor – profesor, etc.

atender las consultas de los estudiantes a través de diversos medios, y proponer ejercicios y actividades de refuerzo. Además los tutores son los encargados de dinamizar las experiencias de aprendizaje, a través de las intervenciones en los espacios de interacción de los estudiantes y la moderación de las intervenciones de éstos.

Colaboradores

Estos usuarios son los encargados de coordinar las labores de logística necesarias para llevar a cabo la experiencia educativa, como la preparación de los medios de comunicación, la supervisión del funcionamiento de los instrumentos tecnológicos, la solución de imprevistos y la atención a las dificultades de los usuarios que participan en dicha experiencia.

Coordinadores

Los coordinadores son los encargados de velar por la coherencia de las experiencias educativas, y su concordancia con las políticas del programa y la institución. Para ello deben realizar el seguimiento de las experiencias desarrolladas, con el fin de obtener información que les permita realizar los ajustes correspondientes.

Administradores

Estos usuarios son los encargados de la configuración y el mantenimiento de la plataforma tecnológica sobre la que se desarrollan las experiencias, así como de la administración roles de usuario, cursos y privilegios.

Especialistas en elaboración de materiales educativos

Estos usuarios son los encargados de construir, estructurar y publicar los materiales educativos que se utilizarán dentro de las experiencias de aprendizaje. Para realizar a cabo su labor deberán poseer suficientes conocimientos en el manejo de las TIC y

de las herramientas de construcción y estructuración de materiales, con el fin de elaborar materiales de alta calidad pedagógica.

3.2.3. Herramientas asociadas con los LMS

Además del concepto de LMS, existe una gran cantidad de herramientas que se utilizan para crear o apoyar las experiencias de enseñanza y aprendizaje en línea. A continuación se describen algunas herramientas que son relevantes para este estudio, a partir de las caracterizaciones de Paulsen (2002) y posteriormente Zapata (2003).

3.2.3.1. Plataformas de Tele-Formación

Las plataformas de tele-formación⁹, o LMS propiamente dichos, son la instanciación de los conceptos asociados con un ambiente de aprendizaje en una herramienta tecnológica que integra los servicios necesarios para ofrecer cursos en línea, desarrollada de acuerdo con determinados propósitos educativos o formativos. Estas herramientas integran los medios de comunicación, el manejo de los materiales de apoyo al aprendizaje, el trabajo colaborativo, las unidades de gestión educativa y las demás herramientas creadas de acuerdo con su propósito educativo, tales como trabajo por proyectos, talleres o aprendizaje basado en competencias. Dentro de los LMS más conocidos se encuentran WebCT, TopClass, LearningSpace, Moodle, Claroline y ATutor. Una lista completa de los LMS más reconocidos puede ser consultada en EduTools (2005).

3.2.3.2. Herramientas para la creación de contenidos

Las herramientas para la creación de contenidos (Course Creation Tools – CCT) o herramientas de autor, son los medios tecnológicos que utilizan los profesores y/o diseñadores de cursos para crear los materiales y recursos que se utilizarán dentro

⁹ Ver Zapata, 2002

de las experiencias en línea. Estas herramientas pueden ser clasificadas de acuerdo con el tipo de material educativo que producen, por ejemplo archivos de texto, presentaciones, gráficos, animaciones, simulaciones, audio o video. Dentro de las herramientas de autor más importantes para el diseño y la construcción de materiales se encuentran los editores de páginas web (DreamWeaver, FrontPage, CouseBuilder, PageMill), los programas de autor propiamente dichos (Authorware, Quest Net+, IconAuthor, ToolBook, Director), y los programas para crear módulos de ejercitación (QuizMaker, QuestionMark, Qform, QuizCode).

3.2.3.3. Sistemas de gestión de contenidos (Content Management Systems – CMS)

De acuerdo con Nichani (2001), los CMS son herramientas que tienen por objetivo primordial la creación y administración contenido para su uso en línea. Este contenido se ensambla a partir de piezas menores de contenido, llamadas *componentes*, que en el contexto de la educación en línea se denominan *Objetos de Aprendizaje* (Learning Objects – LTSC, 2000), y en un sentido más amplio *Objetos de Aprendizaje Reutilizables* (Reusable Learning Objects).

3.2.3.4. Sistemas de Gestión de Contenidos de Aprendizaje (Learning Content Management Systems – LCMS)

El concepto de LCMS puede ser visto desde dos facetas: De acuerdo con Naish (2001), un LCMS extiende la funcionalidad de un LMS para permitir el uso de objetos de aprendizaje reutilizables dentro de las experiencias educativas. Por su parte, un documento de la IDC (2001) toma el LCMS como un componente adicional al LMS, que se encarga únicamente de la creación y estructuración de contenidos, su almacenamiento y entrega automática a los LMS que los soliciten. La diferencia entre las dos concepciones de LCMS no es significativa, ya que en la práctica es posible implementar un LMS que integre las funcionalidades del LCMS, o viceversa. Las características fundamentales de un LCMS de acuerdo con Uriarte (2002) son la existencia de las herramientas para la edición de los contenidos, en forma de

aplicaciones embebidas WYSIWYG (What-You-See-Is-What-You-Get), los sistemas flexibles para el diseño y la distribución de cursos, las herramientas para la reutilización de los objetos de aprendizaje, para la evaluación de los materiales, para la comunicación y el aprendizaje colaborativo, y los mecanismos de protección y seguridad.

Igualmente, Rengarajan (2001) plantea una definición de LMS y LCMS que establece mejor sus similitudes, diferencias y puntos de encuentro. Para Rengarajan, un LMS se centra principalmente en las competencias, las actividades de aprendizaje y la logística necesaria para llevar a cabo estas actividades, es decir que un LMS no se centra en la creación, reusabilidad, manejo o mejora del material educativo en sí.

Por el contrario, un LCMS brinda los medios para crear, localizar, reutilizar, manejar y mejorar el material educativo. Este material puede ser almacenado en forma de piezas pequeñas, identificables y reutilizables, cada una de las cuales busca cumplir con un objetivo de aprendizaje. El LCMS se encarga entonces de entregar estos objetos en forma individual, o como parte de una unidad de aprendizaje mayor (módulo, curso, etc.).

De las definiciones anteriores se observa que los LMS y los LCMS no son mutuamente excluyentes, por el contrario, los dos son elementos complementarios de una meta a más alto nivel: facilitar la construcción de conocimiento, así como la organización y transferencia de información. Los puntos de encuentro entre los LMS y los LCMS que establece Rengarajan (2002) se describen a continuación.

Contenidos

Los contenidos son el insumo principal para el LMS y el LCMS. El LMS los administra y los entrega a los usuarios, y puede realizar un seguimiento de su utilización en el nivel de las unidades de aprendizaje mayores (módulos, cursos, etc.). Por su parte, el LCMS realiza un seguimiento de los objetos individuales que conforman el contenido, y puede proporcionar además mecanismos para medir la

claridad de los objetos individuales, su relevancia y la eficacia que tienen en cumplir el objetivo de aprendizaje para el cual fueron creados.

Usuarios

Independientemente de si se consideran los recursos como objetos granulares o como objetos de aprendizaje de complejidad superior, los LMS y LCMS deben almacenar información de los usuarios que los utilizan. Generalmente los LMS mantienen información de cada usuario, que incluye sus nombres, su edad, direcciones de contacto, roles dentro del sistema, y de acuerdo con la funcionalidad implementada también podría almacenar información de las habilidades, preferencias de aprendizaje y de utilización de las herramientas disponibles. Los LCMS solamente mantienen la información necesaria para conocer las preferencias de uso de los contenidos por parte de los usuarios, para poder ofrecer los materiales que se ajusten a sus preferencias. Además los LCMS pueden ofrecer algunos mecanismos para la edición de materiales en línea, la modificación colaborativa o la evaluación de los materiales.

Administración

Los LMS y los LCMS poseen diferentes intereses administrativos con respecto a los usuarios y los contenidos. Por un lado, los LMS ofrecen herramientas para la administración de usuarios que pueden incluir el manejo de perfiles de usuario de acuerdo con las propiedades organizacionales, con su rol en el proceso de enseñanza y aprendizaje o con su desempeño en las experiencias de aprendizaje. Por su parte el LCMS se concentra en administrar los aspectos que tienen que ver con el acceso de los usuarios a los contenidos, el seguimiento a sus actividades de creación, edición y visualización.

3.2.3.5. Herramientas para la comunicación y el apoyo al trabajo colaborativo

Estas herramientas permiten que un grupo de personas se comuniquen y coordinen sus actividades para cumplir con un objetivo común. Generalmente incluyen características como conferencias electrónicas (Chat, correo, audio-conferencia y/o video-conferencia), edición colaborativa de documentos y pizarra compartida. Dentro de ellas herramientas se pueden resaltar BSCW, ClassAct, ClassPoint, FirstClass, iClass, NetMeeting, LearnLine y CaMILE.

3.2.3.6. Proveedores de servicios de aprendizaje (Learning Service Providers – LSP)

Los proveedores de servicios de aprendizaje son instituciones que se han especializado en ofrecer un servicio comercial de hospedaje de LMS, para instituciones que no cuentan con los recursos de infraestructura para soportar sus programas de educación en línea. Generalmente los proveedores se encargan solamente de la administración y el mantenimiento de la herramienta, mientras que las instituciones que toman el servicio se encargan del manejo de cursos y usuarios, y la gestión de las experiencias educativas que se realizan dentro de estos ambientes.

3.2.3.7. Sistemas de gestión estudiantil (Student Management System - SMS)

Estos sistemas son la base de cualquier sistema de información de una institución educativa, al administrar la información de los estudiantes, las facultades, los planes de estudio, las admisiones, matrículas, notas, etc. Esta clase de sistemas aportan la información básica que requiere el LMS, como la distribución de los cursos y grupos, y la información de los estudiantes y profesores.

3.2.3.8. Sistemas contables (Accounting Systems – AS)

En un modelo educativo de virtualidad total, se debe contar con una herramienta que permita manejar la información de las transacciones realizadas entre la institución, los estudiantes, los profesores, los proveedores, etc. Los datos de esta herramienta podrían ser usados para garantizar o denegar el acceso a determinados cursos, para recibir pagos en línea o para realizar pagos en línea a los tutores.

3.3. OBJETOS DE APRENDIZAJE

En la actualidad existen varias posturas alrededor de la conceptualización de los objetos de aprendizaje¹⁰. Unas se basan en considerar a un objeto de aprendizaje como una agrupación de contenidos con un objetivo de aprendizaje específico, mientras que otras argumentan que se debe profundizar en el concepto de objeto, como herramienta que se puede utilizar para la construcción de conocimiento, en vez de ser utilizado para una experiencia de aprendizaje específica. Por otro lado existen posturas que consideran que cualquier medio electrónico se puede tomar como un objeto simple, a partir del cual se pueden construir objetos de aprendizaje de mayor complejidad.

A partir de las posiciones anteriores se puede observar que presentan un aspecto en común: considerar a los objetos de aprendizaje como entidades granulares, con posibilidades para ser integradas en unidades mayores. Sin embargo el tamaño de estas entidades no se encuentra definido, lo que da a lugar a que cualquier medio (una letra o palabra, una frase, un párrafo, un texto, capítulo, etc.) pueda ser considerado un objeto de aprendizaje.

Existe una definición general de objetos de aprendizaje que es aceptada por la industria. El IEEE Learning Technology Standards Committee – LTSC (2002) define

¹⁰ Adaptado a partir de Alderman (2004)

objeto de aprendizaje como “cualquier entidad, digital o no digital, que puede ser usada, reutilizada o referenciada durante el aprendizaje soportado por la tecnología”. La expresión “cualquier entidad” abre las posibilidades de objetos de tamaño diverso, con diferentes funciones y objetivos. Igualmente la expresión “aprendizaje soportado por la tecnología” de acuerdo con la especificación de la IEEE incluye “sistemas de entrenamiento basados en el computador, ambientes interactivos de aprendizaje, sistemas inteligentes de instrucción asistida por computador, sistemas de educación en línea y ambientes de aprendizaje colaborativo”.

Por otra parte Cisco Systems (Cisco, 2000) define una jerarquía de dos niveles de objetos: Objetos de Aprendizaje Reutilizables (Reusable Learning Objects - RLO), y Objetos de Información Reutilizables (Reusable Information Objects – RIO). Los RIO son piezas granulares de información, que son independientes del medio y del contexto, y que pueden ser reutilizadas en la construcción de diverso RLO.

Según la conceptualización de Cisco, los RIO se construyen con un objetivo específico en mente, y se pueden clasificar en conceptos, hechos, principios, procesos y procedimientos. A su vez la estructura interna de un RIO puede constar de una sección de ítems de contenido, una sección de ítems de práctica y una sección de ítems de evaluación. La agrupación de ítems de contenido y práctica conforman actividades de aprendizaje, mientras que los ítems de evaluación se utilizan para medir el grado de completitud del objetivo propuesto por el RIO.

De las definiciones anteriores se puede observar que no existe una restricción del tamaño de un objeto de aprendizaje. Esta omisión intencional permite establecer el tamaño de los objetos de acuerdo con su *contexto* de utilización, para obtener agrupaciones de objetos que se pueden etiquetar como “cursos”, “módulos”, “lecciones”, “laboratorios”, “casos de estudio”, etc. A continuación se presentará una definición de objeto de aprendizaje, de acuerdo con Alderman (2004).

3.3.1. Definición de objeto de aprendizaje

Un objeto de aprendizaje es una agrupación de materiales y/o elementos multimedia, con un objetivo de aprendizaje definido, y con datos que describen su estructura y finalidad. Para que facilitar su reutilización, el objeto de aprendizaje deberá cumplir con las siguientes características:

- Basado en un objetivo: Cada objeto de aprendizaje deberá tener un objetivo de aprendizaje. Este objetivo se puede lograr al combinar diversos materiales, incluyendo textos, multimedia e interactividad.
- Libre de contexto: El objeto de aprendizaje deberá ser fácilmente identificado dentro de la jerarquía de contenidos asociada, para hacerlo portable, y reutilizable.
- Interactivo: Esta característica no es imprescindible, sin embargo es necesario que cada objeto presente algún grado de interactividad que motive el desarrollo de la experiencia de aprendizaje.
- Auto-descriptivo: Además del contenido destinado al aprendizaje, el objeto deberá contener información que permita conocer su estructura y finalidad. Esta información permitirá la posterior búsqueda y reutilización del objeto.
- Auto-contenido: Cada objeto de aprendizaje deberá estar conformado como una unidad atómica, a partir de la cual se puedan crear unidades mayores. Se debe tener en cuenta que el término atómico no hace referencia al tamaño del objeto de aprendizaje, sino a su conformación y su objetivo de aprendizaje, en el sentido que el objeto deberá contener todos aquellos recursos de los cuales depende la experiencia (imágenes, elementos multimedia, simuladores, etc.)¹¹

¹¹ Sin embargo, gracias a las posibilidades ofrecidas por internet, los materiales dentro de un objeto de contenido pueden contener hiperenlaces a otros objetos o recursos externos a éste.

- Libre de formato: Para permitir su reutilización en diversos contextos, el objeto deberá poseer los elementos mínimos para presentar la información al usuario, y la interfaz de usuario estará determinada por la herramienta que utilice el objeto.

3.3.2. Motivación por el estudio y uso de los objetos de aprendizaje

La motivación por el estudio de los objetos de aprendizaje surge a partir de la necesidad de compartir y reutilizar los materiales educativos en diversos *contextos de aprendizaje*. Reigeluth y Nelson¹² plantean que los profesores en su proceso de preparación de materiales deben realizar una fase inicial de dividir el cada material en sus partes constituyentes, para luego reensamblarlos de acuerdo con las necesidades de la experiencia educativa que planean desarrollar. Desde este punto de vista el uso de los objetos de aprendizaje podría ofrecer un ahorro significativo de tiempo en el proceso de construcción de materiales, debido a que los profesores tendrían la posibilidad de acceder directamente a aquellas partes de los contenidos que se ajustan a sus necesidades y se ahorrarían el tiempo requerido para desagregar y reestructurar los materiales existentes.

Esta premisa permite vislumbrar un panorama en el cual se tendrá una gran cantidad de objetos de aprendizaje, creados en formatos altamente reutilizables, lo cual permitirá que los usuarios de los sistemas de gestión de aprendizaje busquen y localicen los recursos más apropiados para el desarrollo de sus experiencias de enseñanza y aprendizaje.

3.3.3. Clasificación de los objetos de aprendizaje

La definición general de objeto de aprendizaje favorece que este concepto pueda ser aplicado a diferentes contextos educativos. Con el propósito de establecer los lineamientos que permitan determinar con mayor precisión el alcance de los objetos de aprendizaje, se han creado algunas clasificaciones que establecen los criterios

¹² Citados en Wiley, 2000.

básicos para diferenciarlos. Wiley (2001) establece una clasificación de cinco niveles de objetos de aprendizaje, que se describe a continuación:

- **Fundamental:** Un recurso individual, que no se encuentra combinado con ningún otro recurso. El objetivo principal de este objeto es presentar un concepto o un ejemplo.
- **Combinado – Cerrado:** Este tipo de objetos está formado por la combinación de recursos digitales, cuyos componentes no pueden ser reutilizados. Su objetivo es proporcionar una agrupación de conceptos o una práctica.
- **Combinado – Abierto:** Esta clase de objetos de aprendizaje comprende una agregación de materiales, en las cuales algunos componentes pueden ser reutilizados en otras experiencias. Un ejemplo de esta clase de objetos son las páginas web, en las cuales es posible reutilizar el texto, las imágenes o animaciones que las constituyen. Los objetos combinados – abiertos agrupan y combinan los objetos fundamentales y los combinados-cerrados para crear una unidad de aprendizaje completa.
- **Generativo – Presentación:** Estos objetos se componen a partir de la combinación de los objetos anteriores para crear presentaciones que se pueden utilizar en referencias, instrucción, práctica y pruebas. Este tipo de objetos puede ser reutilizado en contextos de aprendizaje similares.
- **Generativo – Instruccional:** Este tipo de objetos de aprendizaje contiene la lógica necesaria para combinar objetos de aprendizaje y evaluar el desempeño del estudiante, y puede ser usado en diferentes contextos.

3.4. ESTÁNDARES PARA LA EDUCACIÓN EN LÍNEA

La educación en línea utiliza de manera integrada y pertinente los computadores y las redes de comunicación, con el propósito de conformar un ambiente propicio para la construcción de diversas experiencias de aprendizaje. Esta actividad ofrece

ventajas sobre las estrategias tradicionales, tales como la independencia geográfica y temporal, la mediación de la tecnología como elemento de organización y distribución de los datos y la posibilidad de evaluación y seguimiento de dichas experiencias. Conscientes de estas fortalezas, distintas organizaciones y empresas relacionadas con el mundo del software, la capacitación y la educación, están trabajando en la creación de estándares y especificaciones que logren crear plataformas, materiales y recursos interoperables entre distintos LMS, además de la creación de nuevos productos que se dediquen a generar estos recursos interoperables, o que ayuden a transformar los existentes, para ajustarlos a los modelos propuestos.

En un comienzo los esfuerzos para lograr la estandarización fueron múltiples y conducidos por distintas industrias, lo que ocasionaba que no fueran compatibles entre sí. Sin embargo, en la actualidad los esfuerzos individuales se han agrupado, y las organizaciones están trabajando en conjunto intercambiando ideas y conocimientos. Cuatro de las más importantes iniciativas para lograr especificaciones y estándares para la educación en línea son el Institute for Electrical and Electronic Engineers (IEEE) Learning Technology Standards Committee (LTSC 2000), Advanced Distributed Learning (ADL 2002), IMS Global Learning Consortium y el Aviation Industry CBT Committee (AICC 1998). A continuación se describe cada iniciativa, y se presenta una reseña de su aporte a la generación de estándares para la educación en línea. Finalmente se realizará una descripción del modelo de referencia SCORM, tomado como base en esta investigación.

3.4.1. AICC, Aviation Industry CBT Committee

Creado en 1988, AICC es un grupo internacional de profesionales dedicados al entrenamiento y la capacitación basados en tecnología. A finales de la década de los 80, Internet no existía en el ámbito público como se conoce hoy en día, por lo tanto AICC se creó cuando sólo se conocía el CBT (Computer-Based Training) o Capacitación Basada en Computadores. El aporte de la AICC consistió en publicar

varias guías, incluyendo algunas relacionadas con el hardware y software¹³. Estas guías se enumeran a continuación (AICC, 1998).

- AGR 001: AICC Publications: Este documento proporciona un resumen de las guías y recomendaciones, los documentos técnicos y los artículos que esta organización ha publicado.
- AGR 002: Courseware Delivery Stations: Contiene las recomendaciones para la adquisición de los equipos que se utilizarán para desarrollar actividades de entrenamiento soportado por el computador. Estas recomendaciones incluyen especificaciones de velocidad del procesador y del bus, voltaje de entrada, memoria RAM, sistema operativo, monitores, teclados, unidades de CD, sistemas de reproducción de sonido y video, y dispositivos de red.
- AGR 003: Digital Audio: Proporciona algunas recomendaciones para la interoperabilidad del audio digital. Esta interoperabilidad se refiere a la capacidad de que un material de curso que contenga audio pueda ser reproducido en diversas plataformas de hardware y de sistema operativo, así como diferentes formatos de audio.
- AGR 004: Operating/Windowing System: Proporciona una recomendación formal de los sistemas de gestión de ventanas utilizados para el desarrollo de entrenamientos basados en el computador. Contiene además los resultados de una encuesta llevada a cabo por la AICC, de los sistemas operativos basados en ventanas
- AGR 005: CBT Peripheral Devices: Este documento recomienda algunos lineamientos que promueven la interoperabilidad de los periféricos como dispositivos de entrada XY (Touch-screen, mouse, trackball) y reproductores de videodiscos.

¹³ AGRs - AICC guidelines and recommendations (AICC, 1998).

- AGR 006: Computer-Managed Instruction: Proporciona los lineamientos para la interoperabilidad de los sistemas de instrucción gestionada por computador (CMI), para permitir que los materiales utilizados en el CBT (Courseware) puedan ser utilizados en diversas herramientas. Estos lineamientos incluyen la posibilidad de que el Courseware pueda intercambiar información con el sistema de instrucción que las utiliza.
- AGR 007: Courseware Interchange: Este documento recomienda los lineamientos para el intercambio de los elementos que conforman una lección de CBT. Estos elementos incluyen texto, gráficos, animaciones y sonido. Los lineamientos contienen la descripción de los elementos que conforman una unidad de instrucción y los formatos estándares para estos elementos.
- AGR 008: Digital Video: Ofrece los lineamientos para la creación, la distribución y el uso de video digital dentro de los contenidos.
- AGR 009: Icon Standards: User Interface: Contiene las guías de trabajo que definen las funciones de la interfaz del estudiante y su representación gráfica asociada.
- AGR 010: Web-Based Computer-Managed Instruction: Este documento basado en la guía AGR 006 contiene las recomendaciones que promueven la interoperabilidad de sistemas de instrucción soportados por computador y basados en Internet.

Adicionalmente la AICC cuenta con un programa de certificación y dispone de una *suite* de pruebas, que permite a las entidades verificar que sus productos estén ajustados a las diferentes guías de trabajo. En la actualidad la guía de trabajo que ha recibido mayor atención es la AGR 010, considerada un estándar de facto en la educación en línea, ya que plantea la interoperabilidad de las plataformas de formación y los cursos que se imparten dentro de ellas. En esta guía se plantean soluciones a dos de los problemas más importantes que deben enfrentar los LMS:

- La utilización en un LMS de cursos creados por terceros. Este objetivo se consigue al definir el curso como una entidad totalmente independiente de la plataforma, y crear un sistema (archivos) de descripción del curso que pueda ser soportado por cualquier plataforma.
- La comunicación entre el LMS y el curso, para que este último el curso pueda obtener información necesaria sobre el usuario, y después transmitir los resultados de las interacciones y evaluaciones realizadas a la plataforma, para su almacenamiento y tratamiento estadístico. Este segundo objetivo es logrado mediante la definición de un mecanismo de comunicación entre el curso y la plataforma, y un conjunto de datos mínimos que deben ser transmitidos del curso a la plataforma y viceversa. La AICC describe dos mecanismos, uno más sencillo y extendido basado en el protocolo HTTP, y otro mediante una API.

3.4.2. Institute for Electrical and Electronic Engineers Learning Technology Standards Committee (IEEE LTSC)

El IEEE es un conocido cuerpo multinacional que desarrolla estándares internacionales para sistemas eléctricos, electrónicos, computacionales y de comunicaciones. Este cuerpo se organiza en distintos comités que, realizan sesiones en las cuales analizan las distintas tecnologías, entregando como resultado una especificación o recomendación en forma de estándar.

Uno de estos comités es el Learning Technology Standards Committee (LTSC) o Comité para los Estándares de la Tecnología del Aprendizaje, el cual ha producido la especificación de los Metadatos de los Objetos de Aprendizaje o Learning Object Metadata (LOM)¹⁴. Esta especificación define elementos para describir los recursos

¹⁴ Interpretado literalmente, el término metadatos significa “datos sobre los datos”, es decir datos que proporcionan algún nivel de información descriptiva acerca de los datos en sí. Como ejemplo se puede tomar una biblioteca: Los libros serían los datos, y las tarjetas de cada libro sus metadatos.

de aprendizaje, y es la base de trabajos posteriores realizados por otras entidades como IMS y ADL, que la incorporan dentro de sus modelos de referencia.

A su vez, el LTSC tiene más de una docena de grupos de trabajo (working groups o WGs) y grupos de estudio (study groups o SGs) que desarrollan especificaciones para la educación en línea. La Tabla 1 muestra algunos grupos que conforman el LTSC, de acuerdo con el área en el cual se centra su trabajo.

Tabla 1. Grupos de trabajo que conforman el LTSC

Actividad	Grupos de trabajo
Actividades Generales de la IEEE LTSC	IEEE 1484.1 Architecture and Reference Model IEEE 1484.3 <i>Glossary</i>
Actividades relacionadas con datos y metadatos	IEEE 1484.12 Learning Object Metadata IEEE 1484.14 Semantics and Exchange Bindings IEEE 1484.15 Data Interchange Protocols
Actividades relacionadas con los LMS y las aplicaciones	IEEE 1484.11 Computer Managed Instruction IEEE 1484.18 Platforms and Media Profiles IEEE 1484.20 Competency Definitions

Adicionalmente, el LTSC trabaja en forma coordinada con otra iniciativa denominada ISO JTC1 SC36, que es un subcomité integrado en forma conjunta por la ISO (International Standard Organization) y por la IEC (International Electrotechnical Commission), dedicado a la normalización en el ámbito de las Tecnologías de la Información para la formación, educación y el aprendizaje.

3.4.3. IMS Global Learning Consortium, Inc.

Este consorcio está formado por miembros provenientes de organizaciones educativas, empresas públicas y privadas, y tiene como misión desarrollar y promover especificaciones abiertas para facilitar las actividades del aprendizaje en línea. El trabajo de la IEEE fue recogido por IMS, con el objetivo de crear un formato que implementara en la práctica las recomendaciones de la IEEE y la AICC.

Uno de los aportes más significativos de IMS consistió en definir un tipo de archivo XML para la descripción de los contenidos de los cursos¹⁵. De esta forma, cualquier LMS puede cargar un curso leyendo su archivo de configuración, el cual contiene los metadatos, la organización de su estructura y sus recursos locales (páginas, imágenes, etc.). La Tabla 2 presenta un resumen de las principales iniciativas lideradas por IMS.

Tabla 2. Iniciativas del IMS Global Learning Consortium

Iniciativa	Objetivos
Learning Object Metadata (LOM)	Identificación y etiquetado de Objetos de Aprendizaje
Empaquetamiento de Contenidos (Content Packaging)	Describir y empaquetar material reutilizable en objetos portables e interoperables
Interoperabilidad de Preguntas y Tests (Question and Test Interoperability, QTI)	Definir una estructura XML para codificar preguntas y Test On-Line, y permitir su intercambio.
Empaquetamiento de Información del Alumno (Learner Information Packaging, LIP)	Definir estructuras XML para el intercambio de información de los alumnos entre sistemas de gestión de aprendizaje
Secuencia Simple (Simple Sequencing)	Definir las reglas que describen el flujo de instrucciones a través del contenido según el resultado de las interacciones del estudiante
Diseño del Aprendizaje (Learning Design)	Este grupo de trabajo del IMS investiga sobre las maneras de describir y codificar las metodologías de aprendizaje incorporadas en una solución e-learning
Repositorios Digitales (Digital Repositories)	Crear especificaciones y recomendaciones para lograr la interoperatividad entre repositorios digitales.

¹⁵ Este archivo se conoce como manifiesto (manifest), y su nombre físico debe ser imsmanifest.xml.

Iniciativa	Objetivos
Definición de competencias (Competency Definitions)	Crear una manera estandarizada de describir, referenciar e intercambiar definiciones de competencias ¹⁶
Accesibilidad ¹⁷ (Accessibility)	Promover el contenido accesible a través de recomendaciones, guidelines, y modificaciones a otras especificaciones

3.4.4. ADL SCORM

Formada en 1997, la iniciativa ADL (Advanced Distributed Learning) es un programa del Departamento de Defensa de los Estados Unidos (DoD) y la Oficina de Ciencia y Tecnología de la Casa Blanca, que busca desarrollar los principios y las guías de trabajo necesarias para el desarrollo y la implementación eficiente, efectiva y a gran escala, de formación educativa sobre nuevas tecnologías, que hacen uso de Internet.

El principal aporte de esta iniciativa consistió en integrar los aspectos más relevantes de sus predecesoras (AICC, IEEE LOM e IMS), para crear un modelo de referencia, SCORM. (Shareable Content Object Reference Model - Modelo de Referencia para Objetos de Contenido Compartibles). Este modelo recoge los aspectos más sobresalientes de las demás especificaciones y establece una serie de requerimientos que deben cumplir los contenidos y los LMS que los gestionan. En la actualidad muchas organizaciones están trabajando de forma colaborativa sobre el modelo SCORM, para proporcionar los lineamientos sobre la creación de contenidos, y en general sobre las tecnologías y servicios ofrecidos en la educación en línea. A pesar que algunos de estos estudios se encuentran en una etapa temprana, SCORM ha demostrado tener la suficiente capacidad para brindar la interoperabilidad y reusabilidad, además de los fundamentos que ayuden a las

¹⁶ El término competencia es usado en un sentido muy general, que incluye habilidades, conocimiento, tareas, y resultados de aprendizaje.

¹⁷ La accesibilidad se refiere a aquella tecnología que puede ser usada sin tener acceso pleno a uno o más canales de entrada y salida, generalmente visuales o auditivos.

instituciones a tener un conocimiento preciso de cómo usar las tecnologías en la educación, para construir y operar en el ambiente de aprendizaje del futuro.

Las especificaciones del modelo de referencia SCORM están organizadas como libros separados en una estructura modular, que permite que cada una de ellas evolucione por separado, de acuerdo con los lineamientos establecidos por el modelo. A continuación se realiza una breve descripción de los libros que conforman el modelo SCORM¹⁸, y posteriormente se revisarán los componentes más importantes de este modelo.

Libro 1. Scorm Overview

Este libro contiene una descripción general de la iniciativa de ADL, un análisis de SCORM, y un resumen de las especificaciones técnicas contenidas en los demás libros.

Libro 2: Modelo de agregación de contenidos - Scorm Content Aggregation Model - CAM

Este libro describe los componentes que se utilizan dentro de una experiencia de aprendizaje, y la forma de empaquetarlos para permitir su intercambio entre diversos LMS; también especifica la forma de describir los componentes por medio de metadatos, con el fin de permitir su búsqueda y localización.

Libro 3: Ambiente de ejecución - SCORM Run-Time Environment - RTE

Este libro describe los requerimientos funcionales que deben cumplir los LMS para gestionar su comunicación con los objetos de contenido, así como un modelo de datos estandarizado para transferir información acerca del usuario entre los objetos de contenido y el LMS.

¹⁸ Para mayor información, consultar los libros SCORM Overview, SCORM Content Aggregation Model, SCORM Run-Time Environment y SCORM Sequencing and Navigation. Versión 1.3.1, 2004.

Libro 4: SCORM Sequencing and Navigation – SN

Este libro proporciona los conceptos sobre los que se establecen los mecanismos para determinar la secuencia de presentación de los objetos de contenido, dentro de las experiencias de aprendizaje. Además define los requerimientos que deben cumplir los SCO y los LMS para permitir diferentes formas de secuenciación.

3.4.4.1. Modelo de agregación de contenidos

El objetivo del modelo de agregación de contenidos de SCORM es proporcionar un medio común para estructurar materiales educativos compartibles y reutilizables, a partir de diversas fuentes. Este modelo define cómo un contenido puede ser identificado, descrito y agregado dentro de un paquete de contenidos, y cómo puede ser compartido por diversos LMS o por diversos repositorios de información. SCORM introduce el concepto de Objeto de Contenido Compartible (SCO), para definir una agrupación de materiales con fin educativo (texto, imágenes, animaciones), que tiene la capacidad de establecer comunicación con el sistema de gestión de aprendizaje que lo utiliza, con el fin de proporcionar mecanismos para el seguimiento del progreso del estudiante.

Adicionalmente, el modelo de agregación de contenidos define una especificación para describir los objetos de contenido por medio de metadatos, basada en el IEEE LSTC Learning Object Metadata (LOM), con el fin de facilitar su búsqueda y localización dentro de repositorios de contenidos. De acuerdo con el modelo SCORM, los metadatos se aplican en varios niveles: a los *assets* (elementos de contenidos de más bajo nivel, como imágenes, archivos de sonido), a los *objetos de contenido compartible* (SCO), a las *actividades* (unidad de aprendizaje básica), a las *organizaciones* (jerarquía de actividades) y a las agregaciones de contenido (El paquete de contenido en sí que podría ser visto como módulo, lección, curso, etc.).

El modelo de agregación de contenidos de SCORM se basa en el trabajo de IMS, y principalmente en la especificación para el empaquetamiento de contenidos (IMS Content Packaging Specification). Esta especificación describe las estructuras que

se utilizan para proporcionar la interoperabilidad del contenido basado en Internet, a través de diversas herramientas buscan importar, exportar, agregar y desagregar paquetes de contenido (SCORM CAM, 2004).

Estructura de un paquete de contenidos

Según la IMS Content Packaging Specification, la estructura de un paquete de contenidos está compuesta por los recursos físicos que conforman el paquete (textos, imágenes, sonidos, animaciones), y un documento XML que describe la estructura del paquete de contenido y los recursos que contiene. Este archivo se denomina manifiesto (Manifest), y debe ser llamado `imsmanifest.xml`. La Figura 1 muestra la estructura de un paquete de contenidos, de acuerdo con la especificación mencionada.

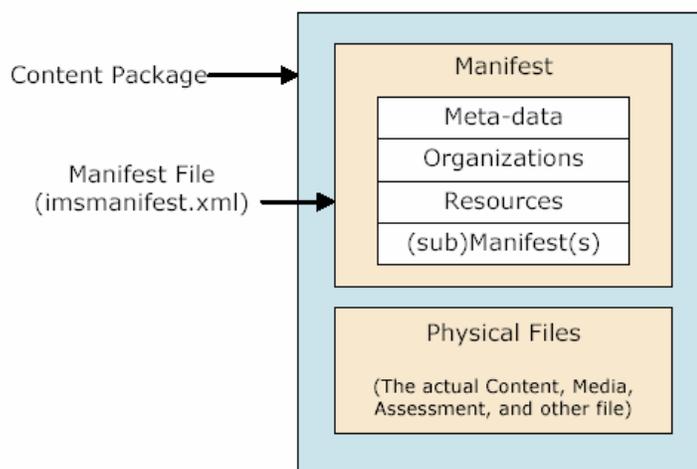


Figura 1. Estructura de un paquete de contenidos SCORM¹⁹

En esta figura se pueden apreciar los componentes de un paquete de contenidos (Content Package): El archivo de manifiesto (`imsmanifest.xml`) contiene los metadatos del paquete de contenidos, (Meta-data) la organización de las

¹⁹ Las imágenes de esta sección fueron tomadas de los libros SCORM CAM, RTE y SN. Versión 1.3.1. 2004. Derechos de autor de Advanced Distributed Learning. 2004

actividades de aprendizaje (organizations), una descripción de los recursos utilizados dentro del paquete (resources) y posiblemente otros archivos de manifiesto (sub-manifests). Adicionalmente el paquete deberá contener todos aquellos recursos que sean locales al paquete (physical files).

Organizaciones de contenidos

Una organización de contenidos puede ser vista como un mapa de la estructura de actividades (ítems) del paquete de contenidos, que debe ser seguido por el estudiante para completar sus experiencias de aprendizaje con el paquete de contenidos. Sus componentes más importantes se muestran en la Figura 2.

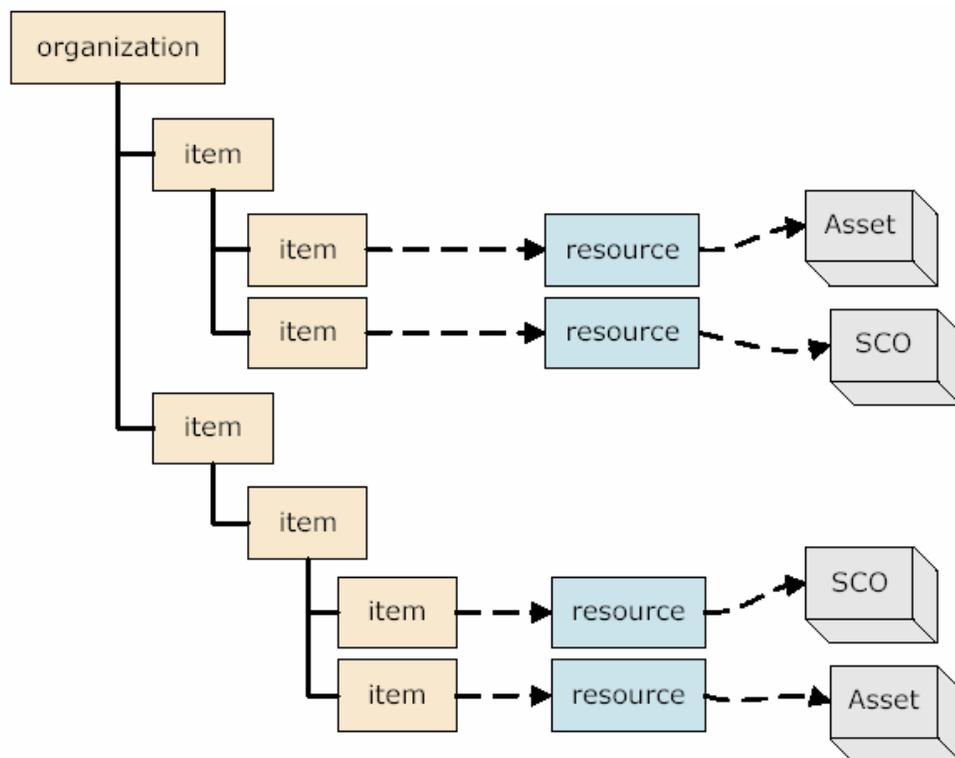


Figura 2. Estructura de una organización de contenidos

Los componentes de la estructura de una organización de contenidos son:

- Jerarquía de actividades: Consiste en una organización lógica de las actividades de aprendizaje descritas en el paquete de contenidos. La jerarquía de actividades

puede ser vista como una estructura de árbol o tabla de contenidos, y representa la estructura del material propuesta por el autor del paquete. Esta jerarquía está compuesta por actividades (ítems), que tienen asociados recursos (resources) o que pueden contener otras actividades, en cuyo caso se denominan clusters²⁰. A su vez un recurso hace referencia a un SCO o un Asset, es decir a un objeto de contenido compartible y reutilizable en diversas agregaciones de contenidos.

- **Metadatos:** Este componente opcional contiene datos descriptivos acerca de las actividades definidas en la organización de contenidos. Estos metadatos pueden ser usados para determinar cómo se debe utilizar el recurso asociado a la actividad en el contexto de la organización de contenidos.
- **Secuenciación y navegación:** Este componente opcional define algunas restricciones que controlan el flujo de presentación de las actividades de aprendizaje del paquete de contenidos. Si este componente no se define dentro del paquete de contenidos, el usuario podrá elegir de forma arbitraria cuál será la próxima actividad a revisar. Al agregar el componente de secuenciación y navegación es posible restringir las actividades que pueden ser desarrolladas, por ejemplo al establecer que las actividades deben ser revisadas en un orden estrictamente secuencial, de la primera a la última. Además el modelo permite definir cierto nivel de adaptabilidad de los contenidos, de acuerdo con el desempeño del estudiante.

Metadatos

El esquema de metadatos adoptado por SCORM para describir sus componentes estructurales (agregaciones de contenidos, organizaciones, actividades, SCOs y Assets) proviene de LOM, el estándar de metadatos para objetos de aprendizaje desarrollado por la IEEE (LOM, 2000). Adicionalmente SCORM hace uso del mapeo de los elementos LOM a XML, también propuesto por IEEE (1484.12.3).

²⁰ Un *cluster* es una jerarquía de dos niveles, que está formada por actividades y sub-actividades.

En la actualidad el estándar LOM define nueve categorías para la descripción de los objetos de aprendizaje. SCORM tomó esta definición y la ajustó a cada uno de sus componentes, determinando diversos perfiles de aplicación para cada uno de ellos, en los cuales se establece los elementos de metadatos obligatorios y opcionales para cada uno²¹. Las categorías de metadatos definidas por LOM son las siguientes:

- General: dentro de esta categoría se describen los aspectos generales de los objetos de aprendizaje, tales como su identificador, título, palabras claves o cobertura.
- Ciclo de vida: se utiliza para definir la historia y el estado actual del objeto de aprendizaje, y las personas o entidades que han aportado a su desarrollo.
- Meta-Metadatos: En esta categoría se describe la información acerca de los metadatos asociados al objeto de aprendizaje, como la versión del esquema de metadatos utilizada, y los aportes realizados a la definición de los metadatos.
- Técnico: Esta categoría agrupa los aspectos técnicos relacionados con el objeto de aprendizaje, tal como el formato, el tamaño y los posibles requerimientos para su instalación y visualización.
- Educativo: En esta categoría se definen los aspectos pedagógicos del objeto de aprendizaje, como el tipo de interactividad (expositivo, interactivo, etc.), la densidad semántica, la dificultad, el tiempo promedio de aprendizaje, entre otros.
- Derechos: Esta categoría define los aspectos legales y de derechos de autor del objeto de aprendizaje.

²¹ Para mayor información consultar el libro SCORM Content Aggregation Model, sección 4.5.2.

- **Relación:** Esta categoría se utiliza para describir las relaciones existentes entre el objeto de aprendizaje y otros objetos, como “basado en”, “es parte de”, “es base para”, etc.
- **Anotaciones:** En esta categoría se pueden definir comentarios acerca del uso del objeto de aprendizaje.
- **Clasificación:** Esta categoría permite definir el esquema de clasificación que utiliza el objeto de aprendizaje, dentro del cual se puede tener una descripción taxonómica, palabras claves, etc.

3.4.4.2. Ambiente de Ejecución

El objetivo del ambiente de ejecución de SCORM es proporcionar un medio para la interoperabilidad entre los objetos de contenidos, los SCO, y los LMS. Un requerimiento de SCORM es que el contenido educativo pueda ser utilizado en múltiples LMS, sin tener en cuenta las herramientas que se usen para crearlo o usarlo. Para que esto sea posible, debe existir un método común para iniciar la visualización de un objeto de contenido (launch), un mecanismo para que los contenidos se comuniquen con el LMS, y los elementos de datos predefinidos que sean interoperables entre el LMS y los objetos de contenido durante su ejecución. Estos tres componentes se muestran en la Figura 3 y se describen con más detalle a continuación.

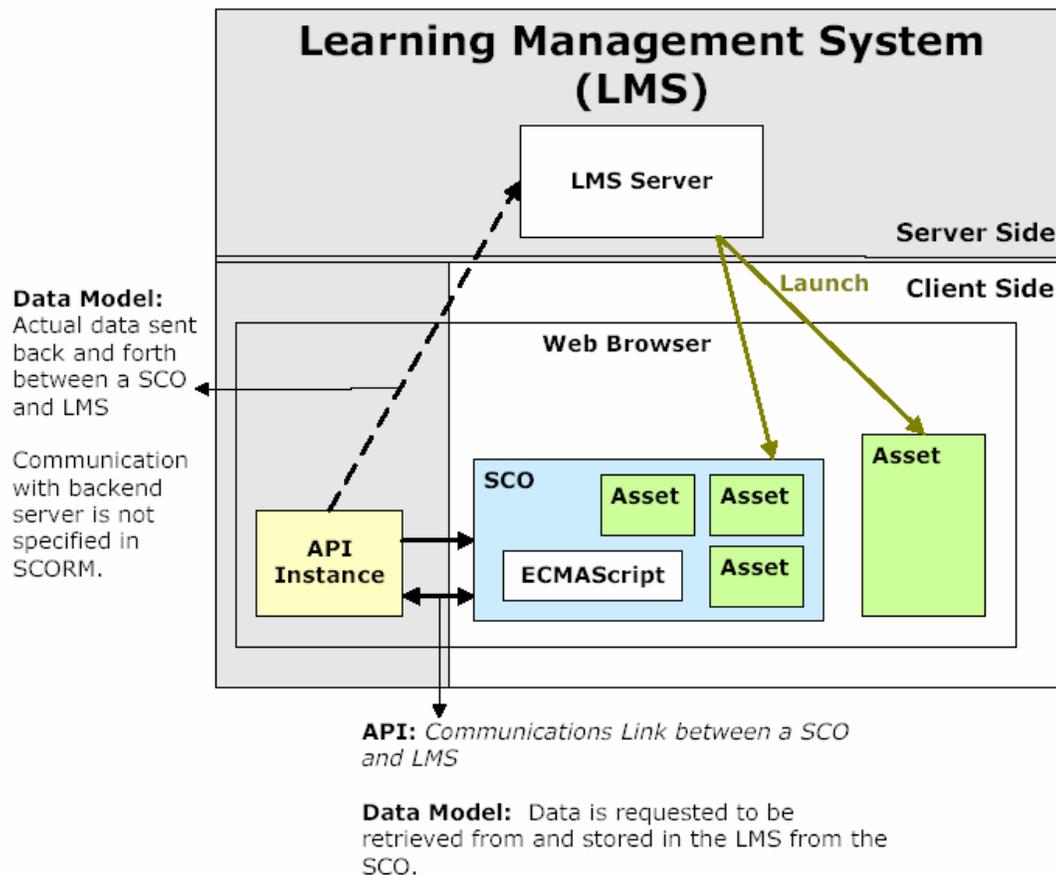


Figura 3. Modelo del ambiente de ejecución de SCORM

Launch

Es el mecanismo que proporciona el método común para que los LMS inicien la visualización un SCO basado en Web. Este mecanismo define los procedimientos y las responsabilidades que se deben seguir para establecer comunicación entre el objeto de contenido a visualizar y el LMS. El protocolo de comunicación está estandarizado a través del uso común del API, que interactúa con el SCO por medio de una interfaz de JavaScript, llamada ECMAScript. Es importante señalar que el LMS puede implementar la presentación de los SCO del modo que se desee, por ejemplo, desarrollando un mecanismo de adaptación ajustado al desempeño del usuario (mediante técnicas de aprendizaje adaptativo), o bien, puede delegar esa responsabilidad al cliente permitiéndole que navegue por el curso libremente a través de menús.

Interfaz de aplicación de programa (Application Program Interface - API)

Es el mecanismo para informar al LMS del estado del objeto de contenido, si se inició su visualización, si está finalizado o en estado de error. También se utiliza para intercambiar datos entre el LMS y los SCO, tales como tiempo total de ejecución (visualización), puntuación en una prueba, etc. EL API está conformado por un conjunto de funciones predefinidas que se ponen a disposición de los SCO, (por ejemplo LMSInitialize o LMSSetValue).

Modelo de Datos (Data Model)

Es una lista estandarizada de elementos usados para definir la información que se desea intercambiar, como el estado del objeto de contenido, o el progreso del estudiante dentro de su desarrollo. SCORM define que es responsabilidad del LMS mantener el estado de los datos requeridos a lo largo de diferentes sesiones de ejecución de un SCO. Existen diversos modelos de datos en desarrollo en varias organizaciones, que incluyen los datos sobre el perfil del estudiante, la información de estado del objeto de contenido, las iteraciones sobre los mecanismos de evaluación entre otros. El Modelo SCORM está basado directamente del CMI Data Model del AICC, que define una serie de categorías para los datos que deben transmitir los SCO hacia el LMS.

3.4.4.3. Secuenciación y Navegación (SN)

La secuenciación define un método estándar para representar el comportamiento deseado de una experiencia de aprendizaje, de forma que cualquier LMS pueda organizar el despliegue de las actividades en un orden consistente. Esta especificación basada en la IMS Simple Sequencing Specification, establece los comportamientos y las funcionalidades que debe implementar un LMS conforme con el modelo SCORM, para controlar la bifurcación y el flujo de las actividades que conforman una organización de contenidos, con base en la secuencia definida por el autor del paquete de contenidos, o el resultado de las interacciones entre el usuario y los objetos de contenido. El libro SN define además cómo se pueden generar y

manejar algunos eventos de navegación iniciados por el usuario o por el SCO, que determinan la próxima actividad a desplegar.

En este sentido, de acuerdo con SCORM, la secuencia de actividades obtenida por la secuenciación y la navegación determina una experiencia de aprendizaje única para cada usuario. A continuación se describen de forma general algunos conceptos relacionados con la secuenciación y navegación, para más detalles se puede consultar el SCORM Sequencing and Navigation Book (SCORM SN, Versión 1.3.1, 2004).

Definición de actividad de aprendizaje

La especificación IMS Simple Sequencing (SS) se basa en el concepto de actividades de aprendizaje. De acuerdo con el SS, una actividad de aprendizaje (o lección) se puede describir como la unidad básica de aprendizaje, que el usuario desarrolla mientras realiza su progreso en un contexto de aprendizaje (módulo, curso, etc.). Una actividad puede proporcionar una referencia a un recurso (SCO o Asset), o puede estar compuesta de otras sub-actividades. En la Figura 4 se puede apreciar una actividad llamada “Take Lesson”: Esta actividad está formada por tres sub-actividades: “Take a Pre-Test”, “Experience Content” y “Take a Final Test”.

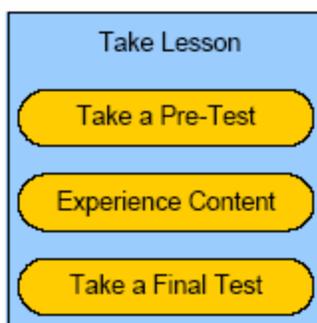


Figura 4. Esquema de una actividad de aprendizaje

Estructura de los contenidos y árbol de actividades

Los diagramas de estructura de los contenidos son una herramienta común que se utiliza para describir la relación de jerarquía de las actividades que forman parte de

una experiencia de aprendizaje. Este concepto se ve reflejado en la especificación de IMS Simple Sequencing en la forma de un árbol de actividades, que permite que el modelo de secuenciación y navegación describa los algoritmos de secuenciación y los comportamientos asociados para la presentación de actividades al usuario. En la Figura 5 se puede apreciar un árbol de actividades, cuya raíz es una actividad llamada *Course* (Curso). A su vez esta actividad se divide en tres sub-actividades: Module 1, Module 2 y Module 3, y así sucesivamente.

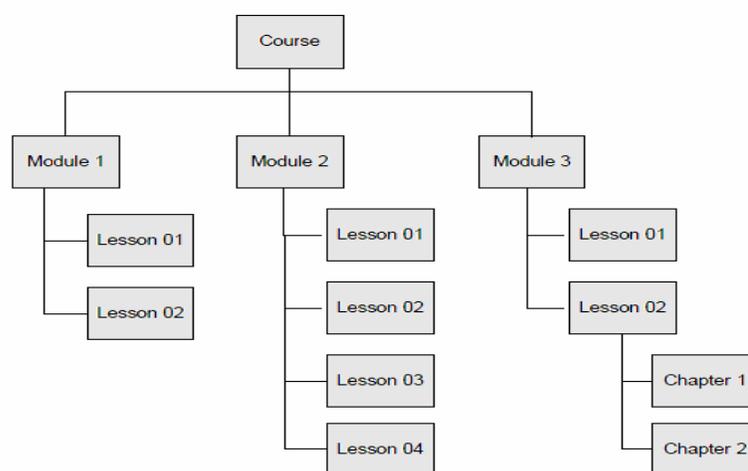


Figura 5. Estructura de un árbol de actividades

De acuerdo con el modelo planteado por la especificación SS de IMS, las actividades que poseen sub-actividades se denominan *Clusters* (Course, Module 1, Module 2, Module 3 y Lesson 02), mientras que las actividades que no contienen sub-actividades se denominan actividades hoja (Leaf Activities), o simplemente actividades (Activities). Un cluster sólo contempla actividades y sub-actividades, es decir dos niveles en la jerarquía (Figura 6). Desde este punto de vista, el árbol de actividades está formado por uno o más clusters, que contienen otros clusters o actividades hoja.

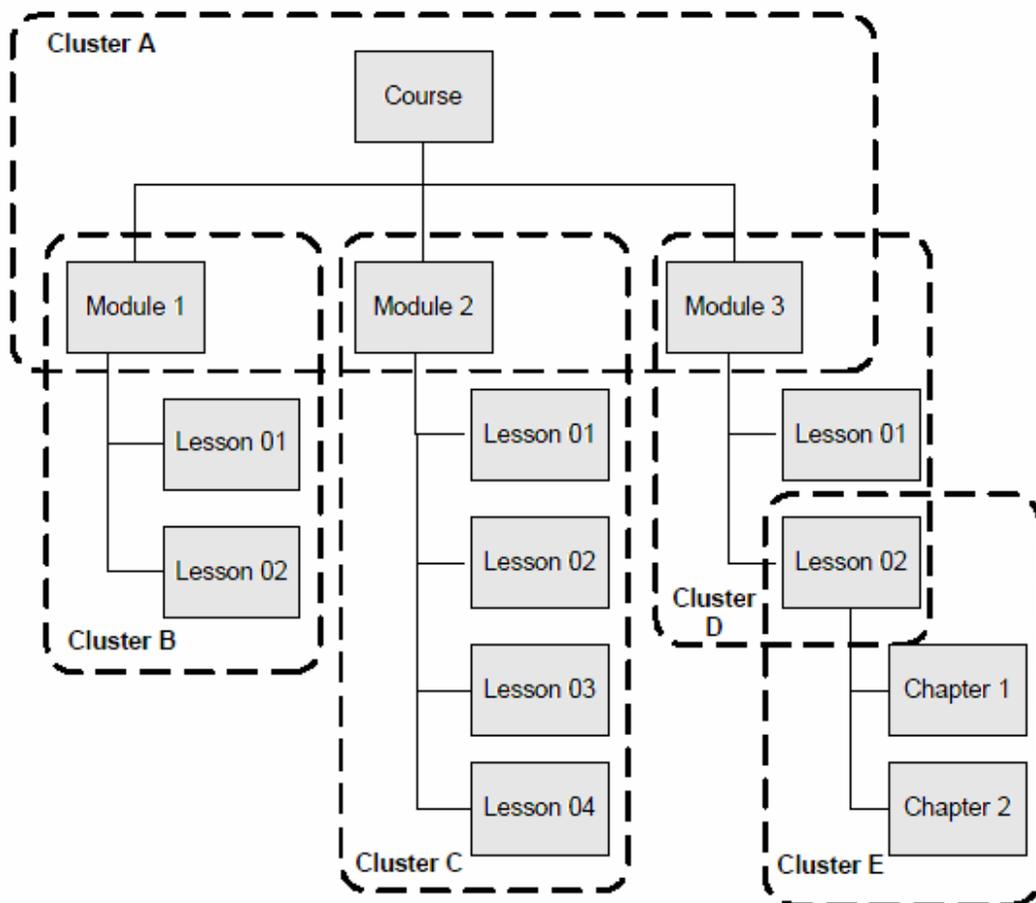


Figura 6. Estructura de clusters de un árbol de actividades

Objetivos de aprendizaje

Los objetivos de aprendizaje permiten almacenar información acerca del progreso del estudiante en una actividad específica, que puede ser utilizada para determinar el grado de completitud de una actividad de aprendizaje, y que puede afectar la secuenciación de un árbol de actividades. En esta dirección el modelo SCORM no establece ninguna interpretación específica al concepto de objetivo, en términos de competencias o nivel de apropiación del conocimiento, con el fin de permitir que los sean los creadores de los paquetes de contenidos quienes determinen el significado de la información almacenada en objetivos definidos dentro de un árbol de actividades.

De acuerdo con el modelo de secuenciación y navegación de SCORM, los objetivos poseen información local a una actividad de aprendizaje, pero pueden ser utilizados para almacenar valores compartidos entre actividades. Los objetivos que se utilizan para compartir información se denominan objetivos globales (global objectives), y pueden ser referenciados por una o más actividades. Esta referencia se puede dar en dos niveles: lectura, en la cual las actividades sólo pueden acceder a la información almacenada en el objetivo, y escritura, que permite que las actividades modifiquen la información del objetivo.

Modos de secuenciación

Los modos de secuenciación definidos en el modelo SCORM SN permiten que los desarrolladores de paquetes de contenidos establezcan las reglas para controlar el orden de presentación de las actividades a los usuarios. Estos modos de secuenciación se aplican únicamente a actividades que contengan sub-actividades, es decir a clusters. La Tabla 3 resume los modos de secuenciación más importantes definidos en el modelo Sequencing and Navigation de SCORM.

Tabla 3. Modos de secuenciación definidos por SCORM SN

Modo	Descripción
Choice	El usuario puede seleccionar cualquier actividad del árbol de contenidos. El LMS debe proporcionar algún mecanismo que le permita al usuario seleccionar la próxima actividad a revisar.
Choice Exit	Este modo de secuenciación determina si es posible elegir una actividad que no esté dentro del mismo cluster actual. Se utiliza para restringir las actividades que el usuario puede seleccionar, de forma que no pueda elegir una actividad que se encuentre muy "lejos" de la actividad actual en el árbol de actividades.
Flow	Indica si se permite la navegación dirigida por las actividades de un cluster. El LMS es responsable de ofrecer los mecanismos para que el usuario pueda revisar la siguiente actividad (Continue), o la actividad anterior (Previous)

Modo	Descripción
Forward Only	Este modo de secuenciación permite limitar la navegación de las actividades de un cluster, de forma que sólo se puede ir hacia adelante en el árbol de actividades. En este caso el LMS deberá restringir la solicitud de actividades anteriores (Previous).

Reglas de secuenciación

La especificación SCORM SN se basa en el modelo basado en reglas de la IMS SS Specification. Estas reglas se aplican a una actividad y se evalúan de acuerdo con la información obtenida a partir del seguimiento de las acciones del usuario, y de los objetivos asociados con las actividades. Cada regla consiste en un conjunto de condiciones, para las cuales se realiza una acción, que se ejecuta si la evaluación del conjunto de condiciones da como resultado un valor verdadero. La Figura 7 ilustra la estructura una condición de una regla de secuenciación. Obsérvese que las condiciones pueden estar negadas (if not).

Conditions	
Attempted	Objective Measure Known
Objective Status Known	Objective Measure Greater Than
Satisfied	Objective Measure Less Than
Activity Progress Known	Always
Completed	Attempt Limit Exceeded

if (not)

Figura 7. Condiciones de una regla de secuenciación

Las condiciones pueden ser combinadas utilizando los operadores lógicos and y or, para dar como resultado condiciones más complejas (por ejemplo *if activity progress known or attempted...*).

La Tabla 4 muestra una breve descripción de cada una de las condiciones que se pueden utilizar en una regla de secuenciación.

Tabla 4. Condiciones de una regla de secuenciación

Condición	Descripción
Satisfied	La condición se evalúa a verdadero si el estado del objetivo primario de la actividad está satisfecho.
Objective Status Known	Verdadero si el objetivo primario de la actividad ha registrado un progreso, sin importar si ha sido satisfecho o no.
Objective Measure Known	Se evalúa a verdadero si se ha establecido un valor para el objetivo primario de la actividad.
Completed	La condición se evalúa a verdadero si la actividad ha sido completada
Activity Progress Known	Verdadero si se ha establecido un progreso en el desarrollo de la actividad, sin importar si se ha completado.
Attempted	Verdadero si el usuario ha revisado la actividad, sin importar su progreso.
Attempt Limit Exceeded	Verdadero si el número de intentos de la actividad son mayores al límite establecido para ésta.
Never	Esta condición siempre evalúa a falso.

Por su parte, las acciones especifican el comportamiento que debe tener el LMS como respuesta a la evaluación de una condición. Estas acciones pueden ser de tres tipos: Pre-condición (Pre-condition), que se aplican en el momento de identificar la próxima actividad a entregar al usuario, Post-condición (Post-condition), que se aplican cuando finaliza un intento en una actividad, y de salida (Exit), que se aplican después que una sub-actividad de la actividad actual termina. La Figura 8 muestra las posibles acciones para una regla de secuenciación, según su tipo.

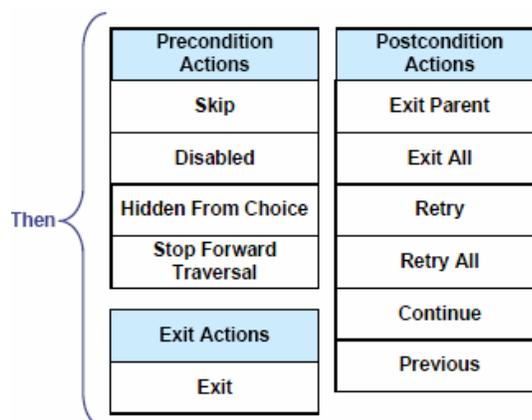


Figura 8. Acciones definidas para una regla de secuenciación

La Tabla 5 resume las diferentes acciones tomadas por el LMS en el caso que la evaluación de la condición para una regla sea verdadera.

Tabla 5. Acciones para una regla de secuenciación

Acción	Descripción
Skip	La actividad no se considera como candidata para ser entregada al usuario en el caso de una solicitud de secuenciación de flujo (flow).
Disabled	La actividad no podrá ser entregada al usuario en ningún caso.
Hidden from Choice	En este caso la actividad no podrá ser seleccionada mediante una solicitud choice.
Stop Forward Traversal	La actividad no permitirá que se entregue al usuario cualquier actividad que se encuentre delante de ella en su cluster.
Exit	Se deberá terminar el intento con la actividad de forma incondicional.
Exit Parent	El LMS deberá procesar una solicitud de Exit Parent
Exit All	El LMS deberá procesar una solicitud Exit All, causando que la sesión con el paquete de contenidos termine.
Retry	El LMS deberá procesar una solicitud Retry y comenzar un nuevo intento con la actividad.
Retry All	El LMS deberá procesar una solicitud Exit All, y posteriormente una solicitud Start, para comenzar una nueva sesión con el paquete de contenidos.
Continue	El LMS deberá procesar una solicitud Continue, para entregar una actividad ubicada después de la actividad actual.
Previous	El LMS deberá procesar una solicitud Previous, para ubicar una actividad anterior a la actividad actual.

3.5. REPOSITORIOS DIGITALES DE OBJETOS DE APRENDIZAJE

Los repositorios digitales surgieron como una respuesta a la necesidad de contar con un medio de organización, búsqueda y distribución de objetos de contenido

entre diferentes sistemas de gestión de aprendizaje²². A pesar que en la actualidad no se cuenta con una definición formal de repositorio, las diversas definiciones coinciden en que los repositorios deberán proporcionar un acceso transparente a la gran cantidad de objetos de contenido que existe en la actualidad, y que se incrementa de forma constante en la medida en que más instituciones adoptan y desarrollan programas de educación en línea.

En este sentido, los repositorios se crean bajo la premisa que existe una comunidad que desea ofrecer contenidos para que puedan ser reutilizados en diversos contextos educativos, lo que permite que las personas involucradas en los procesos de enseñanza y aprendizaje se enfoquen en la construcción de conocimiento a partir de los materiales y las experiencias existentes, en lugar de construir nuevamente los materiales con cada experiencia.

3.5.1. Definición de repositorio digital

Desde un punto de vista general, un repositorio digital es *una herramienta que permite el almacenamiento y la catalogación de objetos de contenido y sus metadatos asociados, para permitir su posterior búsqueda y utilización en diversas experiencias de aprendizaje*. Es importante resaltar la diferencia entre el término “repositorio” y el término “biblioteca” o “librería”, debido a que en ocasiones se utilizan de forma indiscriminada. El primer término hace referencia a una herramienta en la cual es posible que *cualquier persona* pueda contribuir con un nuevo objeto de contenido, que formará parte del conjunto de objetos de contenido existente; el segundo término hace referencia a una herramienta en la cual algunos usuarios “privilegiados” (los bibliotecarios) tienen la capacidad de decidir cuáles objetos de contenido estarán disponibles al público. A pesar que desde cierta perspectiva un repositorio puede ser considerado como una librería de objetos de

²² A pesar ser diferentes desde el punto de vista conceptual, los autores intercambian los términos “objeto de contenido” y “objeto de aprendizaje” de forma indiscriminada.

contenido, se prefiere el término “repositorio” para enfatizar el hecho de estos deben ser espacios abiertos a los aportes de la comunidad académica en general.

3.5.2. Iniciativas de repositorios digitales

Dadas las oportunidades que ofrecen los repositorios digitales de objetos de contenido, en la actualidad existe un gran número de implementaciones de repositorios, que establecen sus propias especificaciones, modelos de referencia e interfaces de servicios, ajustados a los requerimientos individuales establecidos para su creación²³. A continuación se presentan algunas de las iniciativas más importantes para la especificación de los repositorios digitales²⁴.

ARROW

La iniciativa ARROW (Australian Research Repositories Online to the World) fue iniciada por el Australian Department of Education Skills and Training (DEST), con el fin de explorar y probar las posibles soluciones para crear repositorios institucionales en la Universidad de Monash, la Swinburne University of Technology, la Universidad de New South Wales y la Biblioteca Nacional de Australia. Inicialmente ARROW está interesado en los medios digitales de resultados de investigación, como artículos electrónicos, libros y tesis digitales, con la intención de crear objetos de aprendizaje y recursos de investigación para fases posteriores del proyecto.

Johns Hopkins University Digital Knowledge Centre: A Technology Analysis of Repositories and Services

El Digital Knowledge Center de la Universidad Johns Hopkins, junto con la Universidad de Virginia, el Instituto de Tecnología de Massachussets (MIT) y la Sheridan Libraries Network, han comenzado una iniciativa que busca realizar un

²³ La descripción de algunas de estas iniciativas fue adaptada de HEERY, 2005.

²⁴ Una lista más completa de las implementaciones de repositorios digitales se puede encontrar en http://projects.aadicolab.org/repository-directory/repository_listing.asp

análisis y una evaluación de los servicios y el software que se utiliza en la educación en línea²⁵. La evaluación de estos sistemas se realizará con respecto a una serie de casos de uso, y el resultado de este estudio ofrecerá una tipología de los repositorios y sus usuarios.

Digital Library Federation (DLF)

La Federación de Librerías Digitales en convenio con la Universidad de Emory, la Universidad de Illinois-Urbana-Champaign y la Universidad de Michigan, están desarrollando un proyecto para diseñar la “segunda generación” del sistema de búsqueda OAI. El interés de este proyecto es promover una mejor educación a través de una forma más sencilla de búsqueda y recuperación de recursos digitales, al tiempo que se busca mejorar las capacidades de las bibliotecas, al construir servicios basados en colecciones de metadatos distribuidos.

Digital Academic Repositories (DARE)

Esta es una iniciativa conjunta de las universidades Holandesas, que tiene como objetivo que todos sus resultados de investigación sean accesibles a través de Internet. En la actualidad se cuenta con una implementación piloto de un servicio de búsqueda, y se está trabajando en el desarrollo de un sistema de administración de derechos de autor.

Multimedia Educational Resource for Learning and Online Teaching (MERLOT)

Esta iniciativa es un proyecto de código abierto, diseñado para los profesores y estudiantes de educación superior, que ofrece un servicio para almacenar los metadatos y las revisiones de materiales de aprendizaje almacenados en otros repositorios como ARIADNE, EdNA y NIME. El sistema permite que cualquier persona pueda realizar una contribución a las descripciones de los objetos de aprendizaje en un catálogo maestro.

²⁵ Disponible en <http://ldp.library.jhu.edu/repository.html>.

Connexions

Este proyecto liderado por la Rice University, almacena materiales y metadatos en un formato XML específico, que permite que sean fácilmente editados y ensamblados en secuencias de contenidos. Los contenidos almacenados dentro del repositorio están diseñados de tal forma que son independientes de cualquier plantilla, estilo o diseño de lecciones.

DSpace

Este proyecto del Massachusetts Institute of Technology (MIT) creó un repositorio que permite almacenar los resultados de las investigaciones de diversas organizaciones. Este sistema fue diseñado en colaboración con Hewlett-Packard, entre Marzo del 2000 y Noviembre del 2002²⁶. El diseño de este repositorio permite que participen entidades de diversos niveles, como escuelas, centros de investigación y universidades. Debido a que los requerimientos de estas instituciones pueden variar, DSpace permite configurar aspectos del sistema que permita la personalización de las políticas de autorización y propiedad intelectual.

Fedora

Este proyecto coordinado por la Universidad de Virginia y la Universidad de Cornell desarrolló la Arquitectura Flexible y Extensible para Repositorios de -Objetos Digitales (Flexible Extensible Digital Object and Repository Architecture –FEDORA), base del repositorio del mismo nombre, surgió como un repositorio de propósito general, cuyo objetivo es ofrecer una herramienta que puede ser utilizada para construir repositorios y bibliotecas interoperables. De acuerdo con los autores, la versión actual del repositorio soporta un millón de objetos de forma eficiente (FEDORA, 2003).

²⁶ La versión 1.2 de este software fue liberada en abril de 2004.

La versión actual del repositorio ofrece una interfaz basada en servicios web, que se divide en tres componentes:

- Un API de administración, que incluye las interfaces para administrar el repositorio, que incluyen operaciones de crear y mantener objetos digitales.
- Un API de acceso, que facilita el descubrimiento y la diseminación de objetos en el repositorio
- Una versión del sistema de acceso, implementada como un servicio web accesible a través de HTTP.

3.5.3. Interoperabilidad de repositorios digitales

La interoperabilidad de repositorios digitales tiene como objetivo eliminar las barreras presentadas en la búsqueda, la localización y el acceso a los objetos de contenido almacenados en diversos repositorios, que pueden estar implementados en diferentes formatos. En este sentido, y teniendo en cuenta que en la actualidad existe una gran cantidad de repositorios, algunas organizaciones han emprendido la labor de estudiar y desarrollar algunas estrategias que permitan la interoperabilidad entre los diferentes tipos de recursos y esquemas de metadatos que se utilizan dentro de cada repositorio. A continuación se presentan algunas de las principales iniciativas para la interoperabilidad de repositorios digitales de objetos de contenido²⁷.

3.5.3.1. National Science Digital Library Project - NSDL

Este proyecto tiene como objetivo construir una librería digital para la ciencia, las matemáticas, la ingeniería y la tecnología. Debido a que las colecciones de las diferentes librerías pueden variar en los tipos de datos, esquemas de metadatos,

²⁷ Otras iniciativas pueden ser consultadas en <http://metadata.cetis.ac.uk/specs/>

protocolos, esquemas de autenticación y modelos de negocios, NSDL propuso tres niveles de interoperabilidad: *federation*, que se refiere al uso de determinados estándares por parte de las librerías, *harvesting*, que permite capturar los metadatos de las diversas librerías, y *gathering*, que permite consultar y agrupar la información de las organizaciones que no participan formalmente en el programa NSDL. De otro lado, NSDL almacena los metadatos según el esquema establecido por Dublin Core²⁸ y utiliza el protocolo para recolección de metadatos (Protocol for Metadata Harvesting – PMH) propuesto por el *Open Archive Initiative*.

3.5.3.2. POOL

Uno de los principales objetivos del proyecto POOL fue construir una infraestructura para conectar repositorios homogéneos en una red Peer-to-Peer (P2P). Esta red estaba conformada por nodos que podían ser repositorios individuales (SPLASH), o repositorios empresariales (PONDs), y que se comunicaban a través del protocolo JXTA.

3.5.3.3. ELENA/Edutella

Este proyecto colaborativo entre entidades Europeas tiene como objetivo crear mediadores (brokers) de servicios educativos. Estos mediadores permiten utilizar diferentes servicios ofrecidos por LMS, repositorios y sistemas de videoconferencia. ELENA se encarga de construir un perfil dinámico de usuario, que se utiliza para ofrecer al estudiante los recursos adecuados, a partir de una gran variedad de recursos existentes. De esta forma ELENA construye una capa sobre una red Edutella, que a su vez tiene como objetivo conectar repositorios altamente heterogéneos, con diversos esquemas de metadatos y lenguajes de búsqueda.

²⁸ Esquema de datos simplificado de IEEE LOM. Para más información puede consultar <http://www.dublicore.org>.

3.5.3.4. IMS Digital Repository Interoperability Specification - DRI

En el 2003 el IMS Global Learning Consortium Inc. publicó la Especificación para la Interoperabilidad de Repositorios Digitales (Digital Repositories Interoperability Specification - DRI), con el objetivo de establecer un marco de trabajo general que permita la interoperabilidad de las diferentes implementaciones de repositorios. Para la especificación DRI, los repositorios digitales consisten en una colección de recursos que son accesibles a través de una red, sin un conocimiento previo de su estructura. Estos repositorios se encargan de almacenar los recursos de aprendizaje o los metadatos que los describen, pero esto no implica que ambos (metadatos y contenidos) deban estar almacenados en el mismo repositorio. DRI define además los roles de usuario, la arquitectura funcional de un repositorio, y una serie de servicios que los repositorios deben ofrecer.

3.5.3.5. EduSource Canada

Este proyecto EduSource reúne a los principales repositorios de Canadá para crear una infraestructura abierta que soporta un amplio rango de servicios, y que ofrece la posibilidad de que nuevos sistemas se conecten fácilmente a esta red. Para lograr este objetivo EduSource plantea un protocolo de comunicación llamado ECL, que se basa en la especificación IMS DRI y que utiliza servicios Web XML en su implementación.

3.5.3.6. CORDRA

El modelo de referencia CORDRA (Content Object Repository Discovery and Registration/Resolution Architecture, 2004) busca proporcionar las bases para establecer una federación de repositorios de objetos de contenido, sobre las tecnologías desarrolladas por los repositorios y las librerías digitales. La estrategia utilizada por esta iniciativa consiste en combinar los metadatos de varios repositorios en un repositorio central, unido con un proceso de registro de repositorios que permite buscar en el conjunto de los metadatos combinados de todos los repositorios. La iniciativa CORDRA está siendo coordinada por ADL (los

creadores de SCORM), con el soporte de la Corporación de Iniciativas Nacionales de Investigación y el Laboratorio de Arquitectura de Sistemas de Aprendizaje (LSAL) del Carnegie Mellon Institute. Los creadores de CORDRA están trabajando conjuntamente en Australia y Estados Unidos, para realizar una prueba del modelo de registro de repositorios de ADL, con el fin de liberar una versión de producción.

3.5.4. Modelos de interoperabilidad de repositorios digitales de contenidos

A partir de las experiencias obtenidas en el desarrollo de las iniciativas para promover la interoperabilidad de los repositorios, se han obtenido varios modelos que se basan principalmente en especificaciones funcionales que éstos deben cumplir. A continuación se describen los modelos de interoperabilidad más conocidos en la actualidad.

3.5.4.1. IMS Digital Repositories Interoperability Specification²⁹

El trabajo de la especificación IMS DRI se centró en identificar las funciones comunes de los repositorios que pueden ser implementadas como servicios, con el fin de proporcionar una interfaz común a la cual pueden tener acceso los LMS, las aplicaciones de usuario e incluso otros repositorios. Estos servicios cubren la autenticación y autorización, la búsqueda, localización y recuperación de objetos de contenido, y el manejo de las preferencias de los usuarios, ente otros. La Figura 9 muestra la arquitectura funcional propuesta por IMS DRI.

²⁹ Adaptado de IMS Digital Repositories Interoperability - Core Functions Information Model. 2003. IMS Global Consortium. http://www.imslobal.org/digitalrepositories/driv1p0/imsdri_infov1p0.html

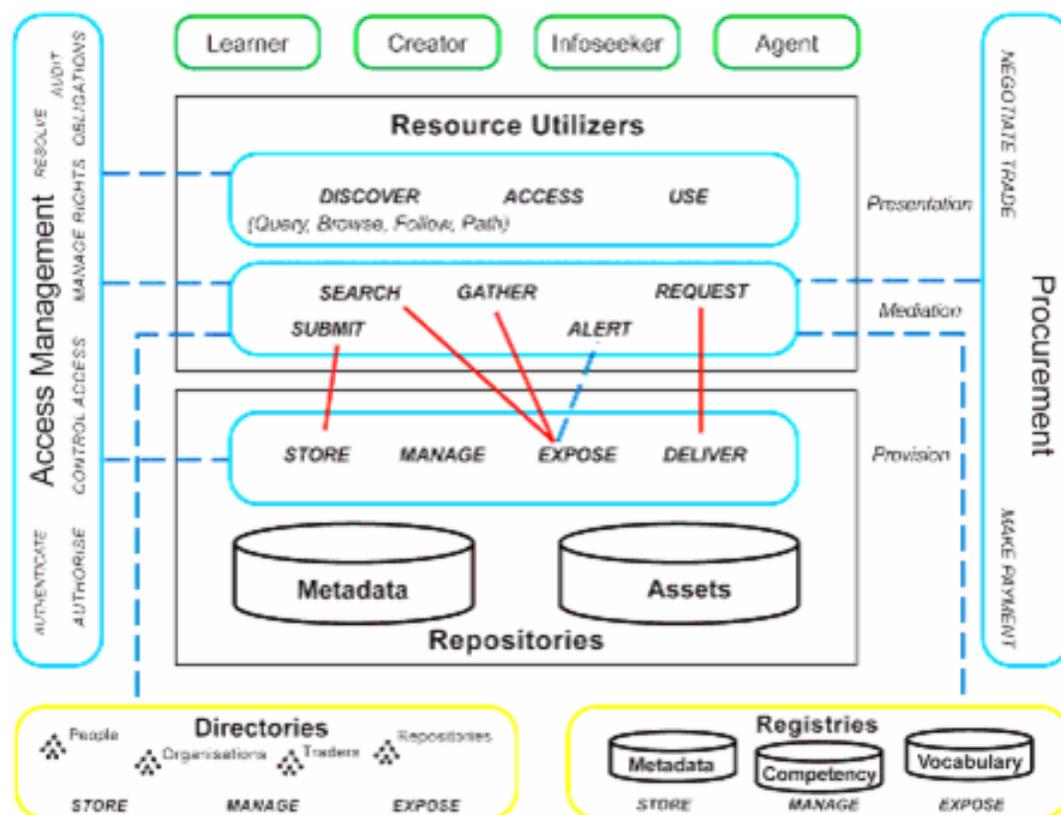


Figura 9. Arquitectura funcional definida en DRI

En la Figura 9 se puede observar la funcionalidad básica de un repositorio, representada por los servicios store (almacenamiento), manage (administración), expose (publicación) y deliver (entrega). En segundo nivel se encuentran los “utilizadores de los recursos”, que son sistemas o personas que hacen uso de los recursos. Estos “utilizadores” se pueden dividir en Learner (usuario final de los recursos), Creator, Infoseeker (elemento recopilador de información) y Agent. En la actualidad la especificación DRI se enfoca en el estudio de los servicios centrales, que se explicarán a continuación.

Search/Expose

El modelo de Search/Expose define los mecanismos de búsqueda sobre los metadatos asociados con los objetos de contenido, a través de los repositorios (ver Figura 10).

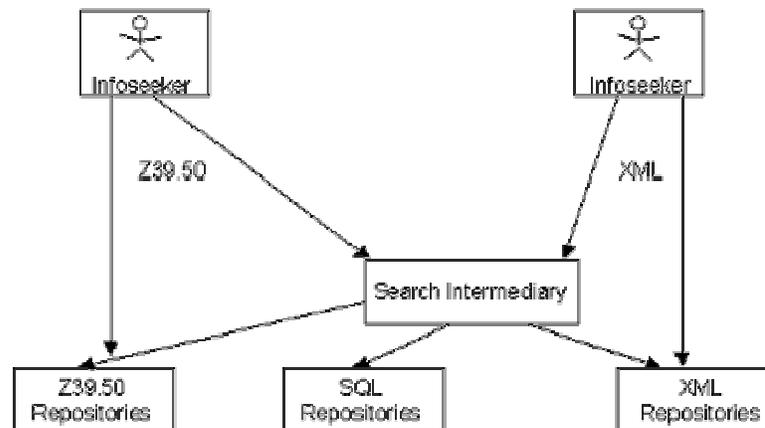


Figura 10. IMS DRI - Modelo Search/Expose

La característica principal de este modelo es que permite realizar búsquedas en diferentes tipos de repositorios, cada uno con su propia especificación. Estas búsquedas se pueden realizar utilizando diversas especificaciones, como XQuery, o Z39.50.

Gather/Expose

Este modelo define los mecanismos para solicitar los metadatos expuestos por los repositorios, y su posterior agregación para crear nuevos repositorios de metadatos (ver Figura 11).

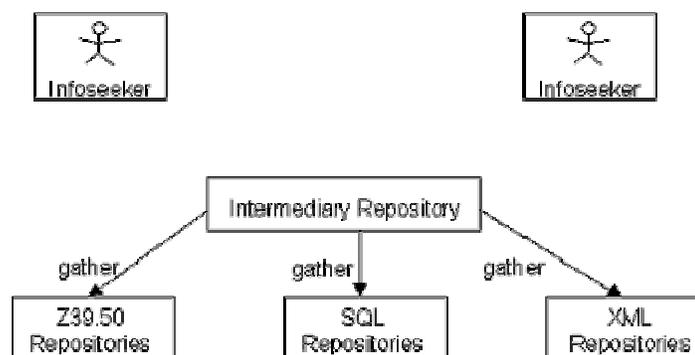


Figura 11. IMS DRI - Modelo Gather/Expose

En la Figura 11 se puede observar el funcionamiento del modelo Gather/Expose. Este componente puede interactuar con los repositorios para solicitar metadatos (Pull), o se pueden suscribir a un servicio de notificación, para recibir las últimas novedades sobre el repositorio (Push). Estos nuevos repositorios podrán ofrecer los servicios de Search/Expose y Alert/Expose.

Alert/Expose

Esta función se deja por fuera del alcance de la fase 1 de la especificación DRI.

Submit/Store

Esta funcionalidad se refiere a la forma como un objeto de aprendizaje (o sus metadatos) se agregan a la colección del repositorio. La localización desde la cual se mueve un objeto hacia el repositorio puede ser otro repositorio, un LMS, y en general cualquier dirección accesible por Internet (Figura 12)

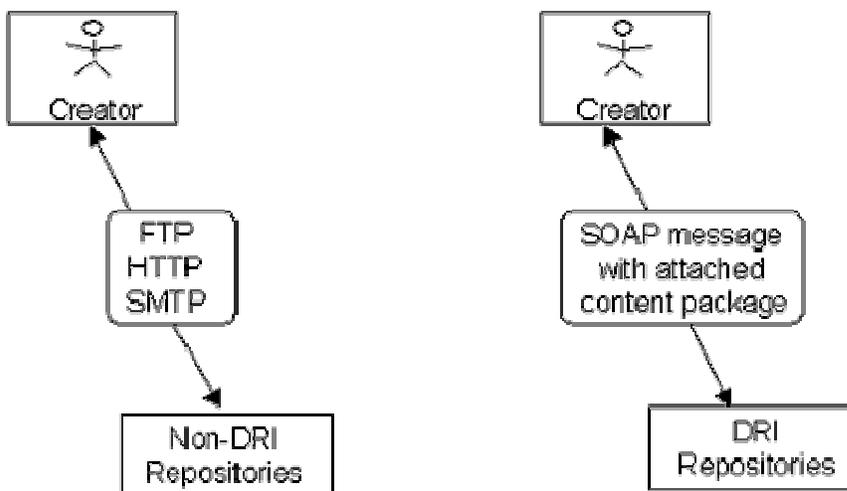


Figura 12. IMS DRI - Modelo Submit/Store

Para el caso de repositorios que no siguen la especificación DRI, el medio de transferencia preferido es FTP. En el caso de los repositorios más recientes se hace uso de la Especificación para el Empaquetamiento de Contenidos de IMS (IMS Content Packaging), para enviar el paquete de contenidos adjunto a un mensaje

SOAP. El paquete de contenidos es un archivo comprimido que contiene los objetos de aprendizaje, su registro de metadatos y un archivo de manifiesto que describe los objetos de aprendizaje dentro del paquete.

Request/Deliver

El componente funcional Request permite que un utilizador de recursos solicite el acceso a un objeto de aprendizaje, localizado a través de la función Search o Alert. Por otro lado la función Deliver se refiere a la respuesta del repositorio para proporcionar acceso al recurso.

3.5.4.2. EduSource³⁰

El proyecto EduSource busca formar una red compuesta por los repositorios de objetos de aprendizaje de Canadá, implementada en forma de una serie de servicios de aprendizaje. Para lograr este objetivo se creó el protocolo ECL (EduSource Communication Language), que se ajusta a la especificación IMS DRI. Los principales componentes de la arquitectura propuesta por EduSource se describen a continuación.

Repositorios de tipo servidor

Estos repositorios generalmente proporcionan acceso a su funcionalidad a través de un portal Web, que incluye los servicios para crear y visualizar objetos de aprendizaje y editar sus metadatos asociados. El modelo EduSource reconoce que pueden existir repositorios con diferentes esquemas de metadatos.

Repositorios Peer-to-peer

En este tipo de repositorios los recursos se almacenan en el sistema de archivos de los computadores clientes, y se utiliza un protocolo Peer-to-peer para su búsqueda y

³⁰ Adaptado a partir de HATHALA, Marek. RICHARDS, Griff y otros. 2003.

localización. Este protocolo incluye también los mecanismos para transferir objetos de aprendizaje entre los clientes peer-to-peer.

Repositorios de metadatos recolectados (Repositories of Harvested Metadata)

Este tipo de repositorios almacenan los metadatos de varios repositorios en una localización central, y las búsquedas se realizan sobre la colección centralizada de los metadatos. Con el fin de optimizar el proceso de búsqueda, los repositorios centralizados pueden almacenar los metadatos a partir de las búsquedas anteriores y sólo conducir nuevas búsquedas sobre toda la colección en el caso de ser necesario. Esta clase de repositorios funciona bien cuando los repositorios que conforman la colección poseen el mismo esquema de metadatos.

Otros repositorios

El proyecto EduSource contempla aquellos repositorios y redes de objetos de aprendizaje que se encuentran fuera de su dominio, al proporcionar un software de capa de comunicación que permite el acceso a estos repositorios.

Lenguaje de Comunicación de EduSource (EduSource Communication Language – ECL)

Este protocolo de comunicación desarrollado por EduSource implementa la especificación IMS DRI, y está construido como una serie de servicios web XML que brindan una interfaz común a los repositorios de diferentes tipos. Esta interfaz de servicios puede ser implementada por los repositorios que desean ingresar a la red de EduSource, y que deben ser incluidos en un registro central. La Tabla 6 muestra la definición de los servicios ofrecidos por EduSource.

Tabla 6. Servicios ofrecidos por la implementación de ECL

Servicio	Descripción
Expose	Este servicio se utiliza por los proveedores de servicios de aprendizaje para regresar los resultados de los servicios search, gather y alert.

Servicio	Descripción
Gather	Este servicio permite realizar solicitudes Gather a los repositorios de la red de EduSource.
Gateway Gather	Este servicio permite que los miembros de EduSource realicen solicitudes Gather a repositorios externos a la red de EduSource. Los resultados de estas solicitudes se transforman por defecto al esquema de metadatos IEEE LOM
Search	El protocolo ECL utiliza XQuery para realizar las búsquedas. Para aquellos repositorios que no soportan XQuery, se utilizan una serie de plantillas de búsqueda.
Alert	Cuando un repositorio recibe nuevos metadatos que se ajustan a los parámetros dados en el momento de su suscripción, envía un mensaje Alert a los demás repositorios de la red.
Submit	Función para mover un objeto de aprendizaje dentro de un repositorio.
Store	Este servicio es llamado por los proveedores de servicios para retornar los resultados de una función Submit.
Request	Proporciona una función para solicitar la entrega de un objeto de aprendizaje a un cliente. El protocolo de transferencia puede ser SOAP o FTP.
Deliver	Este servicio es llamado por los proveedores de servicios para retornar los resultados de una solicitud Request.

3.5.4.3. CORDRA – Content Object Repositories Discovery and Resolution Architecture

De acuerdo con sus autores, CORDRA (Arquitectura para la Resolución y el Descubrimiento de Repositorios de Objetos de Contenido) es *“Un modelo abierto y basado en estándares, sobre cómo diseñar e implementar sistemas de software que tienen como propósito descubrir, compartir y reutilizar contenidos de aprendizaje mediante el establecimiento de federaciones interoperables de repositorios de contenidos de aprendizaje”*³¹. Este modelo brinda soporte a la federación de distintos repositorios de

³¹ Adaptado a partir de REHAK, Daniel y otros. A Model and Infrastructure for Federated Learning Content Repositories, 2005.

contenidos, que se combinan en una fuente central de búsqueda y localización de contenidos³² (ver Figura 13).

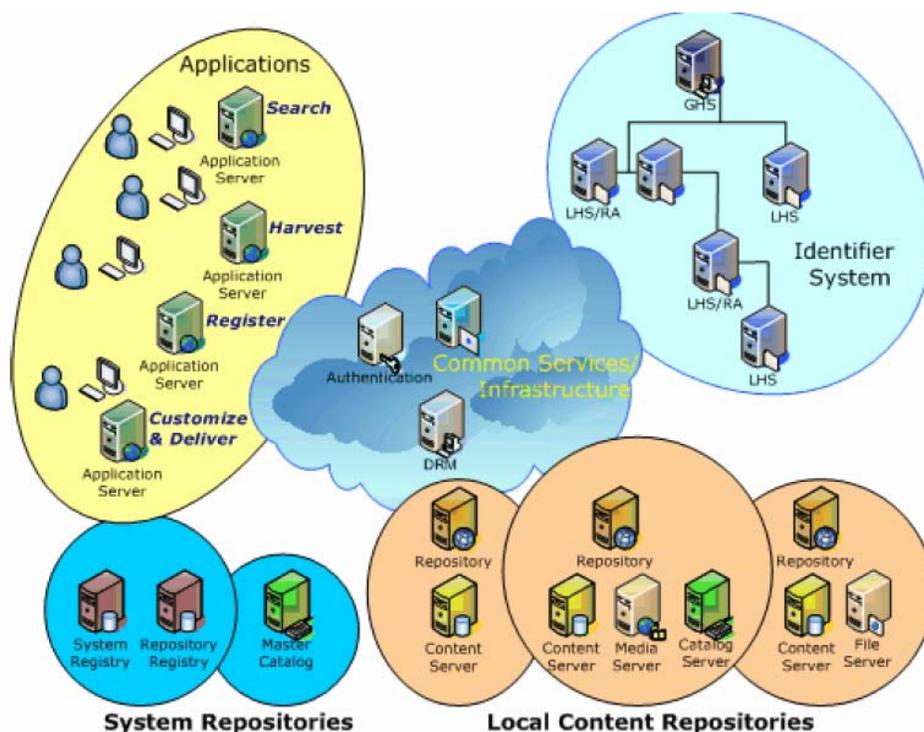


Figura 13. Modelo de federación de repositorios de CORDRA

Según el modelo CORDRA, el contenido se almacena en repositorios locales de contenidos (Local Content Repositories), dentro de los cuales es administrado de siguiendo sus políticas internas. Estos repositorios se registran dentro de la federación, con el propósito de permitir la localización, el acceso y la administración de sus recursos. El registro de las federaciones está compuesto por una colección repositorios del sistema (System Repositories), que mantienen un catálogo maestro de los objetos de contenido y sus metadatos, la lista de los repositorios que se encuentran registrados en la federación, e información adicional, propia del sistema.

³² La “federación” hace referencia a “delegación de responsabilidades”, en el sentido en que el catálogo central delega las acciones (búsqueda, localización, entrega) a los repositorios que forman parte de ella.

Por otra parte, un sistema de identificación proporciona una infraestructura para la identificación y el registro de objetos. A su vez, la infraestructura común de servicios ofrece los mecanismos centrales de administración, seguridad, manejo de derechos de autor, procesamiento de reglas de negocio, entre otros.

Finalmente, se utilizan las interfaces de usuario final y los sistemas de aplicación para catalogar, buscar, administrar y entregar los objetos de contenido. Estas funcionalidades se basan en una serie de servicios de valor agregado, tales como búsqueda, localización, autoría, personalización, entrega, entre otros.

3.6. DESARROLLO DE APLICACIONES BASADAS EN SERVICIOS WEB XML

Los servicios Web ofrecen un nuevo paradigma para construir aplicaciones distribuidas³³. De acuerdo con la W3C, en la actualidad el mayor uso que se le da a Internet es el acceso interactivo a documentos y aplicaciones por parte de los usuarios humanos, que utilizan navegadores web o varias clases de reproductores para visualizar la información disponible. Los servicios Web permiten expandir este uso de Internet hacia el desarrollo de aplicaciones *Inteligentes*, que utilizan diversas funcionalidades disponibles a través de Internet, para cumplir con las tareas para las cuales fueron creadas.

La definición de servicio web, de acuerdo con Cerami (2002) es:

Un servicio web es cualquier servicio o funcionalidad disponible a través de Internet, que utiliza un sistema de mensajes estándar basado en XML, y que no está ligado al sistema operativo o al lenguaje de programación.

³³ Ver Cerami, 2002.

Esta definición revela uno de los aspectos más importantes de los servicios web: al estar contruidos sobre estándares independientes de la plataforma de desarrollo, los servicios web ofrecen las bases para la creación de una nueva generación de aplicaciones, en las cuales la información está totalmente desacoplada de su lógica de manejo y de su presentación. En este sentido es posible que las aplicaciones Web, las aplicaciones para dispositivos móviles y las aplicaciones de usuario final compartan una base común de funcionalidad, la cual estará disponible a través de Internet. De esta forma los servicios web ofrecen la posibilidad de crear una red de aplicaciones automatizadas, esenciales para el desarrollo de la red semántica³⁴.

3.6.1. Arquitectura de un servicio Web

La arquitectura general de un servicio web se muestra en la Figura 14. Esta arquitectura está compuesta por tres componentes principales: El proveedor del servicio, el registro de servicios y el usuario del servicio. Los usuarios de la aplicación se incluyen en la figura para mayor claridad, pero no forman parte de la arquitectura; para ellos el hecho de que la aplicación que están ejecutando haga uso de servicios Web es irrelevante.

³⁴ En efecto Tim Berners Lee, uno de los creadores de Internet, señala que los servicios web son una actualización de la visión de la red semántica. Para mayor información consultar <http://www.sciam.com/2001/0501issue/0501berners-lee.html>

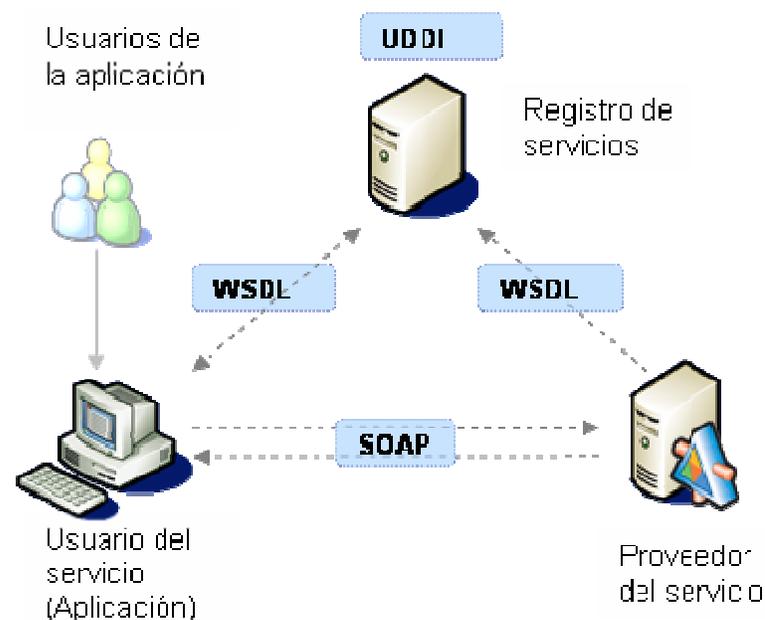


Figura 14. Arquitectura de un servicio Web

Los componentes principales de la arquitectura de un servicio web se describen a continuación:

- **Proveedor de servicios web:** Este componente implementa la funcionalidad del servicio en una plataforma de software específica. Para permitir que el servicio esté disponible en Internet, el proveedor realiza una descripción de la estructura del servicio por medio de WSDL (Lenguaje de Descripción de Servicios Web - Web Service Description Language). Esta descripción puede estar almacenada en el mismo proveedor del servicio, o en un componente especializado en el registro de servicios Web.
- **Registro de servicios:** Este componente es un directorio centralizado de los servicios disponibles. Su función consiste en ofrecer un sitio en el cual los desarrolladores pueden publicar nuevos servicios, o buscar dentro de los servicios existentes. Este registro utiliza UDDI (Descripción, Descubrimiento e Integración Universal - Universal Description, Discovery and Integration) como mecanismo de descripción de los diferentes proveedores de servicios.

- **Usuario del servicio**³⁵: Este componente busca dentro del registro de servicios, para localizar aquellos servicios que le son útiles para cumplir con su tarea. Una vez que los servicios han sido localizados, se pone en contacto con los proveedores de servicio, por medio de solicitudes XML. El protocolo de comunicación entre el usuario del servicio y el proveedor de servicio se denomina SOAP (Protocolo Simple de Acceso a Objetos - Simple Object Access Protocol).

La arquitectura anterior puede ser extendida en diferentes niveles; por ejemplo un usuario (aplicación) puede utilizar servicios de diferentes proveedores, quienes a su vez pueden hacer uso de los servicios ofrecidos por terceros (ver Figura 15).

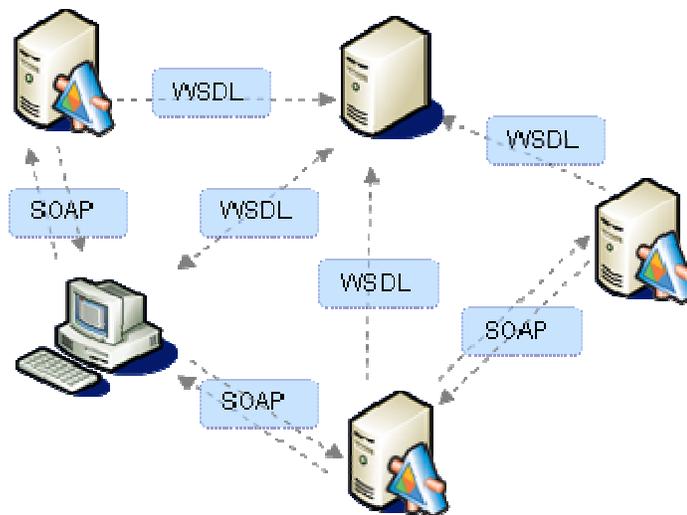


Figura 15. Modelo extendido de un servicio Web

A continuación se realiza una descripción general del protocolo SOAP y la especificación WSDL.

³⁵ A estas aplicaciones se les denomina también “consumidores de servicios”, o “service consumers”.

3.6.1.1. Simple Object Access Protocol – SOAP

Este protocolo de tipo Solicitud-Respuesta basado en XML, permite el intercambio de información entre aplicaciones a través de Internet. Una ventaja de SOAP con respecto a otros protocolos de *middleware*³⁶ es que los mensajes SOAP se transmiten por medio de HTTP, lo cual brinda la posibilidad de que las aplicaciones heterogéneas establezcan un mecanismo común de intercambio de información aún si se encuentran detrás de cortafuegos corporativos (*Firewalls*). El Fragmento de código 1 muestra una solicitud (mensaje) SOAP, para realizar una consulta sobre un servicio de clima (Weather Service)³⁷. Por mayor claridad se emiten los encabezados del mensaje SOAP.

Fragmento de código 1. Esquema de una solicitud SOAP

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://www.w3.org/2001/09/soap-envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:getWeather
      xmlns:ns1="urn:examples:weatherservice"
      SOAP-ENV:encodingStyle="http://www.w3.org/2001/09/soap-
encoding/">
      <zipcode xsi:type="xsd:string">10016</zipcode>
    </ns1:getWeather>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Esta solicitud invoca al método `getWeather`, al cual le pasa como parámetro el código de residencia (`zipcode`), con el valor de 10016. Se debe notar que un mensaje es en esencia un archivo XML, cuya especificación debe cumplir con el estándar SOAP.

³⁶ Puede ser definido como "Arquitectura software que conecta dos aplicaciones separadas": Otros ejemplos de Middleware son RMI, CORBA y DCOM.

³⁷ Tomado de CERAMI, 2002.

Por su parte, la respuesta a esta solicitud también es un mensaje SOAP. En este caso el servicio Web responde con el resultado de la consulta para el código ZIP 10016, que es un número entero (65).

Fragmento de código 2. Mensaje de respuesta a la solicitud SOAP

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://www.w3.org/2001/09/soap-envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:getWeatherResponse
      xmlns:ns1="urn:examples:weatherservice"
      SOAP-ENV:encodingStyle="http://www.w3.org/2001/09/soap-
      encoding/">
      <return xsi:type="xsd:int">65</return>
    </ns1:getWeatherResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Estructura de un mensaje SOAP

Un mensaje SOAP contiene un elemento obligatorio llamado sobre (Envelope), dentro del cual se encuentran las dos secciones principales: Encabezado (Header), y cuerpo del mensaje (Body). La Figura 16 muestra la estructura típica de un mensaje SOAP.

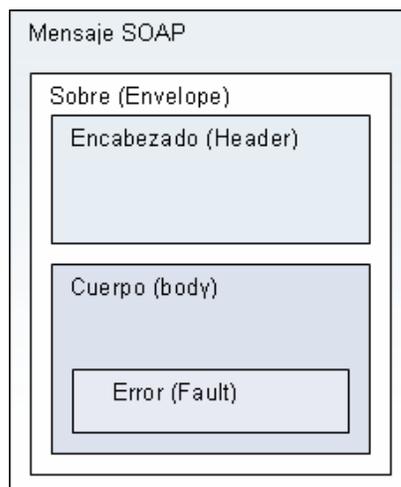


Figura 16. Estructura de un mensaje SOAP

La sección encabezado (header) ofrece un espacio en el cual se pueden expresar aquellos requerimientos que dependen de la aplicación, y que no están contemplados dentro del estándar. Este elemento se utiliza con frecuencia para almacenar firmas digitales o números de cuenta para aquellos servicios que utilizan pago electrónico. A pesar que en la actualidad este elemento no es muy utilizado, se estima que en el futuro servirá para ofrecer mecanismos de autenticación, manejo de transacciones y autorizaciones de pagos en línea.

Por su parte, el elemento Body encapsula la “carga” principal del mensaje SOAP (payload). En el interior de este elemento se tiene un único elemento, que representa una solicitud o una respuesta. Dentro de este elemento se encuentran los parámetros como sub-elementos XML, y para cada uno de ellos se especifica un tipo de datos. Finalmente, el elemento Fault se especifica dentro del cuerpo del mensaje en el caso que exista un error en la llamada al servicio Web. Este mensaje incluye los elementos faultCode, faultString y faultActor.

3.6.1.2. Web Service Description Language – WSDL

El Lenguaje de Descripción de Servicios Web es una especificación que permite definir los servicios utilizando una gramática estándar XML. Un documento WSDL contiene información acerca de las funciones públicas de un servicio web, los tipos de datos para las solicitudes y las respuestas, el transporte y el protocolo a utilizar, y su localización física (una dirección de Internet). En palabras de Cerami (2002), un documento WSDL representa un contrato independiente de plataforma entre los usuarios y los proveedores de servicios, y se utiliza principalmente para describir servicios SOAP.

Estructura de un documento WSDL

Un documento WSDL está dividido en seis elementos, que se muestran en la Figura 17.

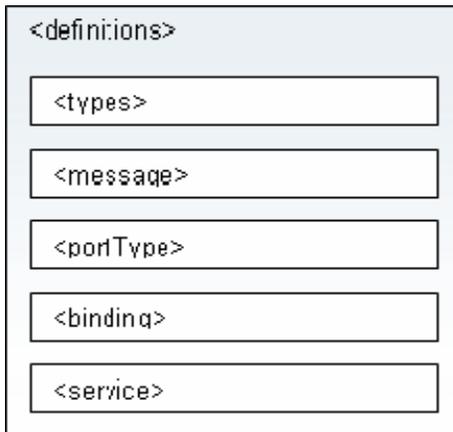


Figura 17. Estructura de un documento WSDL

A continuación se presenta una descripción de los elementos principales de un documento WSDL.

- **definitions:** Este elemento debe ser la raíz de cualquier documento WSDL. En él se define el nombre del servicio Web y algunos nombres de espacios de XML utilizados en los demás elementos del documento.
- **types:** Dentro de este elemento se definen todos los tipos de datos que se utilizan para transferir la información entre el usuario y el proveedor del servicio.
- **message:** Este elemento define un mensaje de una sola dirección, sin importar si es una solicitud o una respuesta. Para cada mensaje se define su nombre, y contiene cero o más partes (`messagePart`), que se refieren a los parámetros de invocación de una rutina o los valores que ésta debe retornar
- **portType:** Este elemento agrupa los mensajes definidos anteriormente en operaciones completas. Las operaciones pueden ser de cuatro tipos: `One-way`, en la cual el usuario envía un mensaje al proveedor del servicio y no espera respuesta, `request-response`, en la cual el cliente envía un mensaje y espera una respuesta del proveedor, `solicit-response`, en la cual el proveedor envía un mensaje al usuario y éste envía una respuesta al servidor, y `notification`, en la cual el proveedor envía un mensaje al usuario y no espera respuesta.

- **binding:** En esta sección se define específicamente cómo va a ser implementado el servicio. El parámetro más importante en este elemento es el transporte a utilizar, que puede ser HTTP GET, HTTP POST o SOAP.
- **service:** Esta sección especifica la localización física del servicio, en forma de una URL.

El Fragmento de código 3 muestra un ejemplo de un documento WSDL, que en este caso define un servicio de tipo request-response llamado HelloService.

Fragmento de código 3. Ejemplo de un documento WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="HelloService"
  targetNamespace="http://www.ecerami.com/wsdl/HelloService.wsdl"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://www.ecerami.com/wsdl/HelloService.wsdl"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <message name="SayHelloRequest">
    <part name="firstName" type="xsd:string"/>
  </message>
  <message name="SayHelloResponse">
    <part name="greeting" type="xsd:string"/>
  </message>
  <portType name="Hello_PortType">
    <operation name="sayHello">
      <input message="tns:SayHelloRequest"/>
      <output message="tns:SayHelloResponse"/>
    </operation>
  </portType>
  <binding name="Hello_Binding" type="tns:Hello_PortType">
    <soap:binding style="rpc"
      transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="sayHello">
      <soap:operation soapAction="sayHello"/>
      <input>
        <soap:body
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="urn:examples:helloservice"
          use="encoded"/>
      </input>
      <output>
        <soap:body
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="urn:examples:helloservice"
          use="encoded"/>
      </output>
    </operation>
  </binding>
</definitions>
```

```

        </output>
    </operation>
</binding>
<service name="Hello_Service">
    <documentation>WSDL File for HelloService</documentation>
    <port binding="tns:Hello_Binding" name="Hello_Port">
        <soap:address
            location="http://localhost:8080/soap/servlet/rpcrouter"/>
        </port>
    </service>
</definitions>

```

Un esquema que resume la estructura de este documento WSDL se muestra en la Figura 18. Se debe notar que para este servicio no se definen tipos de datos especiales, por lo cual la sección <types> no es necesaria.

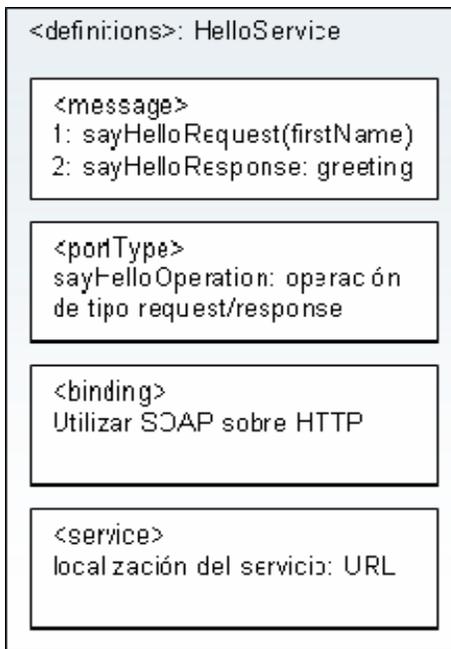


Figura 18. Estructura resumida del documento WSDL

4. CREACIÓN DE MATERIALES EDUCATIVOS BASADOS EN OBJETOS DE APRENDIZAJE

La creación de materiales educativos basados en objetos de aprendizaje aporta una nueva perspectiva al desarrollo de contenidos, desde la cual el proceso de enseñanza se ve favorecido por el uso de materiales con un alto grado de especialización, que integrados de acuerdo con unos objetivos específicos ofrecen la posibilidad de obtener experiencias más flexibles y personalizadas. En este sentido, el desarrollo de los materiales requiere de la participación de profesionales de diversas disciplinas, que deberán tener en cuenta los principios pedagógicos y de diseño que garanticen el aprovechamiento efectivo de los materiales desarrollados.

Sin embargo, el desarrollo y uso objetos de aprendizaje para crear los materiales educativos no garantiza el éxito de cualquier experiencia de educación en línea. Además es necesario realizar un análisis más profundo acerca del aporte de este tipo de materiales en el desarrollo de procesos de apropiación y generación de nuevo conocimiento. De acuerdo con Aguilar, Corredor y otros (2002)³⁸:

Los lineamientos para la selección y organización de contenidos, así como las actividades a realizar, [...], los medios y las estrategias para lograr aprendizajes significativos deben clarificarse dentro de los modelos pedagógicos y los currículos, de forma que orienten las características de la acción educativa [...]. Atendiendo a estos retos, se hacen necesarias la reflexión y construcción de saber alrededor del papel que debemos dar a las mediaciones del proceso educativo, de modo que realmente contribuyan a la formación de los estudiantes.

Con el fin de garantizar la interoperabilidad de los contenidos desarrollados, es necesario seleccionar un estándar para el diseño de los materiales, así como una

³⁸ AGUILAR, Esperanza. CORREDOR, Martha. Aula Virtual. Una alternativa en educación superior. 2003. Pág. 24.

estrategia para su desarrollo, catalogación y almacenamiento. A continuación se presenta el estándar seleccionado en este trabajo de investigación y la estrategia para la creación de los contenidos basada en este estándar.

4.1. SELECCIÓN DE UN ESTÁNDAR PARA LA CREACIÓN DE CONTENIDOS BASADOS EN OBJETOS DE APRENDIZAJE

Dadas las condiciones de la evolución de las especificaciones y los estándares utilizados en la educación en línea, el desarrollo de este trabajo de investigación se basó en SCORM como “estándar” para la creación de contenidos basados en objetos de aprendizaje. Es importante resaltar que SCORM no es un estándar en el sentido literal de la palabra, es un modelo de referencia que integra diversos estándares y especificaciones para definir los requerimientos que deben cumplir las aplicaciones que desean hacer uso de ellos (Figura 19).

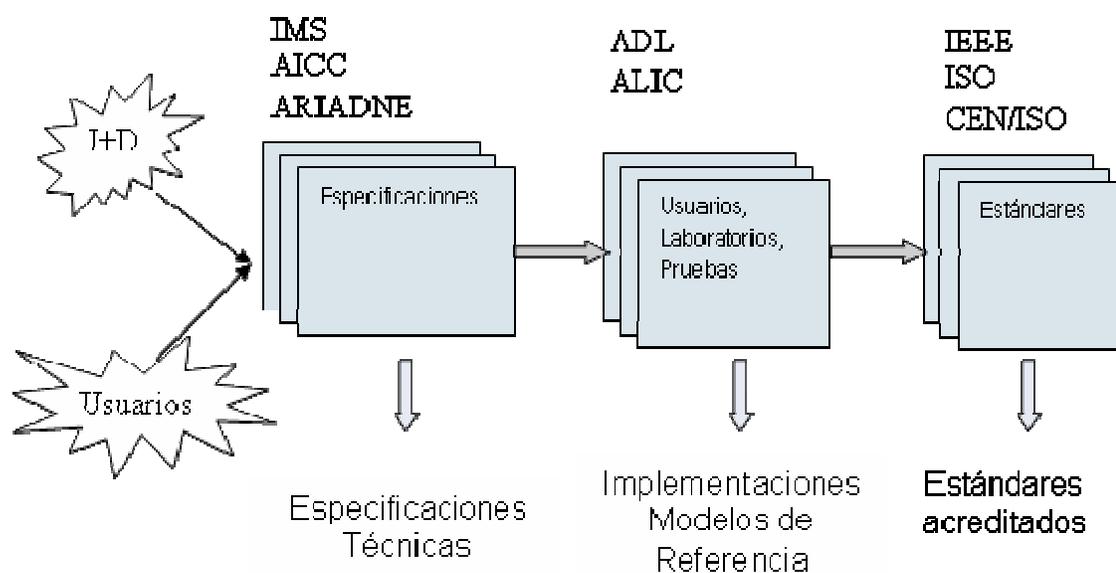


Figura 19. Evolución de los estándares y especificaciones de educación en línea.

Con el fin de aclarar los conceptos alrededor de las especificaciones y los estándares de la educación en línea, el Reporte de Industria (Industry Report)

publicado por el Learning Consortium (MASIE, 2003), ofrece una reseña de la evolución de los estándares y de las especificaciones de educación en línea, y su convergencia hacia un estándar de acogida general. Este proceso se puede resumir en las siguientes etapas:

- El primer paso consiste en definir áreas de investigación y el desarrollo a partir de unas necesidades identificadas por los usuarios. Dentro de estas áreas de estudio, diversas organizaciones desarrollan las especificaciones que identifican los puntos clave para obtener una solución a los problemas planteados. En el caso particular de la educación en línea, las especificaciones de mayor contribución son IMS, AICC y ARIADNE.
- A partir de estas especificaciones, y por medio de implementaciones y pruebas, se generan nuevos modelos de referencia (como SCORM), que permiten identificar aquellos aspectos en los que es necesario obtener nuevas especificaciones y realizar pruebas.
- Posteriormente, las especificaciones que han probado brindar soluciones efectivas a los problemas planteados se remiten a los organismos de estandarización, en los cuales se elimina cualquier dependencia hacia una industria o tecnología en particular, y se realiza un borrador de la propuesta. Si este borrador es aprobado, las especificaciones reciben el aval como estándares acreditados y se hacen públicos, para su adopción por parte de la comunidad en general.

Igualmente es necesario tener en cuenta que las posiciones de cada una de las organizaciones y los grupos que se muestran en La Figura 19 no se encuentran en conflicto, como se asume en algunas ocasiones. En vez de una competencia, existe una disposición a unir fuerzas y de responsabilidades en torno a la creación de un estándar único para la educación en línea. El aporte de SCORM a este objetivo a largo plazo consiste en *proporcionar un marco de aplicación y una referencia de implementación detallada, que permite que los contenidos y los sistemas de gestión*

de aprendizaje que lo adopten puedan asegurar un nivel adecuado de interoperabilidad. Por su parte, el modelo de referencia SCORM continuará con su evolución, incorporando los nuevos desarrollos de cada uno de los estándares y las especificaciones que lo conforman³⁹.

4.2. ESTRATEGIA PARA LA CREACIÓN DE CONTENIDOS BASADOS EN OBJETOS DE APRENDIZAJE

La estrategia para la creación de contenidos que se propone consiste en *desarrollar materiales que cumplen con el modelo de referencia SCORM*, para garantizar que puedan ser utilizados en diversas experiencias educativas y en diferentes ambientes virtuales de aprendizaje. Dado el carácter descriptivo y tecnológico de este trabajo, no se pretende profundizar en los aspectos pedagógicos que implica el desarrollo de estos materiales. No obstante, se espera que la implementación y las posteriores modificaciones a esta estrategia permitan crear un espacio para la reflexión pedagógica, al interior de la institución o las instituciones que hagan uso de ella.

Tomando como base las ideas planteadas por Pasini y Rehak (2003), el éxito de cualquier estrategia de creación de contenidos depende en gran medida que se cumplan las siguientes condiciones:

- Debe existir una planeación de la experiencia educativa que se desea realizar, en la cual se definen sus objetivos y las actividades que el estudiante debe llevar a cabo para cumplir con los objetivos planteados para dicha experiencia.
- Los nuevos contenidos que se van a desarrollar tienen la posibilidad de ser reutilizados dentro de experiencias similares.

³⁹ En esta dirección es importante mencionar la iniciativa Learning Design del IMS Learning Consortium, que busca incluir aspectos pedagógicos y actividades de trabajo colaborativo dentro de los paquetes de contenidos. Se espera que esta y otras iniciativas se incluirán en la nueva versión del modelo de referencia SCORM.

- La experiencia va a ser mediada por un ambiente virtual de aprendizaje que proporciona algún nivel de conformidad con los estándares y las especificaciones para la creación y el manejo de contenidos reutilizables.

A continuación se presentan las fases generales del desarrollo de la estrategia de creación de los contenidos basados en SCORM, obtenidas a partir del análisis del trabajo del Learning Systems Architecture Lab (LSAL 2003), y la guía para el desarrollo de objetos de aprendizaje publicada por Cisco Systems (Cisco, 2003).

4.2.1. Fase 0. Conformación del equipo de trabajo

Esta fase preliminar tiene por objetivo conformar un equipo de trabajo interdisciplinario, en el cual se deben integrar la pedagogía, el conocimiento y la experiencia en el área de estudio en la cual se van a desarrollar los materiales, así como los aspectos tecnológicos que involucra la creación de materiales educativos que cumplan con el modelo de referencia SCORM⁴⁰. La Tabla 7, que fue adaptada a partir de la propuesta del LSAL (2003), presenta los diferentes roles que deben conformar el equipo de trabajo

Tabla 7. Roles que conforman del equipo de trabajo

Rol	Responsabilidades
Diseñador de la Experiencia	Desarrollar la estructura básica de los contenidos, y establecer los requerimientos que deben cumplir de acuerdo con la experiencia a desarrollar.
Experto en el área	Trabajar en conjunto con el diseñador de la experiencia y con los autores y desarrolladores para verificar que los materiales son adecuados desde el punto de vista técnico y/o pedagógico
Autor / Desarrollador de contenidos	A partir del diseño de la estructura de contenidos, crear, adaptar o reutilizar los materiales necesarios para desarrollar la experiencia.

⁴⁰ Se utiliza el término SCORM Compliant (cumple con SCORM), en aquellos materiales que siguen este modelo de referencia pero no están certificados por una entidad como ADL. Los materiales SCORM Conformant (conformes con SCORM) cumplen con SCORM y además han aprobado el proceso de de certificación de ADL

Rol	Responsabilidades
Programador / Desarrollador	Crear el paquete de contenidos de la(s) experiencia(s). Este proceso incluye la secuenciación de los contenidos, en coordinación con el diseñador de la experiencia y el experto en el área.
Productor de gráficos y multimedia	Crear gráficos, video, animaciones e interacciones, de acuerdo con los lineamientos del diseñador de la experiencia y el desarrollador.
Catalogador	Desarrollar e integrar los metadatos para los SCO, assets y agregaciones de contenidos

4.2.2. Fase 1. Análisis del contexto de la experiencia de aprendizaje

De acuerdo con el planteamiento de Wiley (2003), la “utilización” de un objeto de aprendizaje puede ser vista como una “contextualización” de éste dentro de una experiencia educativa. En este sentido es necesario analizar el contexto de la experiencia, con el fin de determinar los escenarios en los cuales es posible integrar o reutilizar los objetos de aprendizaje⁴¹. Este análisis puede tomar como insumos la revisión del desempeño general de los estudiantes en experiencias anteriores, así como la experiencia obtenida por los profesores en el desarrollo de otras experiencias.

Igualmente es importante contar con una definición clara de los objetivos que se desean alcanzar a través de la experiencia educativa, ya que éstos proporcionan la base necesaria para la selección o el diseño de los contenidos, las estrategias pedagógicas y la metodología a utilizar. Unos objetivos bien definidos permiten que el estudiante haga uso de sus habilidades meta-cognitivas para integrar la experiencia de aprendizaje con los modelos mentales que ya posee, lo cual contribuye a que sea él mismo quien organice sus esfuerzos para cumplir con los objetivos propuestos. Sin esta base es posible que los materiales, las actividades y

⁴¹ El contexto puede estar formado por el área de conocimiento, la posible audiencia de la experiencia de aprendizaje, las herramientas tecnológicas y las estrategias que se van a utilizar.

las evaluaciones no aporten de forma significativa a la experiencia, y se conviertan en un obstáculo para el desarrollo del proceso de enseñanza y aprendizaje.

4.2.3. Fase 2. Diseño de la estructura de contenidos y caracterización de los objetos de aprendizaje

El diseño de la estructura de contenidos toma como base el análisis del contexto, los objetivos planteados, y la estrategia de enseñanza y aprendizaje que se desea implementar. Antes de comenzar el diseño de la estructura, es necesario identificar los tipos de materiales que se pueden incluir dentro de las experiencias de aprendizaje. Estos tipos se describen a continuación.

- **Receptivo:** En este tipo de material el estudiante tiene el rol de observador que se ve “forzado” a seguir la secuencia definida por el creador. Algunos ejemplos de este tipo de material son las lecturas y las presentaciones con diapositivas.
- **Directivo:** Este material ofrece la posibilidad de que el estudiante realice algún tipo de interacción o ejercicio, antes o después de revisar las secciones que lo constituyen. Generalmente se establece un “camino” a través del material, en el cual la secuencia lleva de un tema sencillo a uno de mayor complejidad, y en cada tema es posible realizar ejercicios de repaso o refuerzo.
- **Descubrimiento guiado:** Esta clase de material se basa en el concepto de resolución de problemas. En este caso se presenta una situación o problema similar a aquellos que el estudiante encontrará en su área de desempeño, y que deberá resolver utilizando las herramientas disponibles y la información proporcionada por el material. Algunos ejemplos de este tipo de materiales son los estudios de casos y la solución de problemas basada en escenarios o estrategias.
- **Exploratorio:** En este tipo de objetos de aprendizaje el estudiante puede “saltar” de forma arbitraria dentro del contenido, con el fin de encontrar la información que requiere para construir un nuevo conocimiento o solucionar un problema

específico. Se debe tener en cuenta que el éxito de este tipo de materiales depende en gran medida de la motivación del estudiante.

4.2.3.1. Definición de la estructura de contenidos

Una vez identificados los tipos de materiales que se van a utilizar de acuerdo con las experiencias de aprendizaje, se inicia el proceso de definir la estructura de contenidos. Para ello se puede tomar como base la clasificación de la jerarquía de contenidos propuesta por Cisco (2003), que se muestra en la Figura 20.

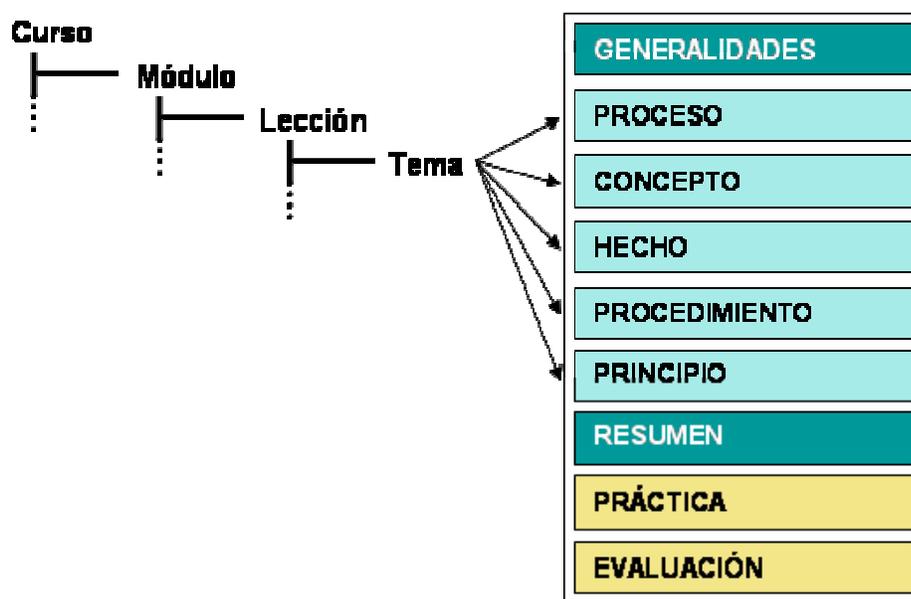


Figura 20. Jerarquía de contenidos propuesta por Cisco

La jerarquía de contenidos caracteriza a los *temas* (Topics) como los objetos de aprendizaje básicos, que pueden ser integrados de acuerdo con la estrategia pedagógica, para obtener objetos de aprendizaje de mayor complejidad⁴². Estos temas se pueden clasificar en conceptos, hechos, procesos, procedimientos y principios. A continuación se presenta una breve reseña de cada uno de estos tipos básicos de objetos de aprendizaje.

⁴² En la práctica un tema puede tener subtemas, pero el esquema básico es el mismo.

Conceptos

Los conceptos responden a la pregunta ¿Qué es...? y consisten en una colección de objetos, símbolos, ideas o eventos que se definen por medio de una palabra o un término simple. Este tipo de temas se ubican al inicio de una lección, y ofrecen las definiciones que se utilizarán en su desarrollo. La Tabla 8 presenta la guía para presentar un concepto.

Tabla 8. Estructura de un concepto

Subtema	Descripción
Introducción	Busca establecer el objetivo del concepto y orientar al estudiante acerca del concepto que se va a presentar
Definición	Puede consistir en un texto, una gráfica o ilustración, y busca identificar las características del concepto
Ejemplo	Pueden consistir en texto o gráficos que contribuyen a la definición del concepto. Se pueden incluir varios ejemplos, en los cuales se puede variar el nivel de complejidad.

Hechos

Este tipo de temas se usan cuando se requiere mostrar una información específica acerca de un objeto (por ejemplo, cómo está compuesto), presentada en forma de texto, datos o imágenes. Generalmente los hechos se encuentran ligados a los conceptos, procedimientos o principios, y se deben ubicar en la estructura de contenidos de forma que fluyan lógicamente dentro de la lección. La Tabla 9 muestra una guía para la construcción de un hecho.

Tabla 9. Estructura de un hecho.

Subtema	Descripción
Introducción	Busca orientar al estudiante acerca del hecho que se va a presentar
Hecho	Puede ser presentado como una combinación de texto, gráficos o una lista de características del objeto que se desea describir.

Procesos

Este tipo de temas se utiliza cuando se desea presentar un flujo de eventos que describen el funcionamiento de un sistema, o una tarea que puede involucrar varias personas o entidades. Los procesos se pueden utilizar como entidades independientes, o como parte de un procedimiento, con el fin de establecer su contexto. La Tabla 10 muestra una guía para la construcción de un proceso.

Tabla 10. Estructura de un proceso.

Subtema	Descripción
Introducción	Busca orientar al estudiante acerca del proceso que se va a presentar
Lista de etapas del proceso	Esta lista puede incluir las acciones y los roles de cada una de las etapas del proceso.
Diagramas	El proceso también se puede representar por medio de diagramas en los cuales se ilustran sus etapas y los roles que participan en su desarrollo. Es posible combinar los diagramas con la lista de etapas.

Procedimientos

Este tipo de temas se utiliza cuando se requiere mostrar el desarrollo de una tarea o la toma de una decisión, y consisten en una serie de indicaciones de las actividades que se deben seguir cada vez que se desarrolle. Generalmente los procedimientos se presentan después de un concepto o un proceso. La Tabla 11 muestra una guía para la construcción de un procedimiento.

Tabla 11. Estructura de un procedimiento.

Subtema	Descripción
Introducción	Busca orientar al estudiante acerca del procedimiento que se va a presentar
Lista de pasos o acciones	Se debe describir cada paso o acción que forma parte del procedimiento. Generalmente comienzan con un verbo. En el caso que el proceso requiera tomar decisiones, cada una de ellas se deberá escribir la condición y las acciones correspondientes.
Demostración	Si es posible, se debe presentar una demostración del procedimiento.

Principios

A diferencia de los procedimientos, este tipo de temas se utiliza cuando se requiere describir las guías que se deben seguir, teniendo en cuenta diversos contextos o escenarios. La tabla muestra la estructura de un principio.

Tabla 12. Estructura de un principio.

Subtema	Descripción
Introducción	Busca orientar al estudiante acerca del principio que se va a presentar
Definición del principio	Se debe presentar una guía con los lineamientos para el escenario ideal
Lineamientos	Se deben presentar diferentes guías de acuerdo con el análisis del desempeño del escenario ideal, y teniendo en cuenta diversos escenarios y situaciones.
Ejemplos	Se recomienda presentar varios ejemplos, que incluyan los diversos escenarios y situaciones, en los cuales se muestren aquellos casos en los cuales el principio no se puede aplicar, con su correspondiente explicación.

4.2.3.2. Caracterización de los objetos de aprendizaje

Luego de definir la los tipos de materiales que se van a utilizar en la experiencia y definir la estructura de contenidos, es posible identificar los objetos de aprendizaje que se deben crear o reutilizar dentro de la experiencia educativa. La caracterización de los objetos de aprendizaje debe estar fundamentada en el análisis del nivel cognitivo que aporta cada componente de la estructura de contenidos (curso, módulo, lección o tema), y a partir de este análisis se obtiene una lista de requerimientos que deben cumplir los objetos de aprendizaje.

El análisis del nivel cognitivo de cada objeto dentro de la estructura de contenidos permite definir los conocimientos que el estudiante debe estar en capacidad de construir, apropiar y aplicar luego de que la experiencia ha finalizado, y puede ser útil para definir los instrumentos de medición que se deben utilizar. Para realizar este análisis es posible hacer uso de las estrategias de clasificación de Merrill o Bloom, que se muestran en la Figura 21.

Merrill	Bloom
Recordar	Conocimiento
	Comprensión
Usar	Aplicación
	Análisis
	Síntesis
	Evaluación

Figura 21. Niveles cognitivos de acuerdo con Merrill y Bloom

De acuerdo con la clasificación de Merrill, el conocimiento puede ser evaluado en dos categorías: recordar y usar. Por su parte, la clasificación de Bloom presenta seis niveles cognitivos, que permiten definir de manera más específica el tipo de actividad de aprendizaje que se busca apoyar con el material. Por ejemplo la actividad “enumerar” requiere que el estudiante recuerde alguna información, y de acuerdo con Bloom se podría clasificar como “Conocimiento”. Por otro lado, la actividad “configurar” requiere que el estudiante realice una tarea específica que requiere de cierto conocimiento, por lo cual podría ser clasificada como “Aplicación”.

4.2.4. Fase 3. Reutilización y construcción de los objetos de aprendizaje

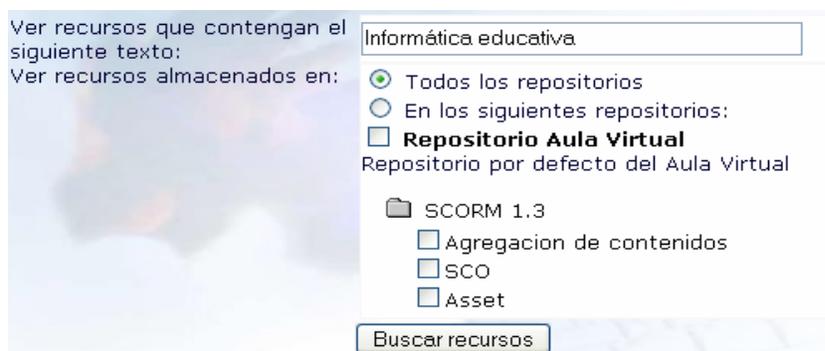
En esta fase se parte del diseño de la estructura de contenidos, los objetivos de aprendizaje y los requerimientos obtenidos en las fases anteriores, con el fin de establecer un plan para la reutilización y el desarrollo de los objetos de aprendizaje que se integrarán en los materiales educativos. Este plan puede hacer uso de diferentes alternativas, que se describen a continuación.

4.2.4.1. Reutilización de objetos de aprendizaje

La posibilidad de reutilizar los objetos de aprendizaje es uno de los aspectos más importantes de la estrategia planteada, ya que permite agilizar el proceso de creación de los materiales, al hacer uso de aquellos recursos que han sido desarrollados para contextos similares. Este proceso requiere tomar decisiones en

cuanto al nivel de granularidad de los objetos de aprendizaje que pretenden ser reutilizados dentro de la nueva experiencia (cursos, módulos, lecciones o temas). De acuerdo con South (2001), el grado óptimo de reutilización debe ser determinado partiendo del nivel del curso hacia el nivel de los temas individuales, y se deben buscar especialmente aquellos temas o lecciones que se enfocan en un objetivo específico, y que pueden ser llevados a otro contexto sin que pierdan el objetivo de aprendizaje para el cual fueron creados.

Una vez definido el nivel de granularidad de los materiales, el experto en el área y el diseñador de la experiencia deben realizar una búsqueda para localizar los objetos de aprendizaje que tienen posibilidad de ser reutilizados. Para llevar a cabo esta búsqueda se deben hacer uso de los datos más relevantes de cada objeto, como algunas palabras clave, el contexto o el posible título de los objetos de aprendizaje⁴³. Como resultado de esta búsqueda se obtiene una lista de los objetos de aprendizaje candidatos, que debe ser puesta a consideración de todo el equipo de trabajo para descartar aquellos objetos que no cumplan con los requerimientos establecidos. La Figura 22 muestra la interfaz de búsqueda de objetos de aprendizaje implementada como parte de Aula Virtual.



Ver recursos que contengan el siguiente texto:
Ver recursos almacenados en:

Todos los repositorios
 En los siguientes repositorios:
 Repositorio Aula Virtual
Repositorio por defecto del Aula Virtual

SCORM 1.3
 Agregacion de contenidos
 SCO
 Asset

Figura 22. Interfaz de búsqueda de objetos de aprendizaje

⁴³ A medida que se adquiere experiencia en la búsqueda de los objetos de aprendizaje, se puede hacer uso de cualquier campo de los metadatos que se almacenan para cada objeto, como la densidad semántica, el nivel de interactividad, etc.

4.2.4.2. Construcción de nuevos objetos de aprendizaje

Al igual que la reutilización de objetos de aprendizaje, la construcción de nuevos materiales parte de los requerimientos establecidos por el grupo de trabajo en las fases anteriores. Dentro de este proceso los roles que deben asumir el liderazgo son el autor o desarrollador de los contenidos, el productor de gráficos y multimedia, y el programador. Estos roles deben estar en contacto permanente con el diseñador de la experiencia y el experto en el área, quienes deberán seguir de cerca el desarrollo de los nuevos objetos de aprendizaje. A continuación se describen las etapas generales de este proceso.

Diseño del material hipermedia

El objetivo de esta etapa es construir la estructura lógica del objeto de aprendizaje, y organizar los recursos aportados por el experto en el área y el diseñador de la experiencia. Para ello se deberá realizar el diseño de la estructura y funcionamiento de la interfaz del objeto de aprendizaje de acuerdo con la estrategia de aprendizaje planteada, la redacción y recopilación del contenido textual y la creación de bocetos para las ilustraciones y animaciones que se desean incluir dentro del material.

Diseño gráfico

En esta etapa se realizará la labor de creación y recopilación de las ilustraciones y el material multimedia, teniendo en cuenta las condiciones de estética y usabilidad necesarias para ofrecer una experiencia de aprendizaje agradable. Esta labor incluye la selección y definición de estilos que determinarán el tipo, el color y el tamaño de las fuentes, las combinaciones de colores y el tratamiento gráfico de los elementos multimedia.

Ensamblado del material

Una vez que se cuenta con todos los elementos tecnológicos que conforman el material del objeto de aprendizaje, se deberá realizar la labor de ensamblado de los recursos, la integración de los gráficos y las animaciones generadas y el ajuste al

formato con los estilos seleccionados. Esta labor podrá ser desarrollada utilizando diversas herramientas tecnológicas, como editores de páginas web, procesadores de texto, gestores de presentaciones o herramientas de producción de animaciones. El producto final de esta etapa deberá ser un material que puede ser visualizado por medio de un navegador Web o de una herramienta disponible en diferentes plataformas y sistemas operativos⁴⁴.

Programación

En algunas ocasiones será necesario llevar a cabo tareas de programación, con el fin de cumplir con los requerimientos establecidos para los nuevos objetos de aprendizaje. La programación puede incluir el desarrollo de simuladores o herramientas de cálculo, la creación de rutinas de validación, seguimiento o evaluación, así como el desarrollo de los mecanismos de comunicación entre el objeto y el sistema de gestión de aprendizaje. Estas tareas estarán a cargo del desarrollador, en coordinación con los demás miembros del equipo de trabajo.

4.2.5. Fase 4. Catalogación y almacenamiento de los objetos de aprendizaje

A pesar que esta fase puede ser desarrollada en forma independiente, es recomendable que la captura de los metadatos para los objetos de aprendizaje se realice en forma paralela a su diseño y desarrollo. Para ello es posible que el diseñador de la experiencia y el experto en el área definan los primeros valores de los metadatos, como el título, la descripción, algunas palabras clave, el nivel de dificultad y el tiempo promedio que se invierte en el desarrollo de la experiencia. De igual forma el desarrollador puede especificar otros valores, como el formato, el tamaño, la fecha de creación, y la información del autor. Asimismo, es posible que la herramienta de desarrollo o almacenamiento de los objetos de aprendizaje pueda proporcionar el identificador único del objeto y su localización, entre otros datos.

⁴⁴ Como visores de documentos postscript, Applets de Java o animaciones en Flash.

La captura de los metadatos para un objeto de aprendizaje puede realizarse por utilizando las categorías resaltadas de la tabla que se presenta en el Anexo 1, propuesta por el LSAL (2003). Estas categorías agrupan los elementos del esquema de metadatos definidos por el estándar IEEE LOM que se utilizan con mayor frecuencia para describir un objeto de aprendizaje⁴⁵. Igualmente es posible utilizar editores especializados para recolectar los metadatos, como el editor RELOAD⁴⁶ o las herramientas desarrolladas dentro de los repositorios, los sistemas de gestión de aprendizaje (LMS) o los sistemas de gestión de contenidos (CMS). Otra estrategia consiste en utilizar un formato para la recolección de los metadatos, como el propuesto en la Tabla 13.

Tabla 13. Formato para la recolección de los metadatos LOM de un objeto de aprendizaje

METADATOS			
Título:		Versión:	
Descripción:		Estado:	
Palabras clave:		Fecha:	
Catálogo:		Estructura:	
Entrada en el catálogo:		Formato:	
Descripción de la clasificación:		Tamaño (bytes):	
Propósito:	Palabra clave de la clasificación:		
Contribución:	Rol:	Esquema de metadatos:	
Entidad:	Catálogo de metadatos:		
	Entrada en el catálogo:		
Tipo de recurso:	Tipo de interactividad:		

⁴⁵ La lista completa de los campos que conforman la especificación de metadatos LOM puede ser consultada en el SCORM Content Aggregation Book, 2004.

⁴⁶ Disponible en línea en <http://reload.ac.co.uk>

		Nivel de interactividad:
Costo:	Derechos de autor y otras restricciones:	
Descripción de los derechos:		Tiempo promedio de aprendizaje:
Ubicación:		

Una vez que los objetos de aprendizaje han sido debidamente catalogados, se deben almacenar en un repositorio o en un sistema de gestión de contenidos, que permita su posterior búsqueda y acceso. Para ello se deberán empaquetar los objetos de aprendizaje en forma de archivos comprimidos, y se deberán incluir dentro de la colección de objetos del repositorio, el sistema de gestión de contenidos o la estructura de archivos y directorios destinada para tal fin. Igualmente se deberán incluir sus metadatos, para permitir su posterior búsqueda y reutilización en otras experiencias de aprendizaje.

Se debe tener en cuenta que esta fase puede ser omitida para aquellos materiales que sólo se van a utilizar dentro de una experiencia educativa. Para ellos no es necesario recolectar los metadatos, ni almacenarlos en los repositorios de contenidos, ya que formarán parte de un solo paquete de materiales.

4.2.6. Fase 5. Agregación de los objetos de aprendizaje

En esta fase se toman los objetos de aprendizaje que se han reutilizado o desarrollado, y se integran en la estructura de contenidos que ha sido definida en las fases anteriores. Para realizar esta tarea es necesario utilizar una herramienta que cumpla con la especificación de empaquetamiento de contenidos seleccionada, y que ofrezca las facilidades para incluir los metadatos y los archivos de los cuales dependen los materiales. Igualmente, en el caso de utilizar el modelo de referencia SCORM, se debe consultar el libro SCORM Runtime Environment - RTE (2003), en el cual se encuentran los lineamientos que se deben considerar para crear materiales cumplan con este modelo de referencia.

La Figura 23 muestra el editor RELOAD en el proceso de creación de la estructura de contenidos para un curso de Photoshop, proporcionado por Advanced Distributed

Learning (ADL). Este editor permite definir la estructura de contenidos ajustada a SCORM, asociar los recursos a las actividades planteadas y definir los mecanismos de secuenciación de estas actividades.

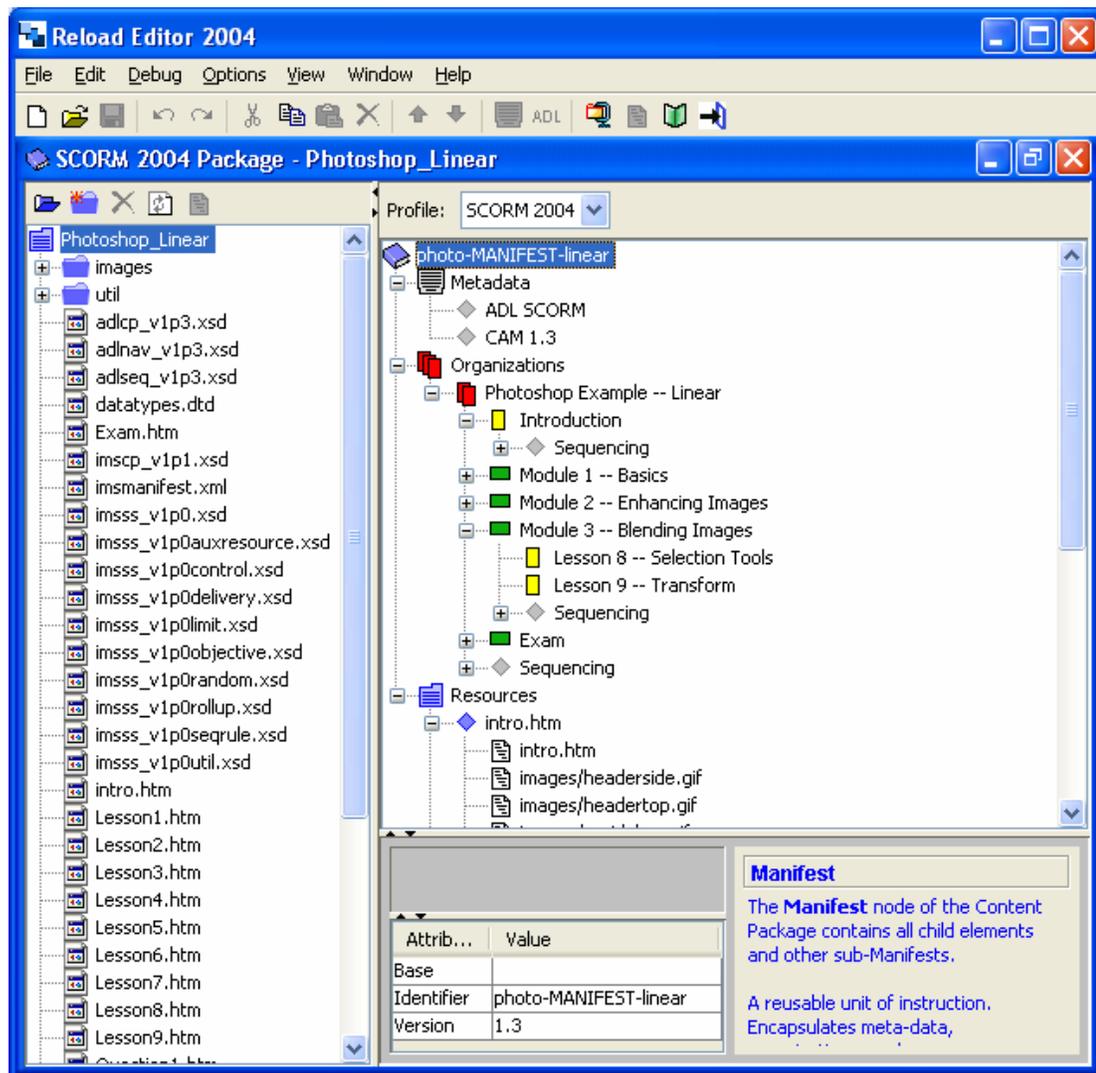


Figura 23. RELOAD: Herramienta para la creación de estructuras de contenidos

Otra estrategia para crear la estructura de contenidos consiste en utilizar sistemas de gestión de aprendizaje o sistemas de gestión de contenidos que posean la capacidad de crear y almacenar materiales que cumplen con SCORM. Con el fin de poner en práctica esta estrategia se implementó un módulo dentro de Aula Virtual,

que permite crear estructuras de contenidos que cumplen con SCORM, y a las cuales se les aplica un mecanismo de secuenciación predeterminado. La Figura 24 muestra una vista del módulo de creación de materiales compatibles con SCORM, en el cual se aprecian una estructura de contenidos, similar a la estructura creada utilizando el editor RELOAD.

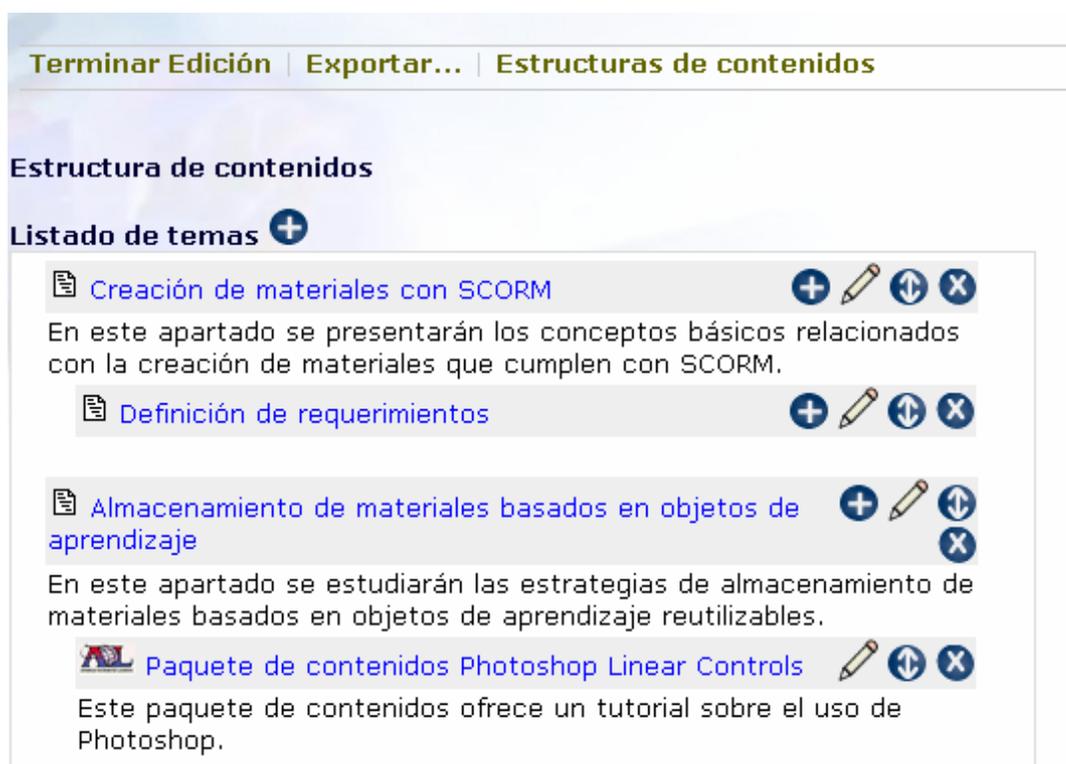


Figura 24. Módulo de Aula Virtual para crear materiales que cumplen con SCORM

4.2.7. Fase 6. Catalogación, almacenamiento y distribución de los paquetes de contenido

Una vez que los paquetes de contenido han sido estructurados, es necesario realizar su descripción por medio de metadatos, así como su empaquetado y almacenamiento en un repositorio o sistema de gestión de contenidos. El paquete de contenidos es un archivo comprimido que deberá contener toda la información

necesaria para desarrollar la experiencia de aprendizaje (curso, módulo, lección o tema).

La catalogación y el almacenamiento de los paquetes de contenidos deberán hacer uso de un repositorio digital o una herramienta similar, que ofrezca los servicios requeridos para organizar los paquetes de forma que puedan ser localizados y reutilizados en experiencias posteriores. Una vez que el paquete de contenidos haya sido incluido dentro del repositorio, se podrá utilizar en varias experiencias educativas, dentro de diversos sistemas de gestión de aprendizaje.

La Figura 25 muestra la interfaz de un repositorio implementado en el desarrollo de este proyecto, que permite adicionar nuevos objetos de contenido (Assets, SCO y agregaciones de contenido) que cumplen con SCORM.

Agregar recurso

Modelo de referencia SCORM 1.3

Tipo de recurso SCO

Nombre

Descripcion

Lenguaje Spanish (es)

Archivo .ZIP del recurso: Examinar...

El recurso es un archivo .ZIP, no descomprimir.
En este archivo comprimido se deben incluir todos los archivos locales a este recurso.

Archivo inicial:

Algunos recursos poseen varios archivos, se debe especificar el archivo principal.

Aceptar Cancelar

Figura 25. Interfaz que permite agregar un recurso a un repositorio

El repositorio también ofrece los servicios necesarios para realizar la edición en línea de los metadatos compatibles con IEEE LOM, importar los metadatos desde

un archivo XML y exportarlos a un archivo del mismo formato. Estos servicios permiten que la información de los objetos de aprendizaje y paquetes de contenidos se mantenga actualizada. La Figura 26 muestra la interfaz de edición de metadatos implementada dentro del repositorio de objetos de contenido.

Estructura de introducción a los computadores

Metadatos	
General	
Identificador	
Catalogo	URN
Entrada	A1927D0B-FCA9-FF09-DCA4-EE441AF2AF4A
Titulo	
Texto	{Lenguaje = es} Introducción a los computadores
Lenguaje	es

Edición de metadatos

Borrar elemento

Lenguaje	es
Lenguaje del texto	
Texto	Introducción a los computadores
El elemento titulo representa el nombre dado al objeto de aprendizaje	

Figura 26. Interfaz ofrecida por el repositorio para la edición de metadatos

4.2.8. Fase 7. Evaluación del proceso

En esta última fase se debe desarrollar una evaluación del proceso de creación de materiales, que permita establecer una serie de criterios y procedimientos que deberán ser considerados en desarrollos futuros. A partir de los resultados de esta evaluación es posible crear o perfeccionar una serie de guías de trabajo, en las cuales se especifican aspectos como el diseño de los objetos de aprendizaje, los mecanismos de presentación y secuenciación, las estrategias de empaquetado,

nombramiento de objetos, catalogación y distribución, y el papel de cada uno de los roles dentro del proceso.

5. DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA PARA EL ALMACENAMIENTO, LA ADMINISTRACIÓN, LA BÚSQUEDA Y LA DISTRIBUCIÓN DE OBJETOS DE APRENDIZAJE A TRAVÉS DE INTERNET

"If we cannot sustain the repositories we create we will retard our progress towards providing access to digital materials. If collections on which individuals have worked hard are forgotten or neglected, as some already seem to be, we will end up recreating, rediscovering and re-describing materials many times over"

Colin Holden, 2003⁴⁷

En la actualidad existe una gran variedad de perspectivas en cuanto a la definición del concepto de plataforma. En el contexto de esta investigación, se toma el concepto de Alvarado (2004), según el cual una plataforma puede ser definida como "lo que sirve para lograr algún fin". Desde este punto de vista, una plataforma es *un programa o una agrupación de programas, que integrados por medio de un protocolo de comunicación, tienen como objetivo proporcionar los medios tecnológicos para facilitar el intercambio de objetos de aprendizaje entre diversos sistemas, tales como aplicaciones de usuario, ambientes virtuales de aprendizaje y repositorios digitales.*

A continuación se presentan una serie de especificaciones conceptuales que describen la plataforma planteada, con el fin de ofrecer una visión general de su diseño y desarrollo⁴⁸. Algunas de estas especificaciones hacen uso del Lenguaje Unificado de Modelado (UML), y en ellas se realizarán algunas referencias que permitirán identificar las fases de desarrollo de la plataforma, de acuerdo con el Proceso Unificado de Desarrollo de Software (Kruchten, 2000).

⁴⁷ Tomado de CAMPBELL, Lorna, Blinco, Henry y Mason Jon. Repository Management and Implementation. White Paper para ati-lab 2004, preparado para Dest (Australia) y JISC-CETIS (UK)

⁴⁸ Este modelo de especificaciones fue adaptado a partir del documento eduSource Suite of Tools Use Cases Specifications. Versión 0.6. 2003.

5.1. ESPECIFICACIÓN DE CASOS DE USO PARA LA PLATAFORMA DE INTERCAMBIO DE OBJETOS DE APRENDIZAJE

Esta especificación proporciona una descripción de los requerimientos y restricciones de la plataforma planteada, y una vista de casos de uso generales para los servicios de creación, almacenamiento, administración y búsqueda de objetos de aprendizaje, que recopila las diferentes versiones de los diagramas de casos de uso que se desarrollaron para describir el comportamiento del sistema. Finalmente se presentará una vista de los paquetes conceptuales más importantes en la plataforma.

5.1.1. Requerimientos y restricciones de la plataforma

Los requerimientos y las restricciones de la plataforma fueron planteados a partir del estudio de las iniciativas presentadas en el marco teórico, específicamente en el modelo de referencia SCORM y la especificación para la interoperabilidad de repositorios digitales IMS DRI. A continuación se presentan el objetivo general y las restricciones que definen el contexto de la plataforma, y que permitieron establecer las bases para el desarrollo de una arquitectura distribuida de repositorios y sistemas de gestión de aprendizaje.

- *Objetivo: Proporcionar una interfaz abierta y distribuida para el almacenamiento, la administración y la búsqueda de materiales educativos reutilizables:* Con el fin de lograr este objetivo se debe diseñar una arquitectura basada en servicios web que permita ofrecer los servicios de almacenamiento, administración y búsqueda de materiales educativos reutilizables.
- *Restricción: Utilizar estándares y protocolos de acceso libre, que ofrezcan un alto grado de interoperabilidad entre diversas implementaciones:* La plataforma debe ser implementada sobre una infraestructura de acceso libre, utilizando estándares (como XML y protocolos como HTTP y SOAP), que permita que los

materiales puedan ser accedidos desde sistemas implementados sobre diferentes arquitecturas y sistemas operativos.

- *Restricción: Permitir el uso de diversas especificaciones de metadatos:* La plataforma deberá estar diseñada de tal forma que soporte diversas especificaciones de metadatos para la descripción de los materiales, de forma transparente a los sistemas que hacen uso de sus servicios.

5.1.2. Modelos de casos de uso

A continuación se presentan los diagramas de casos de uso que ilustran los servicios ofrecidos por la funcionalidad central de la plataforma y que fueron implementados en el desarrollo de este trabajo de investigación.

5.1.2.1. Modelo de actores de la plataforma

La Figura 27 muestra los actores principales de la plataforma, que se describen a continuación.

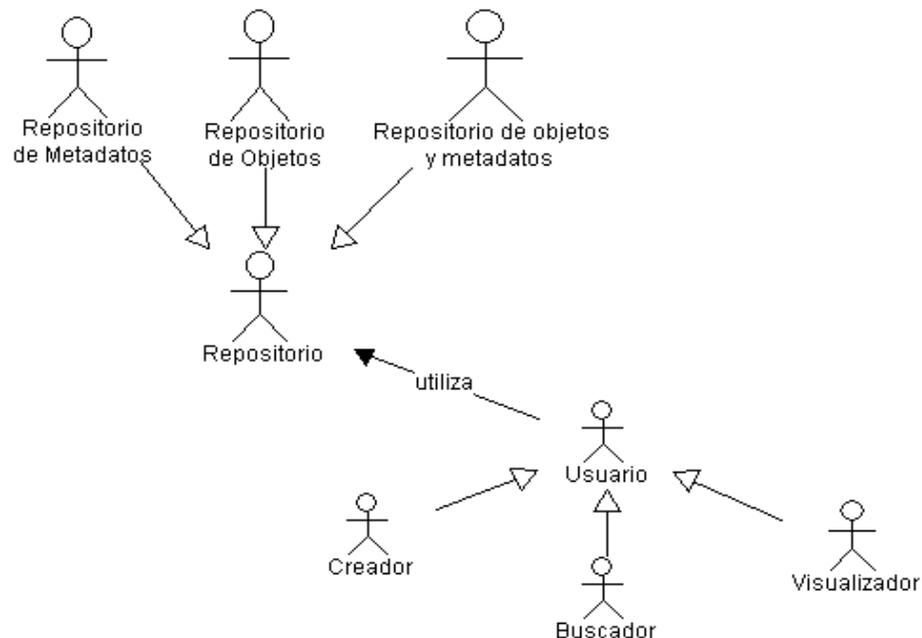


Figura 27. Modelo general de actores la plataforma

Repositorios de contenidos y/o metadatos

Son los encargados de ofrecer los servicios de almacenamiento, búsqueda, administración y distribución de objetos de aprendizaje y sus metadatos. Generalmente los repositorios que almacenan los objetos de aprendizaje también almacenan sus metadatos, pero en un ambiente distribuido pueden existir repositorios que sólo almacenan objetos de aprendizaje y / o metadatos por separado.

Usuarios

El término “usuario” en el modelo debe ser tomado como *la herramienta software que interactúa con los repositorios por medio de solicitudes a la interfaz pública de servicios que estos ofrecen*. Desde este punto de vista general un repositorio puede a su vez ser usuario de otro repositorio, lo cual permite crear una “red” de repositorios que cooperan para aprovechar las fortalezas de cada uno, y de esta forma ofrecer una mayor cantidad de objetos de aprendizaje y metadatos a sus respectivos usuarios.

Usuario Creador

Estos usuarios son los encargados de crear y/o modificar los objetos de aprendizaje y sus metadatos, y remitirlos a un repositorio. En el modelo IMS DRI estos usuarios se denominan *Creator*.

Usuario Buscador

Estos usuarios son los encargados de realizar búsquedas sobre los registros de metadatos de los objetos de aprendizaje almacenados en un repositorio, con el fin de localizar aquellos recursos que le pueden ser útiles para el desarrollo de algunas experiencias educativas. Para IMS DRI estos usuarios son llamados *Infoseeker*.

Usuario Visualizador

Este tipo de usuarios solicita los objetos de aprendizaje de los repositorios, para su posterior visualización dentro de una experiencia educativa. De acuerdo con DRI, esta clase de usuarios se denomina *Learner*.

5.1.2.2. Diagrama general de casos de uso

La Figura 28 presenta el diagrama general de casos que se plantea para la plataforma de intercambio de objetos de aprendizaje. Dentro de este diagrama se incluyen los casos de uso más importantes desde el punto de vista del usuario: almacenar objetos de aprendizaje y sus metadatos, buscar objetos de aprendizaje y visualizar o descargar los objetos de aprendizaje.

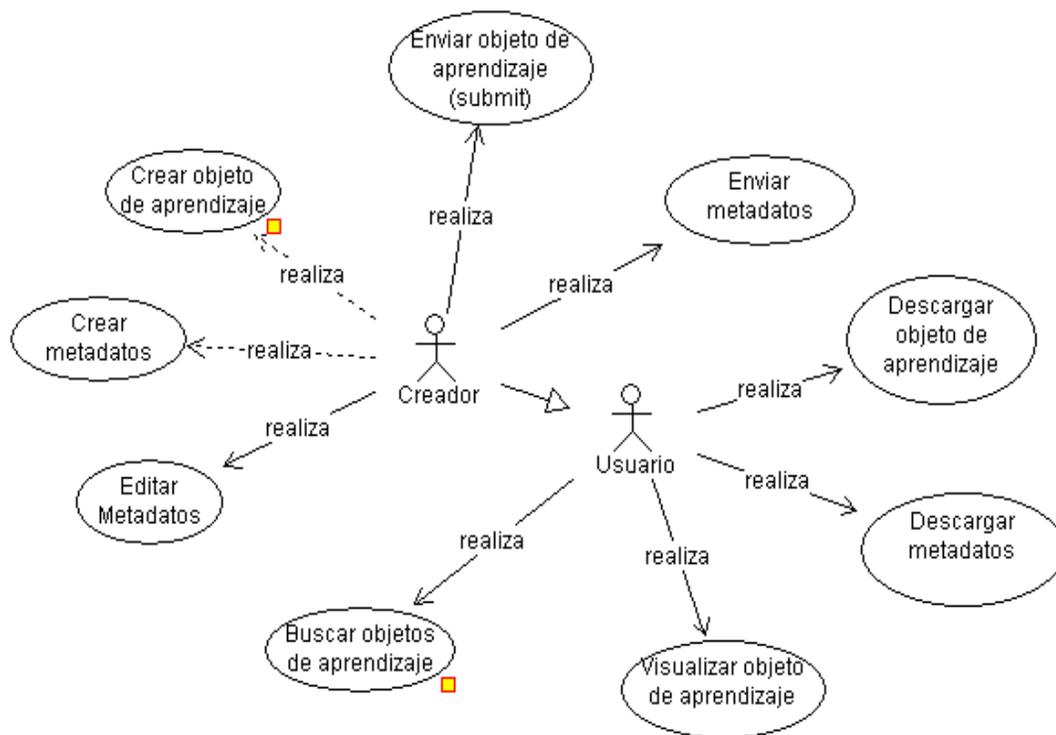


Figura 28. Diagrama general de casos de uso de la plataforma

Adicionalmente, dentro de este diagrama se incluyen los casos de uso crear objeto de aprendizaje, crear metadatos y editar metadatos, debido a que forman parte de la plataforma que se propone en este trabajo de investigación. La Figura 29 muestra el diagrama de casos de uso para la creación de objetos de aprendizaje.

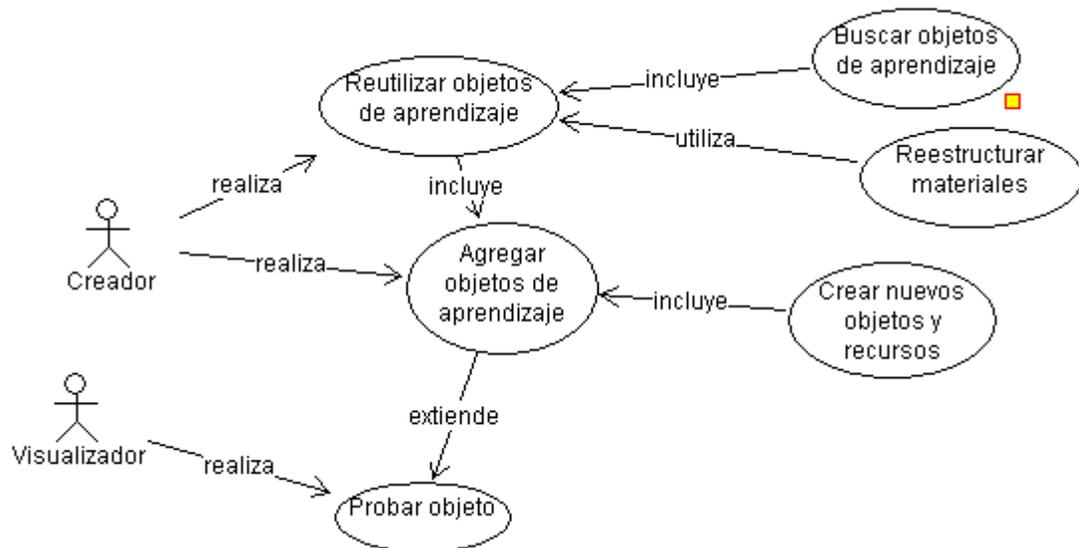


Figura 29. Diagrama de casos de uso crear de objetos de aprendizaje

Como se explicó anteriormente, el proceso de crear materiales educativos basados en objetos de aprendizaje puede ser realizado a través de la reutilización de objetos de aprendizaje existentes, la modificación de algunos materiales existentes para ajustarlos al modelo de referencia elegido, la creación de nuevos objetos de aprendizaje o la combinación de todas estas alternativas. El resultado de este proceso es un material educativo (agregación de contenidos) que también es un objeto de aprendizaje, y pasa a formar parte de la colección de objetos que pertenecen al usuario Creador. Se debe recordar que el Creador es una herramienta software que implementa las funcionalidades requeridas para crear los materiales (un LMS, LCMS, un repositorio o una aplicación de usuario).

Una de las funcionalidades más importantes del modelo propuesto es la búsqueda de objetos de aprendizaje, ya que a través de este servicio que se ofrece la posibilidad de localizar y reutilizar los recursos en experiencias posteriores. La Figura 30 muestra el diagrama de secuencia del caso de uso buscar objetos de aprendizaje (Search/Expose en el modelo IMS DRI).

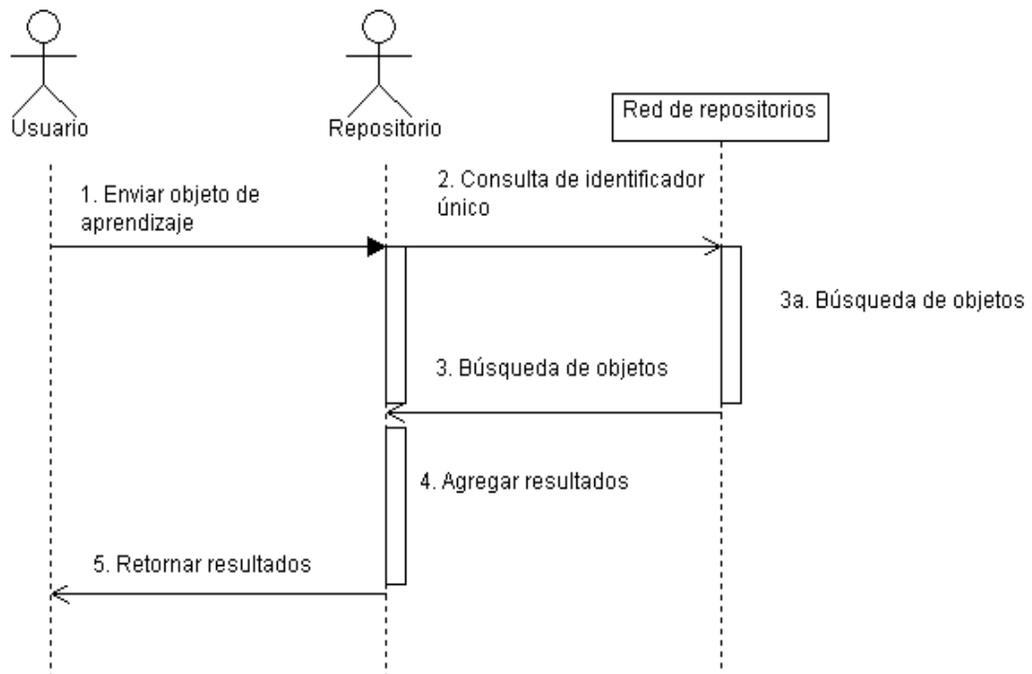


Figura 30. Diagrama de secuencia del caso de uso buscar objeto de aprendizaje

En primer lugar el usuario realiza una solicitud al repositorio, en la cual incluye los parámetros que permitirán configurar la búsqueda. Luego de recibir esta solicitud, el repositorio puede realizar un reenvío de la solicitud del usuario hacia los diferentes repositorios en los cuales se encuentre registrado, al tiempo que realiza la búsqueda en su catálogo de metadatos local. Luego el repositorio realiza una agregación de los resultados locales y los resultados de los demás repositorios, para enviar la información de la búsqueda de vuelta al usuario.

Igualmente, otro servicio de gran importancia que se plantea en la plataforma es la capacidad de almacenar objetos de aprendizaje y metadatos en un repositorio. La Figura 31 presenta el diagrama de secuencia para el caso de uso enviar objeto de aprendizaje y enviar metadatos.

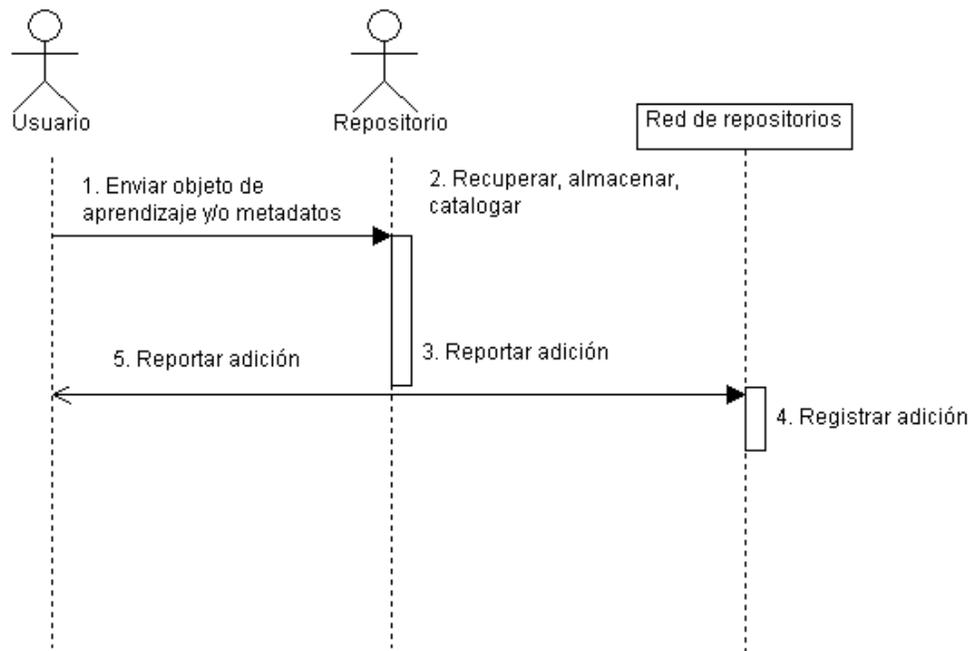


Figura 31. Diagrama de secuencia para la adición de objetos y/o metadatos

En esta figura se pueden apreciar los pasos generales que involucra el envío de un objeto de aprendizaje o sus metadatos a un repositorio (Submit/Store en el modelo de IMS DRI). En primer lugar el usuario (un LMS, LCMS, una aplicación de usuario o un repositorio) envía una solicitud al repositorio, en la cual incluye la dirección URL en donde se puede localizar el objeto de aprendizaje, o un String XML que contiene lo metadatos, según el caso. Si el usuario envía un objeto de aprendizaje, el repositorio deberá establecer una conexión HTTP para recuperar el objeto de aprendizaje. Posteriormente el repositorio almacena el objeto de aprendizaje en su estructura de directorios⁴⁹ o los metadatos, y reporta la nueva adición al usuario y a los demás repositorios que conforman la red. El paso 4 mostrado en la Figura 31 corresponde a la función *Alert* del modelo planteado por IMS DRI.

5.1.3. Modelo general de los paquetes del sistema

⁴⁹ Es posible almacenar los objetos en una base de datos, como objetos binarios (BLOB), pero esta implementación es indiferente al usuario que utiliza los recursos.

El modelo general del sistema que se presenta en la Figura 32 agrupa los diferentes casos de uso de la plataforma de intercambio de materiales. En ella se muestran los roles principales desde el punto de vista de los usuarios, y los servicios que deben ser ofrecidos por la plataforma.

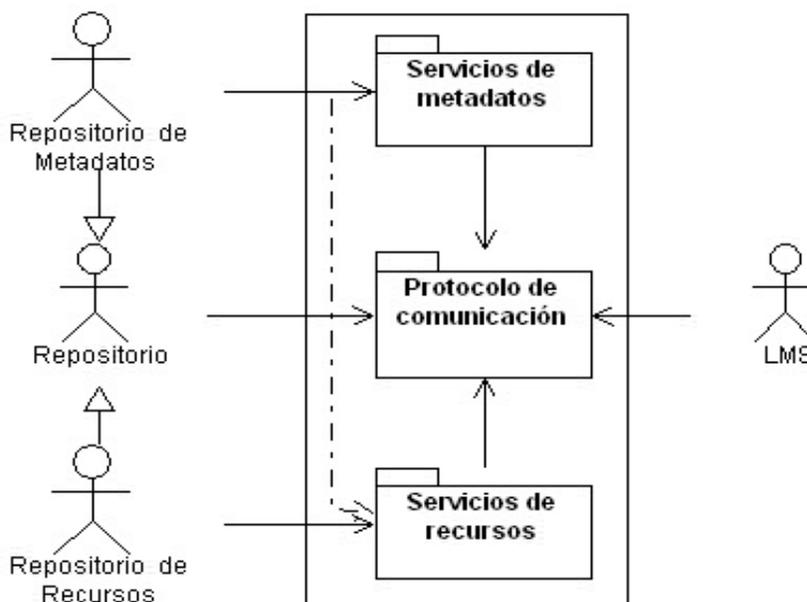


Figura 32. Modelo general de los paquetes de la plataforma

Los LMS y repositorios interactúan por medio de un protocolo de comunicación, que les permite intercambiar información acerca de los metadatos y los objetos de contenido (recursos). Los repositorios pueden ofrecer los servicios de metadatos y recursos, pero es posible que existan implementaciones que se enfoquen en el manejo de metadatos o recursos por separado. Por su parte los LMS hacen uso de los servicios de metadatos y recursos, con el fin de almacenar, estructurar y hacer uso de los objetos de aprendizaje en diversas experiencias educativas.

5.1.3.1. Servicios de recursos (objetos de aprendizaje)

Estos servicios ofrecen los medios requeridos para almacenar, administrar y distribuir diferentes tipos de objetos de aprendizaje, de acuerdo con las especificaciones de contenidos implementadas en el repositorio. En el caso del presente trabajo de investigación, se tomó como base el modelo de referencia

SCORM, que hace uso de la especificación para el empaquetado de contenidos de IMS (IMS Content Packaging Specification). Esta especificación define que los archivos que conforman un objeto de aprendizaje (SCO, Asset o Agregación de contenidos) se deben empaquetar en un archivo comprimido (ZIP), el cual puede ser almacenado, administrado o distribuido en diferentes sistemas y plataformas. A continuación se presentan los servicios de recursos que se definieron en la plataforma, así como su relación con el modelo propuesto por la Especificación de Interoperabilidad de Repositorios de IMS (o IMS DRI).

Almacenamiento de recursos

Este servicio permite adicionar un recurso dentro del repositorio de contenidos, por medio de una operación Submit/Store del modelo de IMS DRI. La implementación del almacenamiento de los recursos puede ser realizada de diversas formas, como utilizar estructuras de directorios, sistemas de archivos distribuidos o bases de datos con soporte para objetos binarios (Binary Large Objects – BLOB). En el presente trabajo de investigación se optó por utilizar un esquema de directorios, en el cual los objetos de aprendizaje se almacenan según la fecha en la cual fueron adicionados al sistema (Ver Figura 33).

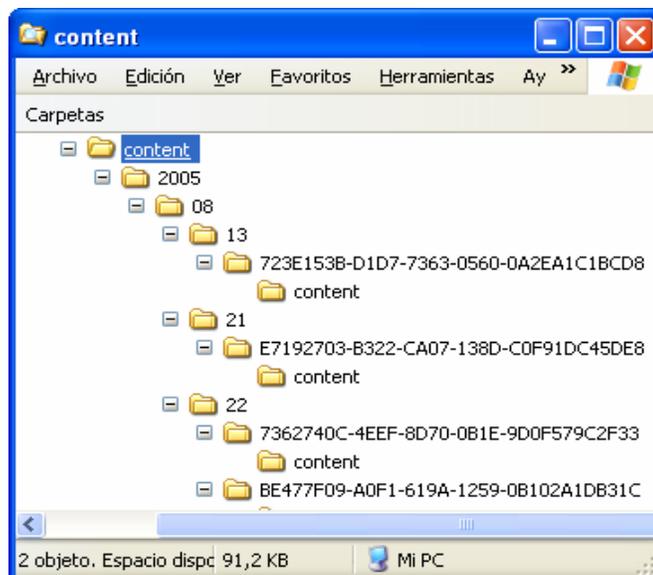


Figura 33. Estructura de almacenamiento de los recursos

Como se observa en la figura, el esquema de directorios implementado permite almacenar los recursos en una estructura de directorios año->mes->día, y para cada recurso se crea un directorio, cuyo nombre es el identificador único del objeto dentro del repositorio, obtenido por medio del algoritmo MD5.

Distribución de recursos

El objetivo de este servicio es facilitar la distribución de los objetos de aprendizaje, una vez que estos han sido solicitados al repositorio (Request/Deliver en el modelo de IMS DRI). Con el fin de ofrecer un alto grado de interoperabilidad y de realizar la transferencia de los objetos de forma transparente al sistema que los solicita, en el presente trabajo se utilizó el protocolo HTTP como medio para distribuir los recursos. Este esquema permite que los objetos de contenido sean accedidos desde diferentes herramientas (navegadores web, sistemas de gestión de aprendizaje, aplicaciones de usuario y otros repositorios), a pesar que no permite establecer la identidad del usuario final del recurso, o implementar políticas de seguridad diferentes a la autenticación HTTP. Una posible mejora al modelo propuesto sería hacer uso de servicios web bajo un ambiente de transmisión segura (utilizando SSL), o implementar mecanismos de identificación con llaves públicas y privadas.

5.1.3.2. Servicios de metadatos

Los servicios de metadatos ofrecen un mecanismo para permitir la catalogación, la búsqueda y el acceso a los objetos de aprendizaje almacenados en diversos repositorios. En principio la plataforma fue diseñada para que los metadatos y los objetos que estos describen se encuentren almacenados físicamente en el mismo repositorio, pero se considera que una extensión a los casos de uso básicos de los servicios de metadatos permitirá describir recursos almacenados en otros repositorios o cualquier tipo de recurso, como enlaces a páginas o servicios web, o

enlaces a aplicaciones específicas dentro de un contexto educativo⁵⁰. A continuación se presenta una reseña de los servicios de metadatos que fueron considerados en el presente trabajo, y su relación con los lineamientos planteados por la especificación IMS DRI.

Carga y descarga de metadatos

Estos servicios proporcionan la funcionalidad básica para incluir y descargar los metadatos de un objeto de aprendizaje, ajustados a un esquema como LOM o Dublin Core. Para el diseño de estos servicios se tomaron como base las funciones “Push”/Gather y “Pull”/Gather definidas en IMS DRI, para cargar y descargar los metadatos, respectivamente.

Búsqueda y localización

Este servicio permite buscar dentro de los registros de metadatos de los recursos almacenados en un repositorio de objetos de aprendizaje, con el fin de localizar aquellos materiales que pueden ser utilizados en las experiencias educativas que se planean desarrollar. Para su diseño se consideró el esquema Search/Expose de IMS DRI, y el esquema de búsqueda utilizado por Google Web APIs (2004). A partir de estos dos esquemas se construyó una solución intermedia, cuyo diseño es similar al esquema Search/Expose, pero debido a las restricciones que plantea este modelo en cuanto al formato de las búsquedas, su implementación obedece al esquema de búsqueda planteado por Google, en el cual se especifican pocos parámetros, y el formato de la búsqueda es independiente al esquema de metadatos almacenado en el repositorio.

Edición de metadatos

⁵⁰ Referencias a simuladores, software de utilidad y en general cualquier herramienta que pueda ser utilizada para apoyar una experiencia educativa.

La plataforma diseñada ofrece un servicio para realizar la edición en línea de los metadatos de cualquier recurso almacenado en un repositorio. Esta implementación presenta una interfaz web de edición, en la cual es posible agregar nuevos elementos, modificar o borrar los elementos del esquema de metadatos propio de un recurso específico. Inicialmente el único esquema de metadatos almacenado en el repositorio es IEEE LOM, pero la estructura del modelo de datos permite almacenar cualquier esquema de metadatos que se represente como XML.

5.2. PROTOCOLO DE COMUNICACIÓN PARA EL ALMACENAMIENTO Y LA GESTIÓN DE CONTENIDOS

El protocolo de comunicación permite establecer un lenguaje común que puede ser usado por cualquier sistema, que busca tener acceso a los servicios proporcionados por la plataforma de intercambio de materiales. Con el fin de obtener un alto nivel de interoperabilidad entre diferentes implementaciones de repositorios y sistemas de gestión de aprendizaje, este protocolo sigue los lineamientos planteados por la especificación IMS DRI, y su diseño está basado en el Protocolo para el Acceso Simple a Objetos (Simple Object Access Protocol – SOAP). A continuación se describen los componentes principales del protocolo propuesto.

5.2.1. Modelo de comunicación

El modelo de comunicación que se plantea con el protocolo de está fundamentado en el uso SOAP, en el cual todos los mensajes enviados y recibidos son representados como un documento XML (Figura 34). Esta característica le ofrece a SOAP una gran ventaja sobre otras implementaciones, tales como CORBA, DCOM COM, y JAVA RMI, que están ligadas a las plataformas o a los lenguajes de programación en las cuales se realizan los desarrollos. Adicionalmente, los mensajes SOAP son entregados por medio del protocolo HTTP, lo cual permite establecer la comunicación entre sistemas que se encuentran separados por redes de área local, redes corporativas e inclusive Internet.

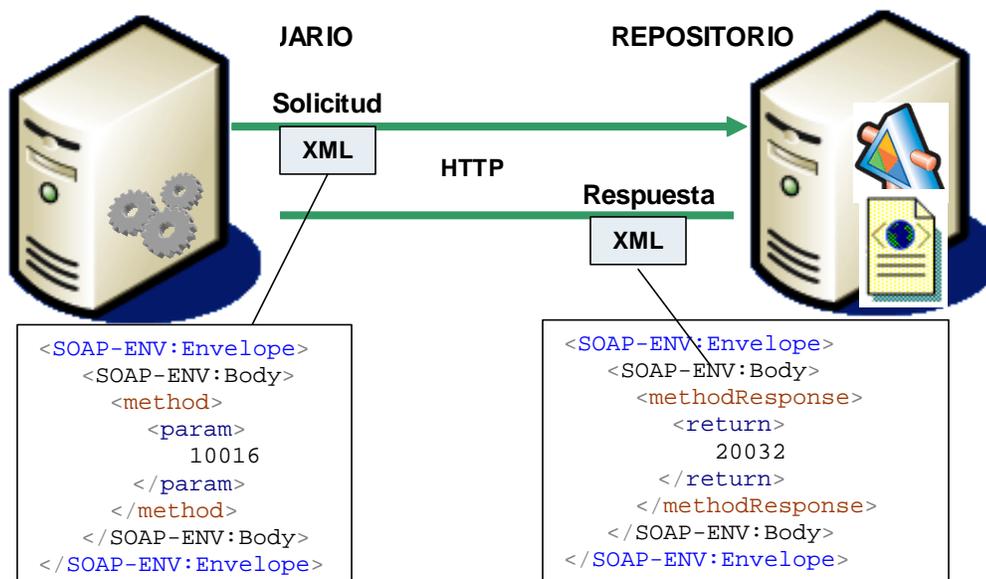


Figura 34. Modelo de comunicación entre los usuarios y los proveedores de servicios⁵¹

En la Figura 34 se muestra el funcionamiento básico del modelo de comunicación de la plataforma de intercambio de contenidos: el usuario envía un mensaje SOAP al servidor, dentro del cual se especifica el método que se desea invocar (*method*), y los parámetros de entrada (en este caso se envía sólo un parámetro entero, el número 10016). El proveedor de los servicios web (repositorio) recibe la solicitud, ejecuta el método invocado, y retorna un mensaje SOAP con el resultado de la ejecución (*methodResponse*). Dentro de este mensaje de respuesta se puede recibir un parámetro de salida (*return*) que puede ser de tipo simple, como un número o una cadena de caracteres, o de tipo complejo, en el cual se tiene una agrupación de diferentes valores. Para este caso el resultado de la ejecución de *method* es un número entero (20032).

5.2.2. Estructura de los mensajes

⁵¹ Por razones de claridad se omitieron algunos elementos de los mensajes, como los tipos de datos de los parámetros de entrada y salida, además de algunos atributos XML de los elementos SOAP-ENV:Envelope y SOAP-ENV:Body.

Uno de los aspectos más importantes del protocolo de comunicación es la estructura de los mensajes que serán intercambiados por los diferentes sistemas que conforman la plataforma. Estos mensajes deberán cumplir con las siguientes características:

- **Estructura simple:** Los mensajes deberán ser estructurados como solicitudes SOAP simples, en las cuales se tienen tipos de datos que son independientes de la implementación de los servicios que hacen uso de ellos, como números, cadenas de caracteres, objetos simples y arreglos de objetos simples. Los tipos de datos complejos deberán ser transformados en documentos XML, y la responsabilidad de interpretar estos documentos recaerá sobre los sistemas que utilizan los servicios de la plataforma⁵².
- **Independencia del esquema de metadatos:** Con el objetivo de ofrecer un alto nivel de interoperabilidad, la estructura de los mensajes no deberá depender del esquema de metadatos de los objetos de aprendizaje sobre los cuales se están realizando las operaciones. Por ejemplo para el servicio de búsqueda no se podrá realizar ningún tipo de suposición acerca del esquema de metadatos o la versión de los metadatos almacenados en determinado repositorio.
- **Mensajes sincrónicos:** En la primera implementación de la plataforma sólo se hará uso de mensajes de tipo solicit-response, en los cuales un sistema envía una solicitud y espera una respuesta por parte de su interlocutor. Se estima que en un futuro se deberán implementar otro tipo de mensajes, como One-Way o Notification, para proporcionar soporte completo a los métodos Gather y Alert de la especificación IMS DRI.

5.2.3. Identificación de usuarios

⁵² Aunque la mayoría de lenguajes de programación con soporte para XML permiten la serialización de objetos complejos en un documento XML, no es recomendable transferir objetos dependientes de alguna plataforma de programación, como J2EE o .NET, debido a que se limita la interoperabilidad entre diferentes implementaciones.

Debido a que la especificación IMS DRI no define de forma explícita un mecanismo para permitir la identificación de usuarios, en el diseño actual del protocolo se optó por utilizar la estrategia de identificación basada en llaves, propuesta por Google (2004). Esta estrategia consiste en incluir dentro de los mensajes un parámetro adicional, que representa la llave de usuario, y que le permite al proveedor del servicio determinar quien realizó la llamada al método (ver Fragmento de código 4). Sin embargo este mecanismo no ofrece por sí mismo un nivel óptimo de seguridad, debido a que la llave de usuario es transportada sin ningún tipo de codificación. Se estima que el desarrollo posterior del protocolo permitirá implementar algunas estrategias de seguridad, basadas en llaves públicas y privadas, o por medio del uso de protocolos seguros, como HTTP sobre SSL.

Fragmento de código 4. Mensaje SOAP que incluye la llave de usuario

```
<SOAP-ENV:Envelope>
  <SOAP-ENV:Body>
    <method>
      <key>
        eff1e1dfb829a074310848ed4a4e05493
      </key>
      <param>
        10016
      </param>
    </method>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

5.2.4. Definición de las funciones que ofrece el protocolo de comunicación

Las funciones que conforman el protocolo de comunicación permiten la llamada remota de servicios por medio de mensajes SOAP. La ventaja de utilizar este tipo de mensajes consiste en que es posible implementar las interfaces de servicios en diferentes lenguajes de programación, plataformas y sistemas operativos (Por ejemplo, el cliente podría estar implementado en Java, mientras que el proveedor de

servicios podría ser implementado en .NET, o PHP, o viceversa). Lo anterior permite aprovechar las actuales implementaciones de ambientes para educación en línea o repositorios, al agregar una nueva capa de funcionalidad que toma como base los servicios existentes, y ofrece los servicios planteados por el protocolo.

A continuación se presentan las funciones del protocolo que se definieron en el presente trabajo, y que buscan ofrecer una plataforma sobre la cual es posible realizar la gestión y el intercambio de objetos de aprendizaje entre diferentes sistemas (LMS, repositorios o aplicaciones de usuario). La organización de estas funciones y su papel dentro de la plataforma se explicará con más detalle en la sección 5.3, arquitectura de la plataforma de intercambio de objetos de aprendizaje. Igualmente, el Anexo 4 muestra el documento WSDL en el cual se realiza la definición de los servicios y los tipos de datos utilizados por el protocolo.

5.2.4.1. Función suscribe

Esta función se utiliza para realizar el registro de un usuario dentro de un repositorio digital de contenidos. Su objetivo es permitir que cualquier persona o sistema pueda tener acceso a los servicios ofrecidos por un determinado repositorio.

Parámetros de entrada

La función suscribe toma los siguientes parámetros de entrada:

- **name:** Nombre del usuario
- **email:** Dirección de correo
- **url:** Página web de contacto. En el caso que el usuario sea un LMS o un repositorio, el URL deberá representar la dirección de la página de ingreso al sistema.

Parámetros de salida

El resultado de la función `suscribe` es una cadena de caracteres de 32 bytes, que representa la llave única de usuario dentro del repositorio. Esta llave es calculada por medio del algoritmo MD5 sobre el correo electrónico proporcionado por el usuario. En teoría un usuario deberá realizar un solo registro, pero en la práctica es posible que un usuario obtenga varias llaves de acceso al repositorio. Esta opción no es recomendable, ya que los recursos agregados al repositorio se asocian con la llave de usuario, y sólo el propietario puede modificar la información de dichos recursos.

Condiciones de error

Si el proceso de registro falla, el valor devuelto por esta función será una cadena vacía.

5.2.4.2. Función `getReferenceModelList`

Esta función permite obtener un listado de los modelos de referencia que soporta el repositorio de materiales, y es útil para determinar los esquemas de metadatos que pueden ser utilizados para describir los objetos de aprendizaje.

Parámetros de entrada

- **key:** El único parámetro de entrada de la función `getReferenceModelList` es la llave única de usuario, obtenida por medio de la suscripción al repositorio.

Parámetros de salida

El resultado de la función `getReferenceModelList` es una cadena de caracteres, que representa un documento XML en el cual se describen los modelos de referencia que soporta el repositorio al cual se realizó la llamada. Para cada modelo se retorna una llave única que lo identifica dentro del repositorio (según el estándar que utilice), y el nombre que se presenta al usuario. El Fragmento de código 5 presenta un posible resultado de una llamada a la función `getReferenceModelList`.

Fragmento de código 5. Resultado de la llamada a la función `getReferenceModelList`

```

<table>
  <referenceModel>
    <refkey>SCORM_CAM_v1.3</refkey>
    <refname>SCORM 1.3</refname>
  </referenceModel>
  <referenceModel>
    <refkey>llave del modelo 2</refkey>
    <refname>nombre del modelo 2</refname>
  </referenceModel>
  . . .
</table>

```

Condiciones de error

La función `getReferenceModelList` retorna una cadena vacía en caso de ocurrir un error.

5.2.4.3. Función `getComponentList`

Esta función permite obtener una lista en forma de documento XML de los componentes de un modelo de referencia que son soportados por el repositorio. En el caso de SCORM, los componentes son Assets, SCO y Agregaciones de contenidos.

Parámetros de entrada

La función `getComponentList` tiene dos parámetros de entrada:

- **key:** Llave única del usuario en el repositorio.
- **model:** Llave única del modelo de referencia del cual se desean conocer sus componentes. Esta llave puede ser obtenida por medio de la función `getComponentList`.

Parámetros de salida

El resultado de esta función es una cadena de caracteres, que representa un documento XML el cual se agrupan los diferentes componentes del modelo de referencia especificado en el llamado de la función. El Fragmento de código 6 muestra el resultado de la invocación de la función `getComponentList` para el modelo de referencia SCORM 1.3.

Fragmento de código 6. Resultado del llamado a la función `getComponentList`

```
<table>
  <comp>
    <comkey>contentAggregation_v1.3</comkey>
    <comname>Agregacion de contenidos</comname>
    <comdescription>Una agregacion de contenidos es un paquete que
    contiene una estructura de organizacion de contenidos (archivo
    imsmanifest.xml) y los recursos (contenidos) asociados a esta
    estructura. Generalmente una agregacion de contenidos representa un
    curso.</comdescription>
  </comp>
  <comp>
    <comkey>sco_v1.3</comkey>
    <comname>SCO</comname>
    <comdescription>SCO - Shareable Content Object - Objeto de
    contenido compartible - Es una agrupacion de medios web (texto,
    imágenes, animaciones, etc) que puede ser rastreada por un sistema
    gestor de aprendizaje.</comdescription>
  </comp>
  <comp>
    <comkey>asset_v1.3</comkey>
    <comname>Asset</comname>
    <comdescription>Al igual que los SCO, un Asset es una
    agrupacion de medios web (texto, imágenes, animaciones, etc). Los
    Asset no pueden ser rastreados por los sistemas gestores de
    aprendizaje.</comdescription>
  </comp>
</table>
```

Condiciones de error

La función `getComponentList` retorna una cadena de caracteres vacía en el caso de ocurrir un error.

5.2.4.4. Función addResource

Esta función permite agregar un nuevo objeto de contenido (asset, sco o agregación de contenido) dentro del repositorio. Para realizar su ejecución es necesario que el objeto de contenido se encuentre empaquetado como un archivo comprimido, de acuerdo con la especificación de empaquetado de contenidos de IMS (IMS Content Packaging). Se debe tener en cuenta que los metadatos del objeto no deberán ser incluidos dentro del archivo comprimido, estos se envían utilizando la función loadMetadata, que se explicará posteriormente.

Parámetros de entrada

Los parámetros de entrada de la función addResource son:

- **key:** Identificador único de usuario.
- **model:** Modelo de referencia, que define el esquema de metadatos a utilizar para el recurso (SCORM 1.3 define a LOM v1.0 como esquema de metadatos).
- **component:** Componente del modelo de referencia (Asset, SCO, Agregación de contenidos). La selección de este componente es necesaria para determinar cuales campos del esquema de metadatos son requeridos para el nuevo recurso.
- **name, description, language:** Datos generales del recurso, una vez que este haya sido agregado formarán parte de sus metadatos.
- **url:** Dirección web en donde el recurso se encuentra disponible, en forma de archivo comprimido. El repositorio realizará las acciones necesarias para acceder a este recurso y almacenarlo de forma local.
- **launch:** Este parámetro define el nombre del archivo principal que deberá ser desplegado al usuario, una vez que se solicite el acceso al recurso. Es útil cuando un recurso posee varias páginas HTML, y una de ellas es la página principal, o cuando el recurso está formado por sólo un archivo.

- **filename:** Nombre de archivo original que tenía el recurso en el computador del cliente.
- **unzip:** A pesar que la mayoría de recursos se almacenan como archivos comprimidos, es posible que un recurso sea por sí mismo un archivo comprimido. Este parámetro se utiliza para definir si el recurso debe ser descomprimido.

Parámetros de salida

La función `addResource` retorna un tipo complejo de datos, que agrupa los siguientes parámetros:

- **ResourceId:** En caso de éxito, este parámetro tomará el valor del identificador único del objeto que se asignó de forma automática en el repositorio.
- **Uri:** En caso de éxito, este parámetro toma el valor de la dirección URL en la cual se puede tener acceso directo al recurso. Esta dirección se puede utilizar para crear hipervínculos al objeto de aprendizaje dentro de otra estructura de contenidos.
- **ErrorCode:** En caso de error, este parámetro contiene el código de error generado por el repositorio.
- **ErrorDescription:** En caso de error, contiene una descripción del error presentado.

Condiciones de error

La Tabla 14 muestra los diferentes códigos de error que genera el repositorio, en caso de que ocurra una falla en el proceso de agregar un nuevo objeto de aprendizaje. Estos códigos de error pueden ser utilizados por el usuario del servicio para determinar la naturaleza del error, y de esta forma tomar las contingencias del caso, e informar al usuario final.

Tabla 14. Errores que se pueden presentar en la adición de un objeto de aprendizaje

Código	Descripción
0	(No existe error)
100	Error no especificado.
101	No se puede realizar la conexión para leer el recurso. El URL especificado no es válido
101	Error en la lectura remota del recurso
103	Recurso ya existe dentro del repositorio
105	No es posible guardar el recurso dentro del repositorio
106	No es posible descomprimir el recurso
107	Llave de usuario desconocida
108	Componente de modelo de referencia desconocido
109	Recurso no existe dentro del repositorio

5.2.4.5. Función loadMetadata

Esta función permite cargar los metadatos de un objeto de aprendizaje dentro del repositorio, que deberán estar ajustados al modelo de referencia y el componente que se definió en el momento de agregar el recurso.

Parámetros de entrada

- **key:** Llave única del propietario del objeto de aprendizaje
- **resource:** Llave única del recurso
- **metadata:** Una cadena de caracteres que contiene los nuevos metadatos, en forma de un documento XML.

Parámetros de salida

En caso que la función loadMetadata se ejecute correctamente, el repositorio retornará una cadena vacía.

Condiciones de error

Si el proceso de carga de metadatos falla, el repositorio retornará una cadena de caracteres, que podrá tener los siguientes valores:

Código	Descripción
0	(No existe error)
107	Llave de usuario desconocida
108	Componente de modelo de referencia desconocido
109	Recurso no existe dentro del repositorio
110	Metadatos no válidos

5.2.4.6. Función *getMetadata*

Esta función permite obtener los metadatos asociados con un objeto de aprendizaje, en forma de un documento XML.

Parámetros de entrada

La función *getMetadata* tiene los siguientes parámetros de entrada:

- **key**: Llave única de un usuario del repositorio (no requiere ser el propietario del objeto de aprendizaje)
- **resource**: Llave única del objeto de aprendizaje

Parámetros de salida

La función *getMetadata* retorna una cadena de caracteres, que contiene el documento XML con los metadatos del objeto de aprendizaje. El Fragmento de código 7 presenta una cadena con los resultados para un Asset.

Fragmento de código 7. Resultado de la función *getMetadata*

```
<lom>
  <general>
    <identifier>
      <catalog>URN</catalog>
      <entry>E7192703-B322-CA07-138D-C0F91DC45DE8</entry>
    </identifier>
```

```

<title>
  <string language="es">Hidráulica - Ecuaciones de energía
y momento</string>
</title>
<language>es</language>
<description>
  <string language="es">
    Este applet de Java muestra dos conceptos de
Hidráulica de canales abiertos - la conservación de Energía y del
Momento.
  </string>
</description>
</general>
<metaMetadata>
  <identifier>
    <catalog>URI</catalog>
    <entry>E7192703-B322-CA07-138D-C0F91DC45DE8</entry>
  </identifier>
  <language>es</language>
  <metadataSchema>LOMv1.0</metadataSchema>
  <metadataSchema>SCORM_CAM_v1.3</metadataSchema>
</metaMetadata>
<technical>
  <format>Java Applet</format>
</technical>
<rights>
  <cost>no</cost>
</rights>
</lom>

```

Condiciones de error

En caso de error esta función retornará uno de los códigos de error mostrados en la Tabla 15.

Tabla 15. Códigos de error para la función getMetadata

Código	Descripción
107	Llave de usuario desconocida
109	Recurso no existe dentro del repositorio
111	No existen metadatos para el recurso

5.2.4.7. Función *metadataTextQuery*

Esta función permite buscar objetos de contenido dentro de un repositorio, con unos parámetros de búsqueda especificados por el usuario.

Parámetros de entrada

La función *metadataTextQuery* toma como entrada los siguientes parámetros:

- **key**: Llave única del usuario dentro del repositorio
- **type**: Tipo de recurso que se busca, que se representa como una o varias llaves únicas de componentes, separadas por comas.
- **text**: Patrón de texto para buscar en los metadatos de los objetos que existen en el repositorio.
- **since**: Cadena de caracteres en formato *aaaa-mm-dd* o *aaaa-mm-dd hh:mm:s*, que permite buscar los objetos recientes más recientes a. determinada fecha.

Parámetros de salida

La salida de la función *metadataTextQuery* es un arreglo de objetos simples, cada uno con los siguientes atributos:

- **Resource**: Identificador único del recurso dentro del repositorio.
- **Type**: Tipo de recurso, dentro de un modelo de referencia. El tipo es único dentro del repositorio.
- **Date**: Fecha de adición del recurso al repositorio
- **Text**: Texto dentro de los metadatos del recurso en el cual se encontró una coincidencia con el patrón de búsqueda especificado por el usuario

- **Name:** Nombre dado al recurso por su creador.
- **Description:** Descripción general del recurso.
- **Url:** Dirección web en la cual se encuentra almacenado el recurso. Esta dirección puede ser utilizada para hacer referencia al objeto de aprendizaje.
- **File:** Dirección web en la cual se encuentran compilados todos los recursos del objeto de aprendizaje, en forma de archivo comprimido. Esta propiedad permite descargar fácilmente los recursos a través de una solicitud http
- **Package:** Define si el objeto de aprendizaje es un recurso (N) o un paquete de contenidos (Y).

El Fragmento de código 8 muestra el resultado de la función `metadataTextQuery`.

Fragmento de código 8. Resultado de la función `metadataTextQuery`

```
Array ( [0] => stdClass Object (
  [Resource] => 890E78E6-6390-2973-1AD9-F1D3C3370107
  [Type] => contentAggregation
  [Date] => 2005-09-01 10:42:02 PM
  [Text] => IntroducciÃ³n a los computadores
  [Name] => IntroducciÃ³n a los computadores
  [Description] => Estructura de introducciÃ³n a los computadores
  [Url] => http://localhost:81/aula/content/.....
  [File] => http://localhost:81/aula/content/.....
  [Package] => Y
)
[1] => stdClass Object (
  [Resource] =>
http://localhost/aula/repository/edit_metadata_tree.php?resource=E7192703-B322-
CA07-138D-C0F91DC45DE8&emid=134&treid=5&objid=8 -
emid=134&treid=5&objid=8E7192703-B322-CA07-138D-C0F91DC45DE8
  [Type] => sco
  [Date] => 2005-09-10 06:23:05 PM
  [Text] => Hidraulica
  [Name] => Hidraulica de canales abiertos
  [Description] => Applet de hidraulica
  [Url] => http://localhost:81/aula/content/.....
  [File] => http://localhost:81/aula/content/.....
  [Package] => N
) )
```

Condiciones de error

La función `metadataTextQuery` retornará un arreglo vacío en el caso de ocurrir un error.

5.2.4.8. Función `metadataQueryXML`

Esta función es similar a `metadataTextQuery`, pero retorna los resultados como una cadena de caracteres, que representa el documento XML con los resultados de la búsqueda.

Parámetros de entrada

La función `metadataQueryXML` toma como entrada los siguientes parámetros:

- **key**: Llave única del usuario dentro del repositorio
- **type**: Tipo de recurso que se busca, que se representa como una o varias llaves únicas de componentes, separadas por comas.
- **text**: Patrón de texto para buscar en los metadatos de los objetos que existen en el repositorio.
- **since**: Cadena de caracteres en formato `aaaa-mm-dd` o `aaaa-mm-dd hh:mm:s`, que permite buscar los objetos recientes más recientes a. determinada fecha.

Parámetros de salida

La salida de la función `metadataQueryXML` es una cadena de caracteres, que representa un documento XML en el que se describen los objetos que cumplen con los parámetros de búsqueda especificados, por medio de los mismos campos de los objetos de resultados de la función de búsqueda `metadataTextQuery`. El Fragmento de código 9 muestra el resultado de la función `metadataQueryXML`, en el cual se obtuvieron dos resultados.

Fragmento de código 9. Resultado de la función metadataQueryXML

```

<results>
  <result>
    <resource>FCB891D5-369D-3757-7EFF-43911B133F3C</resource>
    <type>contentAggregation</type>
    <date>2005-08-22 09:52:50 PM</date>
    <text>Photoshop Lineal</text>
    <name>Photoshop Lineal</name>
    <description>Curso de Photoshop con secuenciación lineal</description>
    <url>http://localhost/aula/content/....</url>
    <file>http://localhost/aula/content/....</file>
    <package>Y</package>
  </result>
  <result>
    <resource>05A6AA14-AABA-43E1-F69B-EF712BF807E1</resource>
    <type>contentAggregation</type>
    <date>2005-08-25 04:52:24 AM</date>
    <text>Photoshop_None</text>
    <name>Photoshop_None</name>
    <description>Curso de Photoshop sin ningún tipo de secuenciación</description>
    <url>http://localhost/aula/content/...</url>
    <file>http://localhost/aula/content/...</file>
    <package>Y</package>
  </result>
</results>

```

Condiciones de error

En caso de error, la función metadataQueryXML retorna una cadena de caracteres vacía.

5.2.4.9. Función getResourceList

Esta función tiene el mismo comportamiento que metadataQueryXML, pero retorna solamente los objetos de aprendizaje que pertenecen a determinado usuario.

Parámetros de entrada

La función getResourceList tiene los siguientes parámetros de entrada:

- **key**: Llave única del usuario dentro del repositorio

- **text:** Patrón de búsqueda

Parámetros de salida

El resultado de esta función es una cadena de caracteres similar al obtenido en la función `metadataQueryXML`. En caso que la búsqueda no arroje resultados, se retornará una cadena vacía.

Condiciones de error

La función `getResourceList` retorna una cadena vacía en caso de error.

5.2.4.10. Función `getResourceInfo`

Esta función permite obtener la información general de un recurso, representada por su nombre y descripción.

Parámetros de entrada

Los parámetros de entrada de la función `getResourceInfo` son:

- **key:** Llave única de usuario en el repositorio
- **resource:** Identificador único del recurso

Parámetros de salida

El resultado de esta función es un tipo complejo que agrupa los siguientes parámetros:

- **name:** Nombre del recurso
- **description:** Descripción del recurso.

Condiciones de error

En caso de error esta función retorna un objeto vacío.

5.2.4.11. Función *updateResourceInfo*

Esta función se utiliza para actualizar la información general de un recurso (su nombre y su descripción), que se especificaron en el momento de agregarlo al repositorio.

Parámetros de entrada

La función `updateResourceInfo` toma como entrada los siguientes parámetros:

- **key**: Llave única del usuario en el repositorio
- **resource**: Identificador único del recurso
- **name**: Nuevo nombre dado al recurso
- **description**: Nueva descripción para el recurso

Parámetros de salida

La ejecución correcta de esta función retorna una cadena de caracteres con el número 0, el cual indica que no existió error durante la actualización de los datos.

Condiciones de error

La función `updateResourceInfo` retorna una cadena de caracteres, que representa el código de error que se obtuvo de su ejecución. Estos valores se muestran en la v.

Código	Descripción
0	(No existe error)
100	Error no especificado.

5.2.4.12. Función *isOnline*

La función `isOnline` se utiliza para verificar si un repositorio se encuentra en línea.

Parámetros de entrada

Esta función toma como parámetro de entrada una cadena de caracteres llamada token, que se envía al repositorio que se desea verificar.

Parámetros de salida

El resultado de esta función es la misma cadena de caracteres enviada como entrada.

Condiciones de error

Esta función no genera condiciones de error.

5.2.4.13. Función *isValidReferenceModel*

La función `isValidReferenceModel` es útil para verificar si un repositorio soporta determinado modelo de referencia.

Parámetros de entrada

Esta función toma los siguientes parámetros de entrada:

- **key**: Llave única del usuario dentro del repositorio
- **model**: Llave única del modelo de referencia dentro del repositorio.

Parámetros de salida

Esta función retorna una cadena de caracteres con los valores "true" o "false".

Condiciones de error

En caso de error, esta función retorna la cadena de caracteres "false".

5.2.4.14. Función *isValidComponent*

La función *isValidComponent* permite verificar si un componente es soportado dentro del repositorio.

Parámetros de entrada

Los parámetros de entrada de la función *isValidComponent* son:

- **key**: Llave única del usuario dentro del repositorio
- **model**: Llave del modelo de referencia al cual pertenece el componente a verificar
- **component**: Llave única del componente a verificar

Parámetros de salida

La salida de esta función es una cadena de caracteres, que toma los valores de “true” o “false”, según el caso.

Condiciones de error

En caso de error, la función *isValidComponent* retorna “false”.

5.3. ARQUITECTURA DE LA PLATAFORMA DE INTERCAMBIO DE OBJETOS DE APRENDIZAJE

La plataforma de intercambio de materiales basados en objetos de aprendizaje que se diseñó involucra una gran cantidad de tareas, entre las cuales se destacan la creación, la construcción, la reutilización la integración, el almacenamiento, la administración, la búsqueda y la distribución de recursos a través de Internet. El diseño de la arquitectura y la implementación que se realizaron en el presente trabajo consideran todas estas acciones, con el objetivo de proporcionar una herramienta que permita poner en práctica, estudiar y mejorar las bases

tecnológicas, conceptuales y metodológicas sobre los cuales se fundamentan los modelos de referencia, los estándares y las especificaciones para la creación de materiales, y que en un nivel más amplio establecen los lineamientos para la educación en línea. La Figura 35 presenta la arquitectura funcional que se planteó en el presente trabajo, fundamentada en el modelo de referencia SCORM, la Especificación para la Interoperabilidad de Repositorios Digitales de IMS, y el modelo CORDRA.

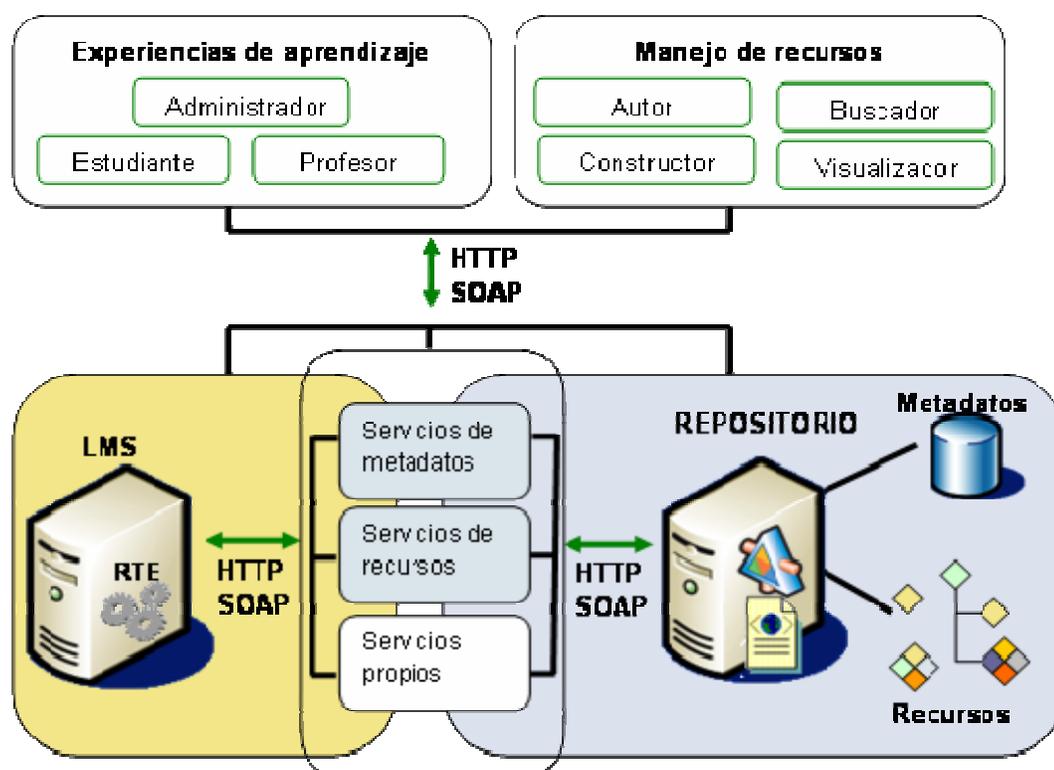


Figura 35. Arquitectura de la plataforma de intercambio de objetos de aprendizaje

A continuación se presentan los componentes que integran la arquitectura planteada, clasificados en roles, interfaz de servicios, componentes de LMS y componentes de repositorio.

5.3.1. Roles

Los roles (obtenidos partir del análisis de los casos de uso generales y de los roles propuestos por la especificación IMS DRI) pueden ser vistos como aquellos usuarios (personas o herramientas software) que hacen uso de los servicios proporcionados por la plataforma, de acuerdo con unos objetivos individuales o colectivos. Siguiendo los planteamientos de IMS DRI, los roles son dinámicos y dependen del contexto de su aplicación. Un usuario puede profesor, buscador, constructor o visualizador en una experiencia educativa, al tiempo que es estudiante y visualizador en otra experiencia. A continuación se presenta una breve reseña de los roles y las funciones más comunes que pueden ejecutar dentro de la plataforma.

5.3.1.1. Roles desde el punto de vista del manejo de los recursos

Desde el punto de vista del manejo de los recursos, los roles son asumidos por herramientas software (LMS, repositorios y demás aplicaciones), que se clasifican en autores, constructores, buscadores y visualizadores. Los autores son los encargados de desarrollar los materiales básicos (Assets y SCO), a partir de los cuales se crean materiales de mayor complejidad, con un objetivo de aprendizaje específico. Los constructores toman los recursos desarrollados por los autores y los integran en unidades mayores de material (SCO o agregaciones de contenidos), que pueden ser utilizados para apoyar una experiencia educativa. Tanto los autores como los constructores tienen la posibilidad de reutilizar los recursos desarrollados anteriormente, con lo cual actúan como buscadores y visualizadores, para localizar aquellos recursos que pueden ser contextualizados dentro de una nueva experiencia educativa.

5.3.1.2. Roles desde el punto de vista de la experiencia de aprendizaje

Desde este punto de vista, los roles más importantes para el desarrollo de una experiencia de aprendizaje son los estudiantes, los profesores y los administradores. Los estudiantes son los beneficiarios principales de los recursos almacenados en los repositorios digitales, y en la mayoría de ocasiones actúan como visualizadores y buscadores. Los profesores como encargados de planear y

administrar la experiencia educativa, y para tal fin deben actuar como buscadores, visualizadores y constructores, para proporcionar los materiales educativos que apoyen el desarrollo de la experiencia. Por su parte, los administradores deben contar con las herramientas necesarias para permitir que los profesores y los estudiantes puedan tener acceso a los servicios y los recursos proporcionados por uno o más repositorios. Igualmente todos los usuarios (profesores, estudiantes o administradores) pueden actuar como creador, de acuerdo con las necesidades de desarrollo de nuevos materiales.

5.3.2. Interfaz de servicios de la plataforma

La interfaz de servicios es el componente central de la plataforma, ya que ofrece los medios necesarios para permitir la interoperabilidad entre los diferentes roles de usuario y los sistemas, en particular los LMS y los repositorios. Para su desarrollo se tomó como base la Especificación para la Interoperabilidad de Repositorios IMS DRI, y se incluyeron algunos aspectos que no se tratan dentro de la arquitectura funcional que plantea esta especificación.

Los servicios que ofrece la plataforma de intercambio se pueden clasificar en servicios de metadatos, servicios de recursos y servicios propios de la implementación. Además durante la implementación fue necesario definir una cuarta categoría de servicios (llamada servicios generales), que permiten realizar algunas tareas generales entre los componentes del sistema. Esta categoría no se incluyó en el diseño de la arquitectura, pero se estima que el estudio de la implementación desarrollada permitirá perfeccionar el modelo propuesto, para que esta categoría incluya otros servicios generales, y pueda ser tomada en cuenta dentro de la arquitectura funcional de la plataforma. A continuación se presenta una descripción las categorías de servicios implementadas.

5.3.2.1. Servicios generales

Los servicios generales de la plataforma no se encuentran contemplados dentro de la especificación IMS DRI, pero surgieron a partir de la necesidad de facilitar la

interacción entre los diferentes sistemas que requieren utilizar los servicios de metadatos y recursos. A continuación se presentan los servicios generales desarrollados para la plataforma de intercambio.

Suscripción

Este servicio permite realizar el registro de un usuario (persona, LMS o repositorio) dentro de un repositorio de recursos o metadatos. Para realizar el registro se deben especificar unos datos generales (nombres, correo electrónico y una dirección URL), y con estos datos se invoca el servicio Web que realiza la suscripción (Ver Fragmento de código 10).

Fragmento de código 10. Llamada al servicio web de suscripción

```
//Llamada al servicio Web de suscripción

ini_set('soap.wsdl_cache_enabled', 0);
$client = new SoapClient($CONFIG->repository_wsdl);
set_time_limit(0);

$result = $client->suscribe(Array("name" => $name),
                             "email" => $email),
                             "url" => $url));
$key = $result->suscribeResult;
```

Una vez que el método *suscribe* ha sido invocado, el repositorio deberá registrar al nuevo usuario y regresar la llave única de usuario, generada con el algoritmo MD5, o una cadena vacía, en el caso de ocurrir un error. La llave de usuario deberá ser utilizada en todas las llamadas a los servicios de metadatos y de recursos.

Es importante destacar que el diseño del servicio de suscripción (y de la mayoría de servicios) es simple, para permitir que tanto las personas como las aplicaciones puedan hacer uso de ellos, sin conocer en profundidad los detalles de su implementación.

Verificación de conexión

Este servicio permite verificar la conexión hacia un repositorio, y su funcionamiento es sencillo: se envía una solicitud al repositorio, en la cual se incluye un texto. Si el repositorio se encuentra en línea, deberá retornar el texto que se envió en la llamada. El Fragmento de código 11 muestra la programación necesaria para invocar el servicio de verificación de conexión.

Fragmento de código 11. Verificación de la conexión con un repositorio

```
$soapClient = new SoapClient($wsdl);  
  
$res = $soapClient->isOnline("Hello");  
if (is_soap_fault($res)) {  
    //Error de conexión  
}else {  
    //Conexión exitosa  
}
```

5.3.2.2. Servicios de metadatos

Esta categoría integra todos aquellos servicios basados en IMS DRI que permiten realizar diversas tareas con los metadatos almacenados en un repositorio. Además de las funciones básicas de metadatos (adicionar, modificar y exportar), fue necesario implementar algunos servicios adicionales (no contemplados en IMS DRI), con el fin de permitir que los usuarios del repositorio obtengan información general acerca de los modelos de referencia y los componentes soportados. A continuación se presenta una descripción de los servicios de metadatos implementados.

Obtener meta-información del repositorio

Estos servicios permiten obtener información acerca de los modelos de referencia soportados por el repositorio, y los componentes de estos modelos que pueden ser almacenados y administrados. Existen cuatro métodos que pueden ser invocados por los usuarios de los repositorios: Obtener modelos de referencia y componentes, y validar modelos de referencia y componentes.

El método para obtener los modelos de referencia permite que el usuario recupere una lista con la información general de los modelos de referencia soportados por el repositorio. El Fragmento de código 12 muestra el código en PHP requerido para obtener el listado de modelos de referencia.

Fragmento de código 12. Obtener el listado de los modelos de referencia disponibles en un repositorio

```
try {
    ini_set('soap.wsdl_cache_enabled', 0);
    set_time_limit(0);
    ini_set('default_socket_timeout', 3);

    $soapClient = new SoapClient($wsdl_path, Array('trace' => 1));
    $ret= $soapClient->getReferenceModelList(Array( "key" => "$key"));
    $xml = $ret->getReferenceModelListResult;
    if ($xml) {
        //Procesar el listado de modelos de referencia
    }
} catch(Exception $e) {
    //Error al realizar la solicitud
}
```

El resultado de la llamada al método *getReferenceModelList* es un documento XML, en el cual se condensa la información de los modelos de referencia que son soportados por el repositorio seleccionado (Fragmento de código 13). Este documento proporciona el nombre de cada modelo de referencia, y la llave (*refkey*), con la cual se conoce a cada modelo dentro del repositorio. Esta llave está definida para cada modelo de referencia y corresponde a un valor único, que contiene su nombre y versión. En el caso de SCORM 1.3 (2004), *refkey* tiene el valor SCORM_CAM_v1.3.

Fragmento de código 13. Resultados de la llamada obtener modelos de referencia

```
<table>
  <referenceModel>
    <refkey>SCORM_CAM_v1.3</refkey>
    <refname>SCORM 1.3</refname>
  </referenceModel>
</table>
```

De forma similar, el método para obtener la información acerca de los componentes de los modelos de referencia que soporta un repositorio permite obtener un listado e forma de documento XML, a partir del cual las aplicaciones pueden construir interfaces de usuario que permiten la inclusión de metadatos o la búsqueda de información, entre otras tareas. El Fragmento de código 14 presenta la codificación necesaria para invocar el método para obtener los componentes de un modelo de referencia dado.

Fragmento de código 14. Obtener el listado de componentes para un modelo de referencia dado

```
try {
    ini_set('soap.wsdl_cache_enabled', 0);
    set_time_limit(0);
    ini_set('default_socket_timeout', 3);
    $soapClient = new SoapClient($wsdl_path, Array('trace' => 1));
} catch(Exception $e) {
    //Error al invocar el método
}
$ret= $soapClient->getComponentList(Array( "key" => "$key",
                                           "model" => $model));
$xml = $ret->getComponentListResult;
```

El resultado de la llamada al método *getComponentList* es el documento XML mostrado en el Fragmento de código 15.

Fragmento de código 15. Listado de los componentes de un modelo de referencia

```
<table>
  <comp>
    <comkey>contentAggregation</comkey>
    <comname>Agregacion de contenidos</comname>
    <comdescription>Una agregacion de contenidos es un paquete que
contiene una estructura de organizacion de contenidos (archivo
imsmanifest.xml) y los recursos (contenidos) asociados a esta
estructura. Generalmente una agregacion de contenidos representa un
curso.</comdescription>
  </comp>
  <comp>
    <comkey>sco</comkey>
    <comname>SCO</comname>
    <comdescription>SCO - Shareable Content Object - Objeto de
contenido compartible - Es una agrupacion de medios web que puede
ser rastreada por un sistema gestor de aprendizaje.</comdescription>
  </comp>
```

</table>

Finalmente, las llamadas a los métodos de validación permiten verificar si un repositorio soporta determinado modelo de referencia y componente. El Fragmento de código 16 presenta el código en PHP que permite verificar si un repositorio soporta agregaciones de contenido del modelo de referencia SCORM 1.3.

Fragmento de código 16. Validación de un componente del modelo de referencia SCORM

```
try {
    $client = new SoapClient($wsdl_path);
    $ret = $client->isValidComponent(
        Array("key" => utf8_encode("$key"),
            "model" => utf8_encode("SCORM_CAM_v1.3"),
            "component" => utf8_encode("contentAggregation")));
    $res = $ret->isValidComponentResult;
    if ($res == "true") {
        //El repositorio soporta agregaciones de contenidos SCORM 1.3
    } else {
        //Estos componentes no son soportados
    }
} catch (Exception $e) {
    //Error al realizar la consulta
}
```

Como se puede observar, el valor del retorno para los métodos de validación son las cadenas de caracteres “true” o “false”, según el caso.

Carga de metadatos

Este servicio permite incluir los metadatos para un objeto de aprendizaje dentro del repositorio, en forma de un documento XML. La configuración actual de este servicio requiere que el recurso exista previamente en el repositorio, pero se estima que en un futuro la carga de metadatos podrá ser utilizada para crear repositorios de metadatos, que hacen referencia a recursos que no se encuentran almacenados dentro del mismo repositorio, y que están disponibles a través de Internet. El Fragmento de código 17 muestra el código PHP necesario para invocar el servicio de carga de metadatos de un repositorio, para un objeto previamente agregado a la colección de recursos del repositorio.

Fragmento de código 17. Llamada al método para cargar los metadatos de un objeto de aprendizaje

```

ini_set('soap.wsdl_cache_enabled', 0);
$soapClient = new SoapClient($CONFIG->repository_wsdl,
                             Array('trace' => 1));
set_time_limit(0);
$result = $soapClient->loadMetadata(Array("key" => "$key",
                                         "resource" => "$resource",
                                         "metadata" => "$contents"));
if ($result->loadMetadataResult == 0) {
    //Carga de metadatos exitosa
}else {
    //Error al cargar los metadatos
}

```

La llamada al método del servicio web incluye la llave del usuario, el identificador del recurso y el documento XML que contiene los metadatos, en forma de una cadena de caracteres (*contents*). Al recibir estos datos, el repositorio realizará una validación preliminar del documento XML, y luego hará uso del esquema de metadatos almacenado en su base de datos para cargar valores para el objeto. Finalmente el repositorio retorna el número 0 en caso de éxito, o un valor diferente de cero en el caso contrario.

Descarga de metadatos

Este servicio puede ser invocado por los usuarios (personas o herramientas software) para obtener los metadatos de un recurso, que se encuentran almacenados dentro del repositorio. Los metadatos retornados cumplen con el esquema de metadatos definido para el objeto de aprendizaje (LOM 1.0 en el caso de SCORM), y pueden ser como base para la creación de repositorios de metadatos (función "Pull"/Gather de la especificación IMS DRI). A pesar que la implementación actual de la plataforma soporta cualquier esquema de metadatos que pueda ser representado como XML, cada objeto de aprendizaje sólo puede ser descrito por un esquema de metadatos. El Fragmento de código 18 presenta el código en PHP requerido para invocar el método de descarga de metadatos para un recurso.

Fragmento de código 18. Llamada al método para descargar los metadatos de un recurso

```

ini_set('soap.wsdl_cache_enabled', 0);

$soapClient = new SoapClient($CONFIG->repository_wsdl,
                             Array('trace' => 1));
set_time_limit(0);

$result = $soapClient->getMetadata(Array("key" => "$key",
                                         "resource" => "$resource"));

if ($result->getMetadataResult) {
    //Metadatos retornados
} else {
    //Error al descargar los metadatos
}

```

La invocación a este método retorna un documento XML con los metadatos del objeto de aprendizaje, o una cadena vacía en el caso de existir error.

Búsquedas sobre los metadatos

Este es uno de los servicios más utilizados por las herramientas que tienen acceso a los recursos almacenados en un repositorio, y permite obtener un listado de los objetos de contenido que cumplen algún criterio de búsqueda. La implementación actual de la plataforma ofrece dos métodos diferentes para buscar y localizar recursos, que difieren en la forma como se obtienen los resultados. El primero permite recuperar los datos en forma de un arreglo de objetos (similar a Google Search) (2004), y el segundo presenta los resultados en forma de un documento XML que puede ser interpretado y transformado en estructuras de datos complejas (por ejemplo, los resultados de se pueden transformar de forma transparente en un Dataset de la plataforma Microsoft .NET⁵³).

⁵³ .NET es la plataforma de desarrollo de aplicaciones de Microsoft, cuya arquitectura está basada en gran parte en aplicaciones distribuidas y servicios Web XML.

El Fragmento de código 19 presenta el código PHP para invocar el método de búsqueda, que obtiene los resultados como un arreglo de objetos (formato Google Search).

Fragmento de código 19. Invocación al servicio de búsqueda (método 1)

```
$result = $soapClient->metadataTextQuery(Array("key" => "$key",
                                             "type" => "$type",
                                             "text" => utf8_encode("$text"),
                                             "since" => "$date"));
$res = $result->metadataTextQueryResult;

if ($res) {
    //Búsqueda exitosa, $res es un arreglo de objetos
} else {
    //Error en la búsqueda
}
```

Los resultados de la invocación de este método están organizados como un arreglo de objetos simples, cuyas propiedades se muestran en el Fragmento de código 20⁵⁴.

Fragmento de código 20. Formato de objetos del resultado del método 1 de búsqueda

```
stdClass Object
(
    [Resource] => 890E78E6-6390-2973-1AD9-F1D3C3370107
    [Type] => contentAggregation
    [Date] => 2005-09-01 10:42:02 PM
    [Text] => IntroducciÃ³n a los computadores
    [Name] => IntroducciÃ³n a los computadores
    [Description] => Estructura de introducciÃ³n a los computadores
    [Url] => http://localhost:81/aula/content/.....
    [File] => http://localhost:81/aula/content/.....
    [Package] => Y
)
```

La segunda estrategia de búsqueda es utilizar el método que retorna los resultados como un documento XML. El Fragmento de código 21 muestra la codificación en PHP para realizar la búsqueda.

⁵⁴ Los caracteres extraños en la palabra Introducción se presentan debido a que la codificación que utilizan los servicios web es UTF-8, la cual no tiene acentos. Para utilizar estos datos es necesario decodificar los resultados (en PHP se puede hacer uso de la función `utf8_decode`).

Fragmento de código 21. Invocación al servicio de búsqueda (método 2)

```
$result = $soapClient->metadataQueryXML(Array("key" => "$key",
                                             "type" => "$type",
                                             "text" => utf8_encode("$text"),
                                             "since" => "$date"));

$res = $result->metadataQueryXMLResult;
```

De forma similar al primer método de búsqueda, los resultados de esta invocación son una lista con los recursos que cumplen con el criterio de búsqueda especificado. En este caso los resultados están estructurados como un documento XML, en el cual cada resultado individual posee los mismos campos descritos en el método anterior, esta vez estructurados como XML. El Fragmento de código 22 muestra el documento con los resultados obtenidos de una solicitud de búsqueda, codificados con UTF-8.

Fragmento de código 22. Resultados de la búsqueda de recursos (método 2)

```
<results>
  <result>
    <resource>890E78E6-6390-2973-1AD9-F1D3C3370107</resource>
    <type>contentAggregation</type>
    <date>2005-09-01 10:42:02 PM</date>
    <text>IntroducciÃ³n a los computadores</text>
    <name>IntroducciÃ³n a los computadores</name>
    <description>Estructura de introducciÃ³n ..</description>
    <url>http://localhost:81/aula/content/2005...</url>
    <file>http://localhost:81/aula/content/2005/...</file>
    <package>Y</package>
  </result>
</results>
```

Búsqueda de recursos propios

Este método es similar a los métodos de búsqueda de metadatos, pero sólo retorna una lista con los registros de metadatos de los objetos de aprendizaje que son propiedad del usuario. El resultado de este método es un documento XML, similar al documento obtenido al realizar el segundo método de búsqueda de metadatos, mencionado anteriormente. El Fragmento de código 23 muestra la solicitud de un

usuario para obtener los registros de metadatos de los recursos propios almacenados en determinado repositorio.

Fragmento de código 23. Consulta de los registros de metadatos para recursos propios

```

$client = new SoapClient($repository_path);
$ret = $client->getResourceList(Array("key" => utf8_encode("$key"),
                                     "text" => utf8_encode("$text")));

$res = $ret->getResourceListResult;
if ($res) {
    //El usuario posee recursos dentro de un repositorio
} else {
    //El usuario no posee recursos dentro del repositorio
}

```

5.3.2.3. Servicios de recursos

Esta categoría reúne los servicios definidos por IMS DRI que permiten la carga la descarga de objetos de aprendizaje de un repositorio, y buscan ofrecer una interfaz al usuario que es independiente de la estrategia utilizada para la administración y el almacenamiento de los recursos. A continuación se describen los servicios de recursos más importantes de acuerdo con IMS DRI, además de otros servicios que no están definidos en esta especificación, pero surgieron durante el desarrollo del presente trabajo.

Carga de recursos

Este servicio definido por IMS DRI permite agregar nuevos recursos al repositorio, por medio de una solicitud Submit/Store. Su diseño actual requiere que el usuario especifique la llave única, el tipo de objeto de aprendizaje (que se obtiene al combinar el modelo de referencia y el componente del modelo de referencia), y unos datos generales que formarán parte de los metadatos del recurso. El Fragmento de código 24 presenta el código en PHP requerido para almacenar un nuevo recurso en un repositorio.

Fragmento de código 24. Llamada al método para adicionar un nuevo recurso

```

$result = $soapClient->addResource(
    Array( "key" => utf8_encode($key),
          "model" => utf8_encode("$model"),
          "component" => utf8_encode("$component"),
          "name" => utf8_encode("$nombre"),
          "description" => utf8_encode("$descripcion"),
          "language" => utf8_encode("$lenguaje"),
          "url" => utf8_encode("{ $url }"),
          "launch" => utf8_encode("$launch"),
          "filename" => utf8_encode($name),
          "unzip" => $unzip));

```

Una vez que el usuario ha invocado el método para adicionar el recurso, el repositorio deberá acceder a la ubicación especificada (url), y cargar el objeto de contenido en su esquema de almacenamiento local. Adicionalmente el repositorio deberá incluir los datos mínimos que se requieren para que el recurso posea una definición de metadatos válida, y para ello se utilizan los datos generales que se enviaron como parte de la solicitud de adición. Si este proceso es realizado exitosamente, el repositorio retorna un objeto con dos propiedades: Objectid y Uri. La primera propiedad representa el identificador del nuevo recurso en el repositorio, y el segundo la ubicación pública del recurso dentro del repositorio. De otro lado, si existe algún error durante el proceso de adición, el método retornará un objeto con dos propiedades (Errorid y ErrorDescription), que representan un código de error y su descripción, respectivamente.

Obtener información general del recurso

Este método permite obtener la información general del recurso (nombre y descripción), que se muestra como resultado en los métodos de búsqueda y que no forma parte de los metadatos almacenados para el objeto de aprendizaje. El Fragmento de código 25 muestra la llamada al método para obtener la información general de un recurso.

Fragmento de código 25. Obtener la información general de un recurso

```

$soapClient = new SoapClient($CONFIG->repository_wsdl);
$ret= $soapClient->getResourceInfo(Array( "key" => "$key",
                                         "resource" => utf8_encode($resource)));
if ($ret->getResourceInfoResult) {

```

```

} //Se han retornado el nombre y la descripcion del recurso

```

Actualizar la información general de un recurso

Este método es complementario al método anterior, y permite actualizar el nombre y la descripción de un objeto de aprendizaje, dentro del repositorio. El Fragmento de código 26 muestra la llamada al método para actualizar la información general de un recurso.

Fragmento de código 26. Llamada al método para actualizar la información general de un recurso

```

$result = $soapClient->updateResourceInfo(
    Array( "key" => utf8_encode($key),
          "resource" => utf8_encode($resource),
          "name" => utf8_encode("$nombre"),
          "description" => utf8_encode("$descripcion"));

```

5.3.3. Componentes de LMS

En la arquitectura planteada los LMS poseen varias características, entre las cuales sobresalen la capacidad para permitir la visualización de materiales creados a partir de objetos reutilizables, que se almacenan en diferentes repositorios accesibles por medio de Internet, la administración del progreso del estudiante durante la visualización del materiales educativos y la presentación de *tests*, y la posibilidad de construir y exportar objetos de aprendizaje para su almacenamiento en repositorios digitales de contenido. Estas características fueron implementadas siguiendo el modelo de referencia SCORM, lo que permite obtener un alto nivel de interoperabilidad de los materiales generados dentro del LMS, al tiempo que ofrece la posibilidad de utilizar materiales creados en diferentes instituciones educativas.

A continuación se presentan los componentes de LMS más importantes de la arquitectura planteada, cuya implementación se encuentra integrada dentro de la nueva versión de Aula Virtual. Para cada componente se realizará una descripción

de su funcionamiento general, y una reseña de las tablas del modelo de datos que utiliza. La versión completa del modelo de datos de Aula Virtual 2.0 puede ser consultada en el Anexo 2.

5.3.3.1. Módulo de creación y edición de estructuras de contenidos compatibles con SCORM

Las estructuras de contenidos que se definen en Aula Virtual consisten en una distribución jerárquica de temas, interpretada de acuerdo con los objetivos del curso, como secciones, capítulos, lecciones, o temas. El módulo de creación y edición permite transformar las estructuras de contenidos desarrolladas por los usuarios de Aula Virtual 2.0 en objetos de contenido que cumplen con el modelo de referencia SCORM, y de esta forma proporciona un mecanismo simple para la reutilización de los materiales en otras experiencias educativas, dentro o fuera de Aula Virtual 2.0. Las características principales de este módulo son:

- **Uso de diversas fuentes:** Los materiales que integran la estructura de contenidos pueden ser archivos HTML creados en Aula Virtual (por medio del editor integrado que posee esta herramienta), objetos de aprendizaje (Assets, SCO y Agregaciones de contenidos) reutilizados de diferentes repositorios de contenidos, o simplemente referencias a recursos disponibles a dentro o fuera de Aula Virtual y accesibles a través de Internet, por medio de un hipervínculo.
- **Generación de estructuras de contenidos que cumplen con SCORM:** Una vez creadas, las estructuras de contenidos pueden ser utilizadas en cualquier curso de Aula Virtual en el cual se encuentre inscrito el usuario. Los profesores podrán hacer uso de sus estructuras de contenidos para integrarlas dentro de los materiales del curso que se harán disponibles para los estudiantes.
- **Edición dinámica de las estructuras de contenidos:** Todas las estructuras de contenidos de un usuario pueden ser editadas en cualquier momento, para agregar nuevos elementos, y modificar o eliminar los elementos existentes. Los

elementos que han sido creados dentro de Aula Virtual pueden ser editados en línea.

La Figura 36 muestra la interfaz de Aula Virtual para crear y/o editar una estructura de contenidos. En este caso la estructura de contenidos está integrada por dos elementos: el primero es una agregación de contenidos SCORM que recopila dos cursos de PhotoShop, y el segundo elemento es un enlace a una página creada dentro de Aula Virtual.



Figura 36. Interfaz de edición de una estructura de contenidos

La reutilización de objetos de aprendizaje es un proceso sencillo, en el cual el usuario busca y localiza los objetos de aprendizaje de su interés, que se encuentran almacenados en cualquiera de los repositorios a los cuales se encuentra suscrito. Para realizar la búsqueda, el LMS invoca los servicios de metadatos de los repositorios seleccionados, y obtiene los diferentes tipos de recursos que pueden ser encontrados en cada repositorio (Ver Figura 37).

Ver recursos que contengan el siguiente texto:
Ver recursos almacenados en:

computadores

Todos los repositorios
 En los siguientes repositorios:
 Repositorio Aula Virtual
 Repositorio por defecto del Aula Virtual

SCORM 1.3
 Agregacion de contenidos
 SCO
 Asset

Buscar recursos

Figura 37. Selección de los criterios de búsqueda de recursos

Luego de realizar la búsqueda dentro de cada uno de los repositorios, el LMS obtiene la información de los recursos que cumplen con los criterios especificados, y presenta una lista al usuario. El usuario podrá visualizar y seleccionar el recurso que se ajusta a sus requerimientos (Ver Figura 38).

Insertar recurso

Listado de recursos **Cerrar Ventana**

[Nueva búsqueda](#) **Repositorio del cual se obtuvieron los resultados**

Repositorio Aula Virtual

[Agregación] **Introducción a los computadores**
 Estructura de introducción a los computadores
 Agregado al repositorio en 2005-09-01 10:42:02 PM
<http://localhost/aula/content/2005/09/01/890E78E6-6390-2973-1AD9-F1D3C...>

[Visualizar](#) [Seleccionar recurso](#)

Total de objetos del repositorio Repositorio Aula Virtual: 1

[Visualizar el recurso](#) [Seleccionar el recurso a insertar](#)

Figura 38. Interfaz para buscar y localizar objetos de aprendizaje

5.3.3.2. Módulo de Administración de repositorios

Este módulo permite definir aquellos repositorios a los cuales los usuarios de Aula Virtual (administradores, profesores y estudiantes) pueden tener acceso. La Figura 39 muestra la interfaz de administración de repositorios disponible en Aula Virtual.



Figura 39. Interfaz de administración de repositorios de Aula Virtual

El módulo de administración de repositorios cumple el papel de un registro (*registry*), en el cual se almacena la información necesaria para hacer uso de sus servicios. En el modelo propuesto es necesario especificar la dirección del documento WSDL en el cual se definen los servicios web disponibles, y una dirección URL de la página inicial de la interfaz web del repositorio. Se estima que en un futuro este registro podrá ser extendido para cumplir con las especificaciones de interoperabilidad planteadas por la Arquitectura de Registro y Resolución de Repositorios de Objetos de Contenido (CORDRA), o por el modelo de repositorios distribuidos propuesto por MERLOT.

5.3.3.3. Módulo de administración de objetos de aprendizaje

Este módulo permite construir Assets, SCO o agregaciones de contenidos compatibles con SCORM, que pueden ser almacenados en diferentes repositorios de recursos. Para tener acceso a la funcionalidad de este módulo, el usuario deberá realizar una suscripción a los repositorios que desea tener acceso, y que han sido agregados al registro de repositorios del Aula Virtual por parte de un usuario administrador. La Figura 40 muestra la interfaz que ofrece Aula Virtual para

gestionar las suscripciones de cada usuario a los diferentes repositorios, en la cual se puede apreciar un listado de los repositorios que han sido registrados. Cada usuario (profesor, estudiante o administrador) puede estar registrado en cualquier cantidad de repositorios, y para cada uno de ellos debe contar con una llave única, que es administrada por el Aula Virtual. Si el usuario desea utilizar los servicios del repositorio fuera de Aula Virtual, deberá conocer los datos de registro a dicho repositorio. La opción Ver llaves de la Figura 40 puede ser utilizada para este propósito.

The screenshot shows a web interface titled "Repositorios de contenidos". At the top, there are four navigation tabs: "Estado", "Ver llaves", "Mis recursos", and "Estructuras de contenidos". Below the tabs, the word "Listado" is displayed. The main content area is titled "Repositorios registrados" and contains two entries:

- Repositorio Aula Virtual**: Repositorio por defecto del Aula Virtual. It includes links for "Verificar conexión", "Ingresar al repositorio", "Editar registro", and "Borrar registro".
- Repositorio 2 Aula Virtual**: Repositorio de copia de Aula Virtual. It includes links for "Verificar conexión" and "Registrarse".

Figura 40. Interfaz para la suscripción a los repositorios

El registro en un repositorio es un proceso sencillo, en el cual el usuario debe proporcionar los datos necesarios para invocar el método de suscripción (nombres, email y dirección url). Debido a que es posible que el usuario haya realizado el registro a un repositorio utilizando una herramienta diferente a Aula Virtual, la interfaz permite realizar un nuevo registro o utilizar una clave que ha sido creada con anterioridad (Ver Figura 41).

Repositorios de contenidos
 Estado | Ver llaves | Mis recursos | Estructuras de contenidos

Suscribirse al repositorio
Repositorio 2 Aula Virtual
 Repositorio de copia de Aula Virtual.

Registro por primera vez
 Seleccione esta opción si posee llave para este repositorio.

Dirección de correo
Pagina Web (opcional)

Usuario ya registrado
 Seleccione esta opción si ya posee una llave en el repositorio

Dirección de correo
Llave

Figura 41. Interfaz de registro en un repositorio

Una vez que se ha completado el proceso de suscripción, el usuario puede consultar los objetos de aprendizaje que ha almacenado en los repositorios (Figura 42).

Mis recursos
 Nuevo Recurso | Estructuras de contenidos | Repositorios

Ver recursos almacenados en:

Todos los repositorios
 En los siguientes repositorios:

Repositorio Aula Virtual
 Repositorio por defecto del Aula Virtual
 Repositorio 2 Aula Virtual
 Repositorio de copia de Aula Virtual.

Que contengan el siguiente texto:

Figura 42. Interfaz de búsqueda de objetos de aprendizaje almacenados en diferentes repositorios

Igualmente, el usuario puede agregar nuevos recursos, que se almacenarán directamente en el repositorio de su elección. Estos recursos deberán estar almacenados en forma de archivos comprimidos dentro del espacio que el usuario

dispone dentro de Aula Virtual, y luego de realizar la solicitud de adición, el repositorio se encargará en forma automática de transferir el recurso. La Figura 43 muestra la interfaz para adicionar un nuevo recurso a un repositorio.

Agregar recurso

Agregar al repositorio: Repositorio Aula Virtual ▼

Modelo de referencia: SCORM 1.3 ▼

Tipo de recurso: (Seleccione) ▼

Nombre:

Descripción:

Lenguaje: Spanish (es) ▼

Archivo .ZIP del recurso: ...

El recurso es un archivo .ZIP, no descomprimir.
En este archivo comprimido se deben incluir todos los archivos locales a este recurso.

Archivo inicial:

Algunos recursos poseen varios archivos, se debe especificar el archivo inicial.

Figura 43. Interfaz para adicionar un nuevo recurso a un repositorio

5.3.3.4. Ambiente de Ejecución de SCORM (SCORM Run-Time Environment)

El objetivo principal del ambiente de ejecución de SCORM consiste en proporcionar los medios para la interoperabilidad entre los SCO y los LMS, que comprenden los mecanismos para la ejecución de los objetos de aprendizaje, la comunicación de

estos objetos con el LMS y el almacenamiento de los datos que se generan durante su ejecución⁵⁵. Estos componentes reciben el nombre de Ejecución (Launch), Interfaz de Aplicación de Programas (API) y Modelo de Datos.

Ejecución (Launch)

La ejecución define las relaciones que existen entre los LMS y los materiales contruidos de acuerdo con el modelo de referencia SCORM, y que permiten que los objetos de aprendizaje sean visualizados por el usuario. Adicionalmente, la versión actual de SCORM establece los parámetros y los procedimientos de secuenciación de los objetos de aprendizaje, que permiten determinar cuál es el siguiente objeto a entregar al usuario. En el presente trabajo se realizó una implementación a nivel de prototipo del ambiente de ejecución, con el fin de proporcionar la plataforma tecnológica que permita poner en práctica, evaluar y en un futuro realizar aportes al modelo SCORM. La Figura 44 muestra el despliegue de un curso de ejemplo proporcionado por ADL, que se utiliza para probar las implementaciones del ambiente de ejecución.



Figura 44. Ambiente de ejecución SCORM

⁵⁵ Adaptado de ADL. Shareable Content Object Reference Model. SCORM Run-Time Environment (RTE). Version 1.3.1. 2004.

La parte superior de la Figura 44 muestra las herramientas de navegación definidas en la Especificación de Secuenciación y Navegación (Sequencing and Navigation), y que presentan las opciones Previous (Anterior), Next (Siguiente), Suspend (Suspend), Salir (ExitAll), y la opción para seleccionar un tema específico (Choice). En la parte inferior de la figura se puede apreciar el objeto de aprendizaje en ejecución, que en este caso consiste en un Asset que no establece ningún tipo de comunicación con el LMS.

Interfaz de Aplicación de Programas (API)

Esta interfaz proporciona un conjunto de funciones predefinidas que permiten la comunicación entre el SCO y el LMS, con el fin de informar los eventos que han ocurrido durante la ejecución del SCO. La Figura 45 muestra un SCO que utiliza el API para comunicar al LMS el resultado de una evaluación.



Figura 45. Ejecución de un SCO

La comunicación entre el SCO y el LMS inicia una vez que el recurso ha sido desplegado en el cliente, y generalmente incluye una rutina JavaScript que se invoca desde el evento onLoad de la página HTML (Ver Fragmento de código 27).

Fragmento de código 27. Rutina de inicialización del SCO

```

<body onLoad="loadPage()" onunload="return unloadPage()">
<script language="JavaScript">
function doInitialize()
{
    var api = getAPIHandle();
    if (api == null)
    {
        alert("Unable to locate the LMS's API
Implementation.\nInitialize was not successful.");
        return "false";
    }

    var result = api.Initialize("");

    if (result.toString() != "true")
    {
        var err = ErrorHandler();
    }

    return result.toString();
}
function loadPage()
{
    var result = doInitialize();
    exitPageStatus = false;
    //Demás código de inicialización
}

</script>

```

La función `doInitialize()` realiza la tarea de localizar la implementación del API en el cliente (`getAPIHandle()`), a través de la cual se realiza la comunicación con el LMS. Esta implementación consiste en un Applet de Java, que recibe las solicitudes JavaScript del cliente y las traduce en solicitudes HTTP hacia el LMS. En el presente trabajo de investigación se desarrolló una implementación del API que soporta todos los métodos definidos en el libro SCORM Run-Time Environment, del modelo de referencia SCORM 2004. La Figura 46 muestra los mensajes emitidos por el API, que permiten determinar el estado de su ejecución.

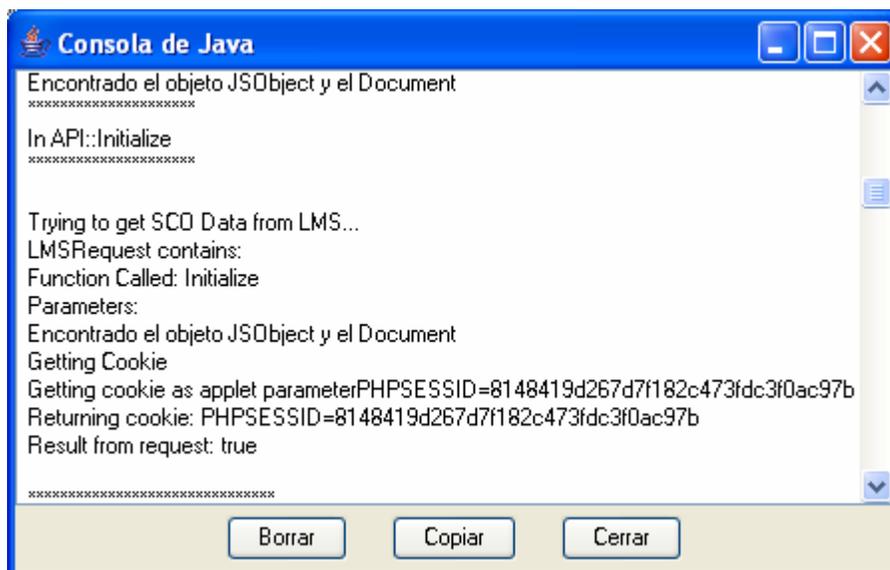


Figura 46. Mensajes emitidos por la implementación del API

Modelo de datos de ejecución

El modelo de datos de ejecución de SCORM (SCORM Run-Time Data Model) define un conjunto de variables que puede ser utilizado para transmitir información del SCO al LMS, y dos funciones para recuperar (get) y establecer (set) sus valores. Este conjunto de variables se basa en la especificación CMI, y debe ser implementada por los ambientes de ejecución para permitir el seguimiento de las actividades que el estudiante desarrolla con el SCO.

En la implementación de modelo de datos desarrollada en el presente trabajo sólo se consideraron aquellos elementos del modelo de datos que determinan el estado de ejecución de un SCO, entre los cuales se destacan los elementos `cmi.comments_from_learner`, `cmi.comments_from_lms`, `cmi.objectives`, `cmi.score` y `cmi.interactions`⁵⁶. La Figura 47 ilustra el proceso de envío de datos del SCO hacia el LMS, por medio del API de SCORM.

⁵⁶ Estos elementos se encuentran descritos en el libro SCORM Run-Time Environment.

```

Consola de Java

In API::SetValue
cmi.score.scaled = 1
*****

Encontrado el objeto JSObject y el Document
Encontrado el objeto JSObject y el Document
Getting Cookie
Getting cookie as applet parameterPHPSESSID=8148419d267d7f182c473fdc3f0ac97b
Returning cookie: PHPSESSID=8148419d267d7f182c473fdc3f0ac97b
Encontrado el objeto JSObject y el Document

*****

Processing done for API::SetValue
*****

*****

In API::SetValue
cmi.success_status = passed
*****

Borrar Copiar Cerrar

```

Figura 47. Envío de datos desde SCO hacia el LMS

Antes de finalizar su ejecución, el SCO reporta toda la información que no ha sido enviada al LMS, y se ejecuta la llamada Commit para cada dato que se transfiere al LMS. Finalmente, el fin de ejecución del SCO debe ser reportado al LMS, y para ello el API invoca el método Finalize(). La Figura 48 muestra el proceso final llevado a cabo por el SCO, a través del API.

```

Consola de Java

Parameters:
Encontrado el objeto JSObject y el Document
Getting Cookie
Getting cookie as applet parameterPHPSESSID=8148419d267d7f182c473fdc3f0ac97b
Returning cookie: PHPSESSID=8148419d267d7f182c473fdc3f0ac97b
Encontrado el objeto JSObject y el Document

*****

Done Processing LMSFinish()
*****

Borrar Copiar Cerrar

```

Figura 48. Fin de la ejecución de un SCO

5.3.4. Componentes de repositorio

Además de la implementación de la interfaz de servicios de metadatos y recursos, en el diseño de la plataforma se definió el desarrollo de una interfaz web para el repositorio de objetos de aprendizaje, que le permite a un usuario tener acceso a los servicios que ofrece, sin necesidad de utilizar un LMS, LCMS o cualquier otra aplicación como intermediarios. La Figura 49 muestra la página de entrada a la interfaz web del repositorio implementado.

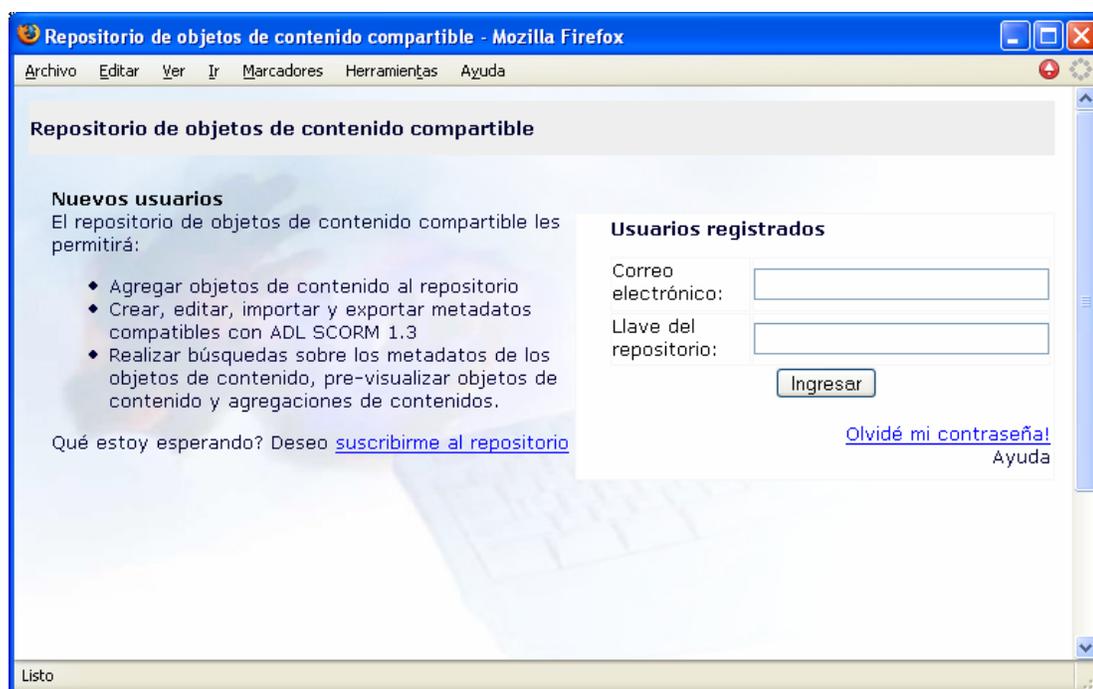


Figura 49. Página de entrada a la interfaz web del repositorio

Para ingresar al repositorio, el usuario deberá especificar su correo electrónico y la llave de usuario que le fue asignada en el momento de su registro. Si el usuario no se encuentra registrado, puede realizar el registro de forma gratuita, para lo cual se deberán especificar algunos datos generales como el nombre, el correo electrónico y una página de contacto. La Figura 50 muestra la interfaz de suscripción al repositorio.

Suscribirse al repositorio (Regresar)

Para utilizar los servicios del repositorio se debe contar con una clave personal (Personal Key), que identifica de manera única al usuario y le permite tener acceso a los servicios de inclusión de nuevos recursos de aprendizaje y a la modificación de su información descriptiva (metadatos). Esta clave se expide de forma gratuita para cualquiera que utilice el repositorio, sea una persona, un LMS o un repositorio.

Para expedir la clave personal deberá especificar los siguientes datos:

Nombre

Correo Electronico

Pagina Web (opcional)

Figura 50. Interfaz de suscripción al repositorio

Una vez realizado el registro en el repositorio, el usuario podrá tener acceso a los servicios que ofrece. A continuación se presentarán las interfaces para la gestión de recursos, la carga y descarga de objetos de aprendizaje y metadatos, y la interfaz para la búsqueda de recursos. Finalmente se realizará una descripción de una herramienta para visualizar objetos de aprendizaje, que se encuentra integrada dentro del repositorio.

5.3.4.1. Interfaz de gestión de recursos

La interfaz de gestión de recursos ofrece un listado de los objetos de aprendizaje que el usuario ha agregado al repositorio, y para cada uno de ellos presenta las opciones de visualizar, descargar, editar sus datos generales y sus metadatos (Ver Figura 51). Es importante destacar que la interfaz no implementa nuevamente la lógica para gestionar los recursos, para ello hace uso de los diferentes servicios que ofrece el repositorio, en forma de llamadas a los métodos de la interfaz de servicios web (para esta interfaz se utiliza una llamada al método para obtener los recursos propios del usuario).

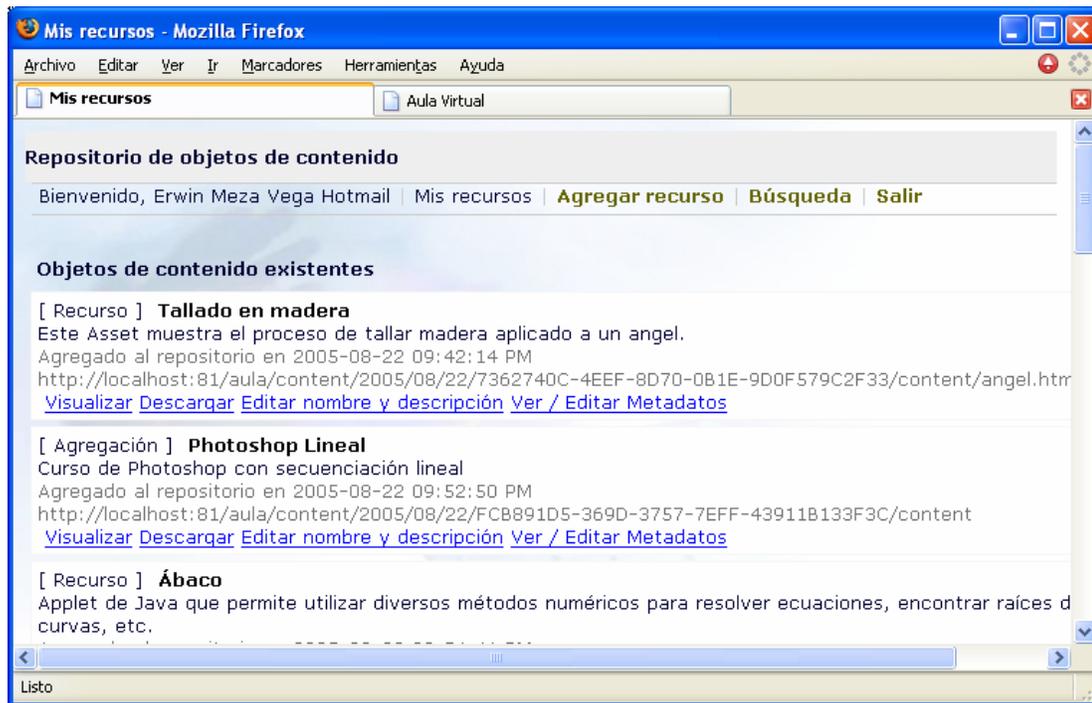


Figura 51. Interfaz de gestión de recursos

5.3.4.2. Servicios de carga de objetos de aprendizaje

Esta interfaz permite adicionar nuevos objetos de aprendizaje dentro del repositorio por medio de un formulario sencillo, en el cual el usuario debe especificar tipo del recurso, seleccionando un modelo de referencia soportado por el repositorio (que se obtiene mediante una llamada al método `getReferenceModelList`) y el componente dentro de este modelo (SCO, Asset, agregación de contenidos, obtenidos por medio de una llamada al método `getComponentList`), un nombre y una descripción general del recurso, el lenguaje, el archivo comprimido en el cual se encuentran todos los medios digitales que lo conforman, y el nombre del archivo que deberá ser lanzado por el LMS al tener acceso al recurso (Ver Figura 52).

Figura 52. Interfaz de adición de un nuevo recurso

Luego de especificar todos los datos, y tras presionar el botón aceptar, la interfaz hace una llamada al método del repositorio para adicionar un nuevo recurso (`addResource`), al cual le pasa como parámetro los datos proporcionados por el usuario. El método `addResource` del repositorio se encarga de tomar el recurso (almacenado en un directorio temporal) y de generar una llave única para el, por medio de la rutina MD5⁵⁷. Esta llave se compara con las que ya existen en el repositorio, y en el caso de coincidir se informa al usuario que este recurso ya ha sido adicionado al repositorio. Si el recurso no existe dentro del repositorio, se agrega y se crea su registro de metadatos, en el cual se incluyen de forma automática aquellos elementos que son requeridos dentro del esquema LOM (para observar cuales elementos se crean para un SCO, Asset o agregación de contenidos, ver Anexo 1), y finalmente se informa al usuario del éxito de la operación.

⁵⁷ Al hacer uso de esta rutina se obtiene una cadena de caracteres de 31 bits, que representa la “huella digital” de un archivo.

5.3.4.3. Servicios de edición, carga y descarga de metadatos

El objetivo de esta interfaz consiste en ofrecer los mecanismos para modificar los metadatos de un objeto de aprendizaje, y permitir su carga o descarga en forma de documentos XML. A diferencia de la mayoría de herramientas para la edición de metadatos que existen en la actualidad, la implementación desarrollada sigue las recomendaciones del estándar LOM, para permitir más de una instancia de cada elemento XML que conforma el registro de metadatos de un objeto de aprendizaje. Esto es especialmente útil para definir metadatos en múltiples lenguajes, en los cuales se tienen varias instancias del elemento título, cada una en diferentes lenguajes, por ejemplo. La Figura 53 muestra la interfaz de edición de metadatos, en la cual se observan varias instancias de un mismo elemento.

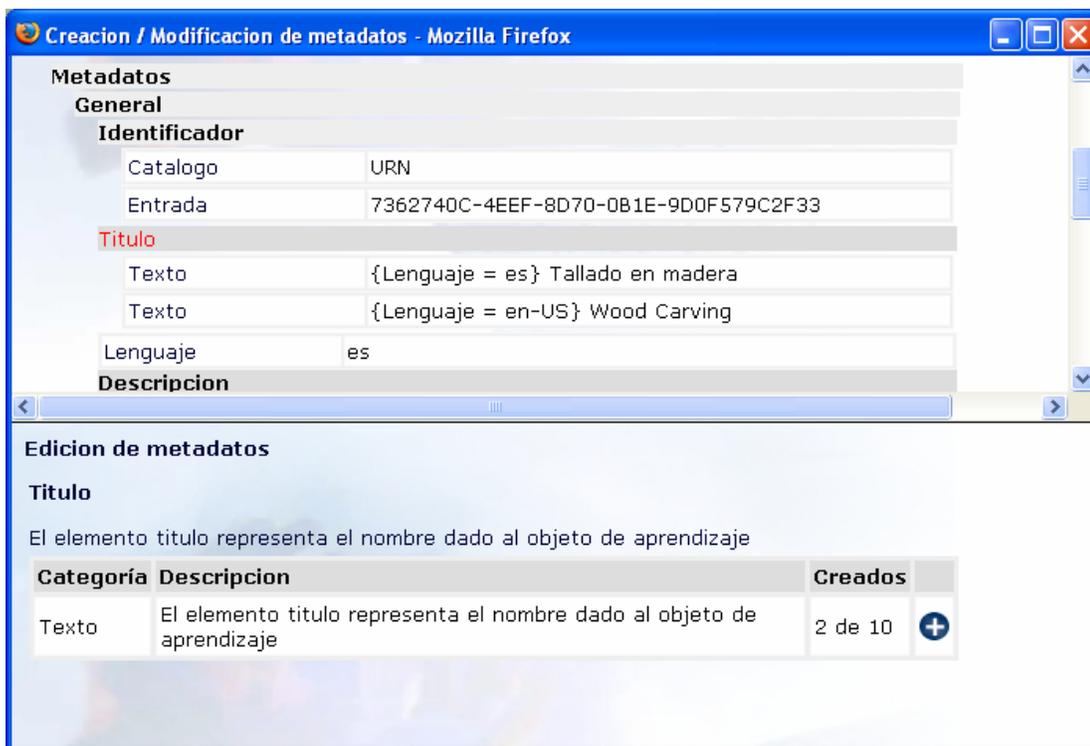


Figura 53. Interfaz para la edición de metadatos de un objeto de aprendizaje

Al seleccionar un elemento en la parte superior de la interfaz (en este caso el elemento Título), se despliegan los elementos que contiene, de acuerdo con el estándar de metadatos definido para el objeto. De igual forma, al seleccionar un

elemento “hoja” (que no contiene otros elementos), se despliega un formulario en el cual es posible introducir la información.

5.3.4.4. Servicios de búsqueda de objetos de aprendizaje

Además de los servicios búsqueda implementados como parte de la interfaz de interoperabilidad, el repositorio ofrece al usuario un mecanismo sencillo para buscar y localizar objetos de aprendizaje. Este mecanismo consiste en un formulario simple que permite localizar objetos de aprendizaje de acuerdo con un patrón de búsqueda, la fecha de adición de los recursos dentro del repositorio y el tipo de recurso que se desea localiza. La Figura 54 muestra la interfaz de búsqueda (avanzada) disponible dentro del repositorio.

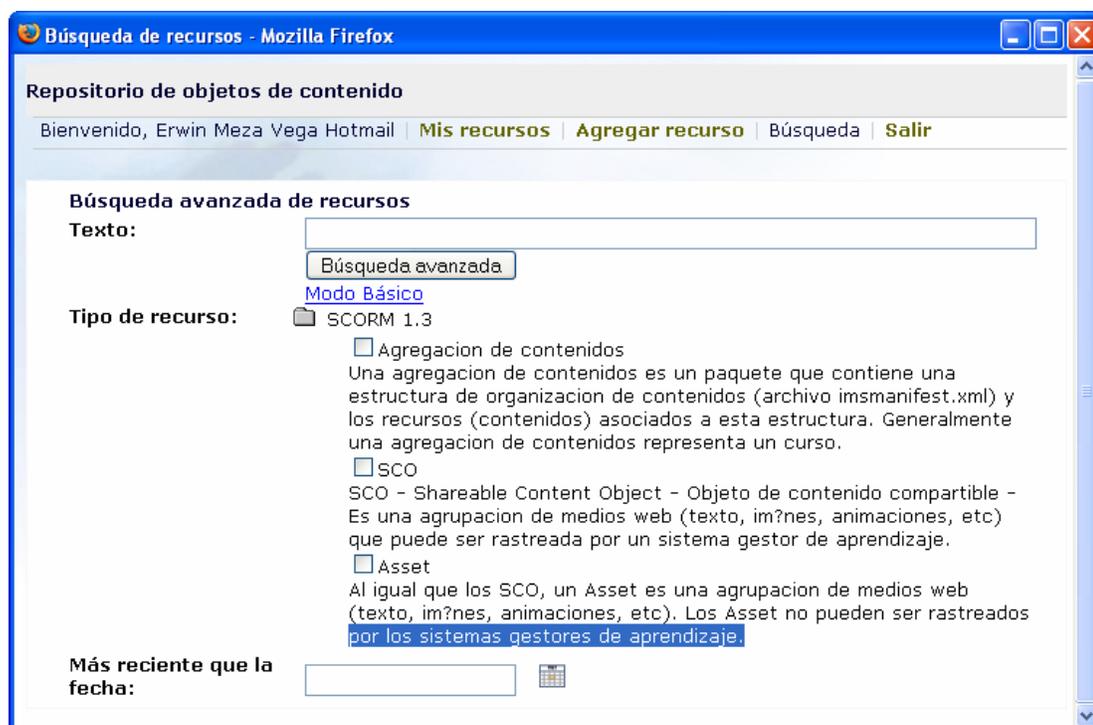


Figura 54. Interfaz de búsqueda de objetos de aprendizaje

Una vez que el usuario ha especificado los parámetros de búsqueda, se confronta el patrón de texto con los valores presentes en los metadatos de todos los objetos de aprendizaje presentes en el repositorio, y se presenta un listado con los recursos

localizados. A partir de este listado el usuario tiene la posibilidad de visualizar el objeto de aprendizaje, descargarlo, ver sus metadatos, y si el recurso es de su propiedad, podrá editar sus metadatos.

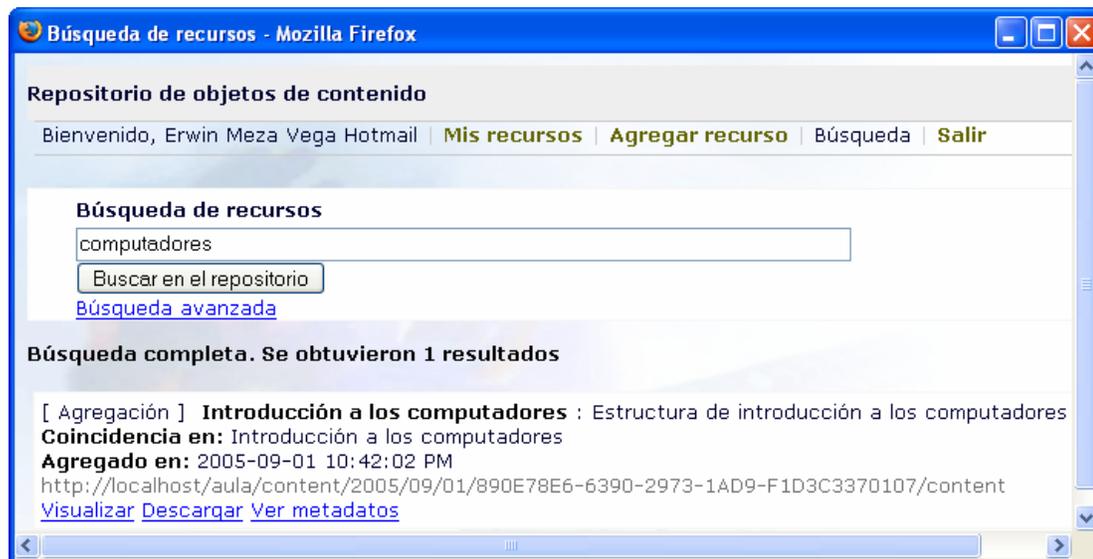


Figura 55. Resultados de la búsqueda de recursos

5.3.4.5. Herramienta para visualizar agregaciones de contenido SCORM

Este componente de la interfaz web que ofrece el repositorio permite visualizar paquetes de contenidos compatibles con SCORM, sin necesidad de cargar completamente el ambiente de ejecución. Esto es especialmente útil durante la búsqueda de agregaciones de contenidos, en las cuales el usuario desea obtener una "vista preliminar", para poder determinar cuál es el recurso que mejor se ajusta a sus necesidades. El componente fue desarrollado en forma de un visor, que despliega la estructura de la agregación de contenidos, y en la cual el usuario puede seleccionar el tema que desea observar. La Figura 56 muestra la interfaz que presenta la herramienta de visualización de agregaciones de contenidos, en la cual se aprecia en la parte izquierda el árbol de temas que la conforman, el tema seleccionado (en rojo), un menú en la parte superior que permite ocultar el árbol de temas, y la opción para terminar la visualización.



Figura 56. Herramienta para visualizar agregaciones de contenidos SCORM

En esta interfaz el usuario podrá revisar libremente la estructura de contenidos, al hacer click sobre cualquier tema de la estructura desplegada en la parte izquierda del visor. Sin embargo, debido a que el visor no implementa el ambiente de ejecución de SCORM, no es posible observar el comportamiento de las reglas de secuenciación que se hayan definido dentro de la organización de contenidos.

CONCLUSIONES Y RECOMENDACIONES

El desarrollo de este trabajo permitió estudiar la evolución de los diferentes estándares, las especificaciones y los modelos de referencia que en la actualidad determinan la construcción de materiales que se utilizan para apoyar experiencias educativas en línea. A pesar que SCORM está siendo adoptado como un estándar *de facto* para la educación en línea, es necesario que la comunidad académica se involucre en el diseño y el desarrollo de proyectos que hagan uso de este modelo, con el fin de contribuir a su mejoramiento o evolución, y dado el caso en la creación de un nuevo modelo de referencia, si este no cumple con las expectativas desde el punto de vista tecnológico y pedagógico.

La creación de materiales educativos es una tarea compleja, que requiere la conformación de grupos de estudio y equipos de trabajo, que por medio de experiencias en diferentes contextos educativos, puedan determinar los mejores escenarios para la reutilización, la creación y la construcción de objetos de aprendizaje. A partir de estas experiencias se deberá generar una serie de guías de trabajo, documentos de estilo, manuales de procedimientos, que podrán ser utilizados por otros grupos en diferentes instituciones del país.

El protocolo de intercambio de objetos de aprendizaje desarrollado en el presente trabajo ofrece las primitivas básicas para la construcción y la gestión distribuida de materiales basados en objetos de aprendizaje. Algunas mejoras que se deberán ser consideradas en el futuro son las políticas de seguridad, la existencia de modelos de referencia con múltiples esquemas de metadatos, así como el manejo de esquemas de metadatos que no se representan como documentos XML. Igualmente se deberá profundizar en cuanto a la posibilidad de integrar mecanismos de edición colaborativa de contenidos y metadatos.

La implementación de la plataforma diseñada ofreció la posibilidad de estudiar la mayoría de aspectos relacionados con el ciclo de vida de los materiales educativos, que incluyen la creación, la construcción, el almacenamiento, la catalogación, la búsqueda, localización y visualización, la edición y la reutilización. De igual forma,

los servicios ofrecidos por la nueva versión de Aula Virtual y el Repositorio de Objetos de Contenido podrán ser utilizados por diferentes grupos de trabajo e instituciones, para consolidar una red de estudio y colaboración en torno al uso de materiales educativos en experiencias de aprendizaje.

Igualmente los ambientes virtuales de aprendizaje desarrollados con anterioridad a este proyecto o en el futuro podrán hacer uso de los servicios desarrollados, en la medida que se realice una implementación de las interfaces de acceso a la plataforma de intercambio de contenidos. Con ello se logrará potenciar estas herramientas, y no se tendrá la necesidad de importar y adaptar soluciones pre-configuradas, que ocultan la mayoría de detalles tecnológicos y pedagógicos de su implementación.

En la actualidad el modelo de referencia SCORM plantea la creación de objetos de aprendizaje que son experimentados por un solo estudiante, durante una sesión de trabajo. Luego de realizar el estudio de SCORM desde el punto de vista pedagógico, es necesario plantear nuevas estrategias para lograr una mejor integración de los materiales dentro de las experiencias educativas, como el desarrollo de actividades basadas en roles e interacciones entre los participantes, y actividades de trabajo colaborativo. En esta dirección se puede hacer uso de la especificación Learning Design (IMS LD), desarrollada por IMS. Se estima que en un futuro ADL integre IMS LD dentro de las versiones futuras de SCORM.

También es necesario considerar la necesidad de desarrollar patrones de diseño y guías de trabajo para la implementación de herramientas utilizadas en la educación en línea, tales como las aplicaciones de creación y manejo de materiales, los ambientes virtuales de aprendizaje y los repositorios digitales. A partir de estas guías de trabajo se podrán establecer nuevos mecanismos para la interoperabilidad entre repositorios, la comunicación entre los objetos de aprendizaje y los ambientes virtuales, y la distribución de materiales. Esta dinámica de cambio permitirá generar nuevos proyectos de investigación, que reforzarán las bases conceptuales de la educación en línea.

BIBLIOGRAFÍA

- ADL. Advanced Distributed Learning. Disponible en: <http://www.adlnet.org>. 2000.
- ADL. Shareable Content Object Reference Model. SCORM Content Aggregation Model (CAM). Version 1.3.1. 2004. Disponible en línea en <http://www.adlnet.org>
- ADL. Shareable Content Object Reference Model. SCORM Run-Time Environment (RTE). Version 1.3.1. 2004. Disponible en línea en <http://www.adlnet.org>
- ADL. Shareable Content Object Reference Model. SCORM Sequencing and Navigation. (SN) Version 1.3.1. 2004. Disponible en línea en <http://www.adlnet.org>
- AGUILAR, Esperanza. CORREDOR, Martha y otros. Aula Virtual. Una alternative en educación superior. Ediciones Universidad Industrial de Santander. Primera Edición, 2003.
- ALDERMAN, F Lee. BARRIT Jr. Chuck. Creating a Reusable Learning Objects Strategy: Leveraging Information and Learning in a Knowledge Economy. Wiley. 2004.
- ALVARADO, Vigny. ELIZONDO, Fernando. Estructura tecnológica para una educación a distancia de calidad. Universidad Estatal a Distancia, Costa Rica. 2002.
- AICC. Aviation Industry CBT Committee. Disponible en línea en <http://www.aicc.org/pages/aicc3.htm>. 1998.
- CERAMI, Ethan. Web Service Essentials. Distributed Applications with XML-RPC, SOAP, UDDI & WSDL. O'Reilly Associates. Primera edición. 2002.
- Cisco Systems, Inc. Reusable Learning Object Strategy. Definition, creation process, and guidelines for building. Version 3.1. 2000.
- Cisco Systems, Inc. Reusable Learning Object Strategy: Designing and Developing Learning Objects for Multiple Learning Approaches. Version 4.5. 2003.
- EduTools. Información actualizada sobre las últimas versiones de las herramientas utilizadas en E-Learning. <http://www.edutools.info/course/productinfo/index.jsp>. Última revisión: septiembre de 2005.
- Google Web APIs Reference. <http://www.google.com/apis/reference.html>. 2004.
- HATHALA, Marek. RICHARDS, Griff y otros. The EduSource Communication Language: Implementing Open Network for Learning Repositories and Services. School of Interactive Arts and Technology, Simon Fraser University 2003.

HEERY, Rachel, ANDERSON, Sheila. Digital Repositories Review. Arts and Humanities Data Service, UKOLN. 2005.

IDC. A New eLearning Market Segment Emerges. Artículo disponible en <http://www.internettime.com/Learning/lcms/IDCLCMSWhitePaper.pdf>. 2001. Última revisión: septiembre de 2005.

IEEE P1484.12.1-2002 Learning Object Metadata Working Group http://ltsc.ieee.org/wg12/s_p.html. 2000.

IEEE 1484.12.3 Draft Standard for Extensible Markup Language (XML) Binding for Learning Object Metadata Data Model.

IMS Digital Repositories Interoperability. Versión 1.0. IMS Global Inc. Disponible en línea en http://www.imsglobal.org/digitalrepositories/driv1p0/imsdri_infov1p0.htm. 2003.

Information Retrieval (Z39.50): Application Service Definition and Protocol Specification. ANSI/NISO. ISSN: 1041-5653. 2003.

KRUCHTEN, Philippe. The Rational Unified Process An Introduction, ,Second Edition. Addison Wesley, 2000.

Learning Systems Architecture Lab, LSAL. SCORM Best Practices for Content Developers. Carnegie Mellon University. Primera edición, 2003. Disponible en línea en <http://www.lsal.cmu.edu/lsal/expertise/projects/developersguide/>

Learning Technology Standards Committee. Disponible en: <http://ltsc.ieee.org>. 2000.

Learning technology standards committee: Learning Object Metadata. *LOM working draft v4.1* Disponible en línea en <http://ltsc.ieee.org/doc/wg12/LOMv4.1.htm>. 2000.

MASIE Center. Making sense of Learning Specifications and Standards: A Decisions Maker's Guide to Their Adoption. Industry Report. Versión Noviembre de 2003. <http://www.masie.com>

NICHANI, Maish. LCMS = LMS + CMS [RLOs]. How does this affect the learner? The instructional designer? 2001. Disponible en línea en <http://www.elearningpost.com/elthemes/lcms.asp>. Última revisión: septiembre de 2005.

PASINI, Nina. REHAK, Daniel. A Process Model for Applying Standards in Content Development. Carnegie Mellon University. Disponible en línea en <http://lsal.cmu.edu> 2003.

PAULSEN, Morten Flate. Online Education Systems: Discussion and Definition of Terms. NKI Distance Education. 2002.

REHAK, Daniel. DODDS, Phillip. LANNOM, Laurence. A Model and Infrastructure for Federated Learning Content Repositories. 2005.

RENGARAJAN, Raghavan. LCMS and LMS. Taking advantage of Tight Integration. Click2Learn, inc. 2001.

SAMPEDRO, A. SARRIEGO, R. Procesos implicados en el desarrollo de Materiales Didácticos reutilizables para el fomento de la Cultura Científica y Tecnológica. Universidad de Oviedo. Disponible en línea en <http://www10.uniovi.es/spi>. 2004.

SOUTH, Joseph. MONSON, David. A University-wide System for Creating, Capturing, and Delivering Learning Objects. Disponible en línea en <http://www.reusability.org/read/chapters/south.doc>. 2001.

STAPLES, Thornton. WAYLAND, Ross. PAYETTE, Sandra. The Fedora Project: An Open Source Digital Object Management System.D - Lib Magazine 9/4 Abril 2003. Disponible en línea en <http://www.dlib.org/dlib/april03/staples/04staples.html>

URIARTE, Jesús. GÓMEZ, Manuel. E-Learning: perspectivas de las plataformas que lo soportan. 2001.

WILEY, David A. Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. Disponible en línea en <http://www.reusability.org/read/chapters/wiley.doc>. 2001.

WILEY, David. PADRON, Sandie y Otros. Using O2 to Overcome Learning Objects Limitations. Publicado en Learning Objects 2003 Symposium: Lessons Learned, Questions Asked Honolulu, Hawaii. Disponible en línea en <http://www.cs.kuleuven.ac.be/~erikd/PRES/2003/LO2003/Wiley.pdf>. 2003.

ZAPATA, Miguel. Sistemas de gestión del aprendizaje – Plataformas de teleformación. 2003.

ZAPATA, Miguel. Formación abierta y a distancia a través de redes digitales: Modelos de redes de aprendizaje RED, num. 1 pag 6 a 9. URL: www.um.es/ead/red/ 2002.

ANEXOS

(Espacio en blanco intencional)

Anexo 1. Esquema de metadatos LOM

A continuación se presenta una tabla con la clasificación de los metadatos LOM1.0, propuesta por el Learning Technology Standards Comitee de la Universidad Carnegie Mellon (LSAL), y que es utilizada por el modelo de referencia SCORM para la descripción de los objetos de aprendizaje (Assets, SCO, y agregaciones de contenido). Las filas resaltadas corresponden a las categorías del esquema de metadatos que se utilizan con mayor frecuencia para describir un objeto de aprendizaje, de acuerdo con LSAL (general, ciclo de vida, meta-metadatos, técnico, educacional, derechos y clasificación).

(Espacio en blanco intencional para conservar el formato del documento)

Número	Nombre del elemento	Elemento XML	Descripción
	Metadatos	lom	El elemento lom encapsula todos los metadatos
1	General	general	La categoría general agrupa la información que describe el recurso como una unidad
1.1	Identificador	identifier	El identificador representa un mecanismo para asignar una etiqueta global única que identifica el recurso SCORM
1.1.1	Catalogo	catalog	El catalogo representa el nombre del esquema de identificación de la entrada. Ejemplo: URN (Uniform Resource Name - Nombre de Recurso Uniforme), URI (Universal Resource Identifier - Identificador Universal de Recurso)
1.1.2	Entrada	entry	Este elemento representa el valor del identificador dentro del esquema de identificación escogido
1.2	Título	title	El elemento titulo representa el nombre dado al objeto de aprendizaje
1.3	Lenguaje	language	El elemento lenguaje representa el o los lenguajes que se usan dentro del recurso SCORM para comunicarse con el usuario. Algunos ejemplos son es (Español), en (Ingles), en-US (Ingles de Estados Unidos)
1.4	Descripción	description	Descripción textual del recurso SCORM
1.5	Palabra clave	keyword	Se debe utilizar para definir las palabras o frases que describen el objeto de aprendizaje
1.6	Cobertura	coverage	El elemento cobertura se utiliza para describir la fecha, la cultura o la ubicación geográfica para los cuales aplica el recurso SCORM que se esta describiendo
1.7	Estructura	structure	El elemento estructura describe la organización del recurso SCORM

1.8	Nivel de agregación	aggregationLevel	Este elemento describe la granularidad funcional del objeto de aprendizaje. 1). nivel mas bajo de granularidad, se utiliza para fragmentos multimedia (un fragmento de texto, una imagen, video, etc.). 2) Describe una colección de objetos de nivel 1, es decir un conjunto de páginas, archivos, etc. Equivalente a una lección. 3) Agrupación de elementos de nivel 2, equivalente a un curso. 4). Colección de recursos de nivel 3, puede representar una agregación de cursos tendiente a una certificación
2	Ciclo de vida	lifeCycle	Esta categoría agrupa las características relacionadas con la historia y estado actual del recurso SCORM
2.1	Versión	version	Este elemento describe la edición del recurso SCORM
2.2	Estado	status	Este elemento describe el estado o condición del recurso SCORM
2.3	Contribución	contribute	Este elemento se usa para describir aquellas entidades que han contribuido al estado del recurso SCORM durante su ciclo de vida
2.3.1	Rol	role	Este elemento define la clase de contribución realizada
2.3.2	Entidad	entity	Este elemento identifica las personas o entidades que han contribuido al desarrollo del recurso SCORM
2.3.3	Fecha	date	Este elemento describe la fecha de la contribución
3	Meta - Metadatos	metaMetadata	Esta categoría proporciona los elementos que describen los metadatos (no el recurso SCORM)
3.1	Identificador	identifier	El identificador representa un mecanismo para asignar una etiqueta global única que identifica el

			registro de metadatos actual
3.1.1	Catalogo	catalog	El catalogo representa el nombre del esquema de identificación de la entrada. Ejemplo: URN (Uniform Resource Name - Nombre de Recurso Uniforme), URI (Universal Resource Identifier - Identificador Universal de Recurso)
3.1.2	Entrada	entry	Este elemento representa el valor del identificador dentro del esquema de identificación escogido
3.2	Contribución	contribute	Este elemento se usa para describir aquellas entidades que han contribuido al estado de los metadatos
3.2.1	Rol	role	Este elemento define la clase de contribución realizada
3.2.2	Entidad	entity	Este elemento identifica las personas o entidades que han contribuido al desarrollo de los metadatos
3.2.3	Fecha	date	Este elemento describe la fecha de la contribución
3.3	Esquema de metadatos	metadataSchema	Este elemento representa el nombre y la versión de las especificaciones que se utilizaron para crear los metadatos. SCORM 1.3 exige que se incluyan por defecto dos elementos: LOMv1.0 y SCORM_CAM_v1.3
3.4	Lenguaje	language	Este elemento representa el lenguaje utilizado para describir los metadatos
4	Técnico	technical	Esta categoría describe todas las características técnicas y requerimientos del recurso SCORM
4.1	Formato	format	Este elemento representa el tipo de datos técnico de TODOS los recursos usados en la construcción del recurso SCORM
4.2	Tamaño	size	Este elemento representa el tamaño del recurso

			SCORM en bytes
4.3	Localización	location	Este elemento es un string que especifica la localización del recurso SCORM descrito por los metadatos
4.4	Requerimiento	requirement	Este elemento expresa los aspectos técnicos necesarios para utilizar el recurso SCORM
4.4.1	Compuesto	orComposite	Este elemento representa un requerimiento simple
4.4.1.1	Tipo	type	Este elemento representa la tecnología que se requiere para utilizar el recurso SCORM
4.4.1.2	Nombre	name	Este elemento representa la herramienta tecnológica requerida para utilizar el recurso SCORM
4.4.1.3	Versión mínima	minimumVersion	Este elemento representa la mínima versión de la herramienta tecnológica para usar el recurso SCORM
4.4.1.4	Versión máxima	maximumVersion	Este elemento representa la máxima versión (si es necesario) de la herramienta tecnológica para usar el recurso SCORM
4.5	Observaciones de instalación	installationRemarks	Este elemento contiene las instrucciones adicionales que se requieren para instalar y utilizar el recurso SCORM
4.6	Otros requisitos de instalación	otherPlatformRequirements	Este elemento se utiliza para representar la información acerca de otros requerimientos de software o hardware para utilizar el recurso SCORM
4.7	Duración	duration	Este elemento representa el tiempo que se requiere para visualizar el recurso SCORM
5	Educacional	educational	Esta categoría describe las características educacionales o pedagógicas del recurso SCORM
5.1	Tipo de	interactivityType	Este elemento representa la estrategia de

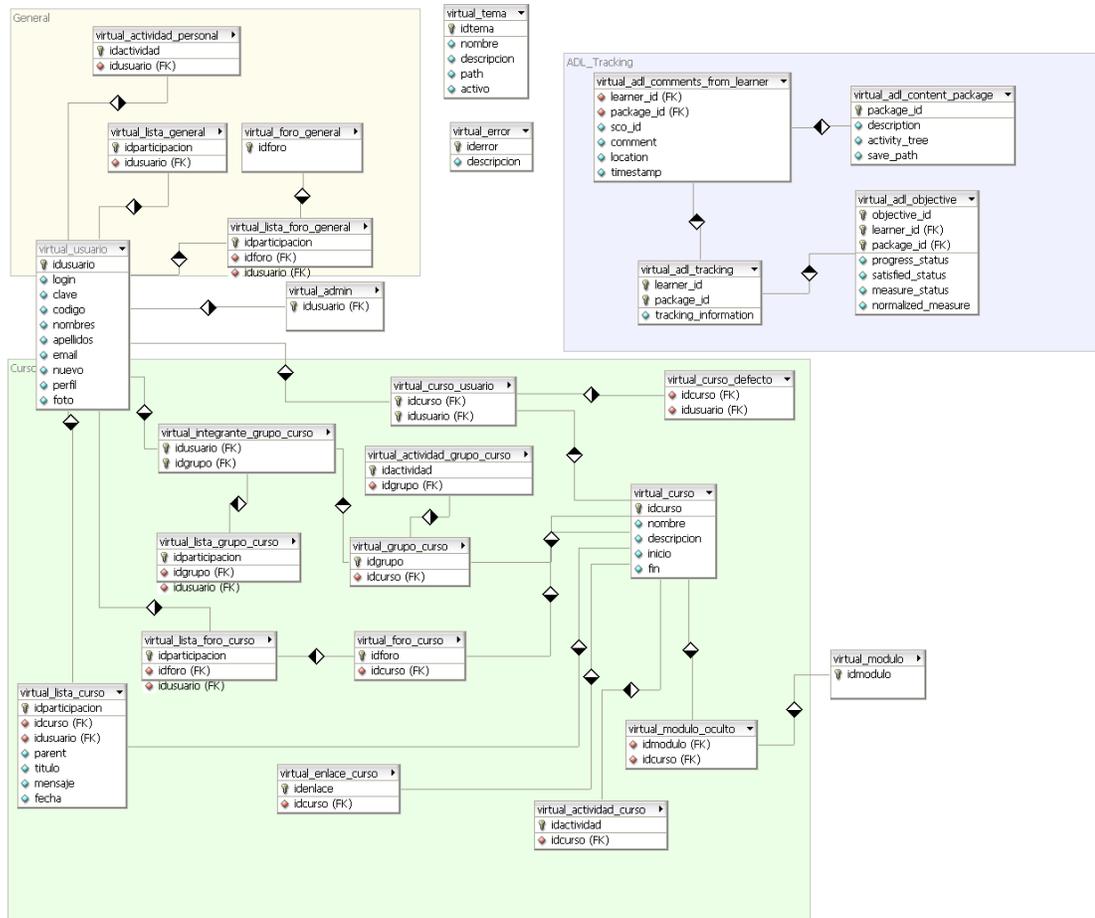
	interactividad		aprendizaje que soporta el recurso SCORM
5.2	Tipo de recurso de aprendizaje	learningResourceType	Este elemento especifica la clase específica del recurso SCORM
5.3	Nivel de Interactividad	interactivityLevel	Este elemento representa el grado de interactividad que caracteriza el recurso SCORM
5.4	Densidad Semántica	semanticDensity	Este elemento representa el grado de densidad semántica del recurso SCORM
5.5	Rol del usuario final	intendedEndUserRole	Este elemento especifica los principales usuarios para los cuales fue diseñado el recurso SCORM
5.6	Contexto	context	Este elemento representa el contexto dentro del cual se debe utilizar el recurso SCORM
5.7	Rango de edad típico	typicalAgeRange	Este elemento representa la edad o rango de edades de los usuarios potenciales de este recurso SCORM.
5.8	Dificultad	difficulty	Este elemento representa la dificultad del recurso SCORM en términos de su utilización
5.9	Tiempo promedio de aprendizaje	typicalLearnigTime	Este elemento representa el tiempo promedio que se utilizaría para revisar este recurso
5.10	Descripción	description	Este elemento se utiliza para describir como se debería utilizar el recurso SCORM
5.11	Lenguaje	language	Este elemento representa el lenguaje que se utiliza en el recurso
6	Derechos	rights	Esta categoría abarca los derechos de autor y otras restricciones del recurso SCORM
6.1	Costo	cost	Este elemento representa si el recurso SCORM tiene costo o no
6.2	Derechos de autor y otras restricciones	copyrightAndOtherRestrictions	Este elemento representa si el recurso SCORM tiene alguna restricción para su uso
6.3	Descripción	description	Este elemento representa la descripción de las condiciones de uso del componente SCORM
7	Relación	relation	Esta categoría define las relaciones entre recurso SCORM, si existen.

7.1	Clase	kind	Este elemento describe la naturaleza de la relación entre el recurso actual y el recurso referenciado
7.2	Recurso	resource	Este elemento describe el recurso SCORM que se referencia en esta relación
7.2.1	Identificador	identifier	El identificador representa un mecanismo para asignar una etiqueta global única que identifica el recurso SCORM
7.2.1.1	Catalogo	catalog	El catalogo representa el nombre del esquema de identificación de la entrada. Ejemplo: URN (Uniform Resource Name - Nombre de Recurso Uniforme), URI (Universal Resource Identifier - Identificador Universal de Recurso)
7.2.1.2	Entrada	entry	Este elemento representa el valor del identificador dentro del esquema de identificación escogido
7.2.2	Descripción	description	Este elemento describe el recurso SCORM al que se hace referencia
8	Anotación	annotation	Esta categoría proporciona comentarios del uso educativo del recurso SCORM e información de quien creó estos comentarios.
8.1	Entidad	entity	Este elemento identifica las personas o entidades que crearon la anotación
8.2	Fecha	date	Este elemento describe la fecha en la que se realizo la anotación.
8.3	Descripción	description	Este elemento representa el contenido de la anotación.
9	Clasificación	classification	Esta categoría describe la clasificación del recurso SCORM dentro de algún sistema de clasificación, tal como un catalogo bibliográfico.
9.1	Propósito	purpose	Este elemento describe el propósito de la clasificación asignada al recurso SCORM

9.2	Taxonomía	taxonPath	Este elemento describe el camino taxonómico dentro de un sistema de clasificación. Cada nivel del elemento Taxon representa un refinamiento en la definición del nivel Taxon anterior.
9.2.1	Fuente	source	Este elemento describe el nombre del sistema de clasificación.
9.2.2	Taxon	taxon	Este elemento describe un término particular dentro de una taxonomía. Un Taxon es un nodo que contiene una etiqueta y un identificador.
9.2.2.1	Id	id	Este elemento describe el identificador del Taxon.
9.2.2.2	Entrada	Entry	Este elemento contiene la etiqueta textual del Taxon
9.3	Descripción	description	Este elemento contiene la descripción del recurso SCORM relativo al propósito establecido.
9.4	Palabra clave	keyword	Este elemento contiene las palabras claves relativas al propósito establecido

Anexo 2. Modelo de datos de Aula Virtual 2.0

A continuación se presenta el modelo de datos de Aula Virtual 2.0, implementado en una base de datos MySQL⁵⁸.



Modelo 1. Modelo de datos de Aula Virtual 2.0

A continuación se presenta el diccionario de datos de Aula Virtual 2.0, que describe la estructura de las tablas y los campos.

⁵⁸ Sin embargo, este modelo puede ser implementado en cualquier motor de bases de datos, gracias a la estructura modular del repositorio.

virtual_actividad_curso

Esta tabla se utiliza para almacenar las actividades de curso.

Columna	Tipo	Llave	Descripción
idactividad	BIGINT(20)	PK	Identificador de la actividad
idcurso	BIGINT(20)		Identificador del curso
inicio	BIGINT(20)		Fecha de inicio (timestamp ⁵⁹)
fin	BIGINT(20)		Fecha de finalización (timestamp)
descripcion	TEXT		Descripción de la actividad

virtual_actividad_grupo_curso

Esta tabla almacena las actividades para los grupos creados dentro de un curso.

Columna	Tipo	Llave	Descripción
idactividad	BIGINT(20)	PK	Identificador de la actividad
grupo	BIGINT(20)		Identificador del grupo
inicio	BIGINT(20)		Fecha de inicio (timestamp)
fin	BIGINT(20)		Fecha de finalización (timestamp)
descripcion	TEXT		Descripción de la actividad

virtual_archivo_taller

Esta tabla almacena la información de los archivos que se envían como solución a un taller

Columna	Tipo	Llave	Descripción
---------	------	-------	-------------

⁵⁹ Un *timestamp* es un entero largo que indica la cantidad de segundos transcurridos desde el 1 de Enero de 1970. Este número puede ser transformado en fechas estándares. La ventaja de almacenar fechas en este formato consiste en que no dependen del idioma o la representación interna de la base de datos.

idarchivo	BIGINT(20)	PK	Identificador del archivo
idtaller	BIGINT(20)		Identificador del taller
idusuario	BIGINT(20)		Identificador del usuario
nombre	TEXT		Nombre del archivo
filename	TEXT		Nombre físico del archivo
descripcion	TEXT		Descripción del archivo
fecha	BIGINT(20)		Fecha de adición

virtual_curso

Esta tabla almacena la información de los cursos ofrecidos en Aula Virtual

Columna	Tipo	Llave	Descripción
idcurso	BIGINT(20)	PK	Identificador del curso
nombre	TEXT		Nombre del curso
descripcion	TEXT		Descripción
inicio	BIGINT(20)		Fecha de inicio
fin	BIGINT(20)		Fecha de finalización
teachername	VARCHAR(50)		Alias del rol profesor
teachernames	VARCHAR(50)		Alias del rol profesor, en plural
studentname	VARCHAR(50)		Alias del rol estudiante
studentnames	VARCHAR(50)		Alias del rol estudiante, en plural

virtual_curso_defecto

Esta tabla almacena la información del curso por defecto al cual ingresará un usuario al iniciar sesión en Aula Virtual.

Columna	Tipo	Llave	Descripción
idusuario	BIGINT(20)		Identificador del usuario
idcurso	BIGINT(20)		Identificador del curso

virtual_curso_usuario

Esta tabla permite almacenar la información de los cursos asignados en los cuales se encuentra inscrito un usuario..

Columna	Tipo	Llave	Descripción
idcurso	BIGINT(20)	PK	Identificador del curso
idusuario	BIGINT(20)	PK	Identificador del usuario
rol	INTEGER(10)		Rol del usuario dentro del curso (profesor, estudiante)
lastlogin	BIGINT(20)		Última fecha de entrada al curso

virtual_enlace_curso

Esta tabla almacena la información de los enlaces que el profesor ha definido dentro de un curso.

Columna	Tipo	Llave	Descripción
idenlace	INTEGER(11)	PK	Identificador del enlace
idcurso	BIGINT(20)		Identificador del curso
descripcion	TEXT		Descripción del enlace
url	TEXT		Dirección URL del enlace
tipo	INTEGER(11)		Tipo de enlace (archivo, página web)

virtual_foro_curso

Esta tabla almacena la información de los foros de un curso.

Columna	Tipo	Llave	Descripción
idforo	BIGINT(20)	PK	Identificador del foro
idcurso	BIGINT(20)		Identificador del curso
nombre	TEXT		Nombre del foro
descripcion	TEXT		Descripción

inicio	BIGINT(20)		Fecha de inicio (timestamp)
fin	BIGINT(20)		Fecha de finalización (timestamp)
activo	INTEGER(11)		Permite determinar si el foro está activo (1=activo, 0=inactivo)

virtual_foro_general

En esta tabla se almacena la información de los foros generales de Aula Virtual, manejados por los usuarios administradores.

Columna	Tipo	Llave	Descripción
idforo	BIGINT(20)	PK	Identificador del foro
nombre	TEXT		Nombre del foro
descripcion	TEXT		Descripción
inicio	BIGINT(20)		Fecha de inicio (timestamp)
fin	BIGINT(20)		Fecha de finalización (timestamp)
activo	INTEGER(10)		Permite determinar si el foro está activo (1=activo, 0=inactivo)

virtual_grupo_curso

Esta tabla almacena la información de los grupos creados dentro de un curso.

Columna	Tipo	Llave	Descripción
idgrupo	BIGINT(20)	PK	Identificador del grupo
idcurso	BIGINT(20)		Identificador del curso
nombre	TEXT		Nombre del grupo
descripcion	TEXT		Descripción
activo	INTEGER(11)		Permite determinar si el grupo está activo (1=activo, 0=inactivo)
privado	INTEGER(11)		Permite definir si todos los participantes tienen acceso al grupo, o sólo los integrantes del grupo (1=sólo grupo, 0=todos)

virtual_integrante_grupo_curso

Esta tabla almacena la información de los integrantes de un grupo definido dentro de un curso.

Columna	Tipo	Llave	Descripción
idusuario	BIGINT(20)	PK	Identificador del usuario
idgrupo	BIGINT(20)	PK	Identificador del grupo
rol	INTEGER(11)		Rol dentro del grupo (1=monitor, 2=integrante)

virtual_item

Esta tabla almacena los ítems (temas) que conforman una organización de contenidos.

Columna	Tipo	Llave	Descripción
idorganizacion	BIGINT(20)		Identificador de la organización
idtema	BIGINT(20)	PK	Identificador del tema
titulo	TEXT		Título del tema
descripcion	TEXT		Descripción del tema
texto	TEXT		Texto del tema (HTML)
tipo	TEXT		Tipo de tema <ul style="list-style-type: none">• webcontent: hipervínculo a una página, a un Asset o a un SCO• Label: texto HTML• package: Agregación de contenidos SCORM
url	TEXT		Dirección URL del ítem
orden	INTEGER(11)		Orden del ítem con respecto a sus similares
visible	INTEGER(11)		Determina si el ítem es visible o no dentro de los temas mostrados a los estudiantes (1=visible, 0=no visible)

virtual_lista_curso

Esta tabla almacena las intervenciones realizadas en la lista de discusión de un curso determinado.

Columna	Tipo	Llave	Descripción
idparticipacion	BIGINT(20)	PK	Identificador de la participación
idcurso	BIGINT(20)		Identificador del curso
idusuario	BIGINT(20)		Identificador del usuario
parent	INTEGER(11)		Participación que se está respondiendo. Para las participaciones que inician un nuevo tema de discusión, este campo toma el valor de 0
titulo	TEXT		Título de la participación
mensaje	TEXT		Texto de la participación
fecha	BIGINT(20)		Fecha de la participación (timestamp)

virtual_lista_foro_curso

Esta tabla almacena las participaciones en los foros de un curso específico.

Columna	Tipo	Llave	Descripción
idparticipacion	BIGINT(20)	PK	Identificador de la participación
idforo	BIGINT(20)		Identificador del foro
idusuario	BIGINT(20)		Identificador del usuario
parent	BIGINT(20)		Participación que se está respondiendo. Para las participaciones que inician un nuevo tema, este campo toma el valor de 0
titulo	TEXT		Título de la participación
mensaje	TEXT		Texto de la participación
fecha	BIGINT(20)		Fecha de la participación (timestamp)

virtual_lista_foro_general

Esta tabla almacena las participaciones en el foro general de Aula Virtual.

Columna	Tipo	Llave	Descripción
idparticipacion	BIGINT(20)	PK	Identificador de la participación
idforo	BIGINT(20)		Identificador del foro
idusuario	BIGINT(20)		Identificador del usuario
parent	INTEGER(10)		Participación que se está respondiendo
titulo	TEXT		Título de la participación
mensaje	TEXT		Texto de la participación
fecha	BIGINT(20)		Fecha de la participación (timestamp)

virtual_lista_grupo_curso

Esta tabla almacena las participaciones en la lista de discusión de un grupo, dentro de un curso determinado.

Columna	Tipo	Llave	Descripción
idparticipacion	BIGINT(20)	PK	Identificador de la participación
idgrupo	BIGINT(20)		Identificador del grupo
idusuario	BIGINT(20)		Identificador del usuario
parent	BIGINT(20)		Participación que se está respondiendo
titulo	TEXT		Título de la participación
mensaje	TEXT		Texto de la participación
fecha	BIGINT(20)		Tipo de tema <ul style="list-style-type: none"> • webcontent: hipervínculo a una página, a un Asset o a un SCO • Label: texto HTML • package: Agregación de contenidos SCORM

virtual_organizacion

Esta tabla almacena las organizaciones de contenido creadas por un usuario.

Columna	Tipo	Llave	Descripción
---------	------	-------	-------------

idorganizacion	BIGINT(20)	PK	Identificador de la organización
idusuario	BIGINT(20)		Identificador del usuario
titulo	TEXT		Título de la organización
descripcion	TEXT		Descripción de la organización
lenguaje	VARCHAR(6)		Lenguaje de la organización de contenidos

virtual_organizacion_tema

Esta tabla almacena los temas que se despliegan en el módulo de contenidos del Aula Virtual.

Columna	Tipo	Llave	Descripción
idcurso	BIGINT(20)		Identificador del curso
idtema	BIGINT(20)	PK	Identificador del tema
idparent	BIGINT(20)		Tema padre
titulo	TEXT		Título del tema
descripcion	TEXT		Descripción del tema
texto	TEXT		Texto del tema
tipo	TEXT		Tipo de tema <ul style="list-style-type: none"> • webcontent: hipervínculo a una página, a un Asset o a un SCO • Label: texto HTML • package: Agregación de contenidos SCORM
url	TEXT		Dirección URL del tema
orden	INTEGER(11)		Orden del tema dentro de la jerarquía
visible	INTEGER(11)		Determina si el tema es visible (1=visible, 0=no visible)

virtual_taller

En esta tabla se almacena la información de los talleres definidos dentro de un curso.

Columna	Tipo	Llave	Descripción
---------	------	-------	-------------

idtaller	BIGINT(20)	PK	Identificador del taller
idcurso	BIGINT(20)		Identificador del curso
nombre	TEXT		Nombre del taller
descripcion	TEXT		Descripción
inicio	BIGINT(20)		Fecha de inicio (timestamp)
fin	BIGINT(20)		Fecha de finalización (timestamp)
activo	INTEGER(11)		Determina si el taller está activo

virtual_actividad_personal

Esta tabla almacena las actividades personales de un usuario,

Columna	Tipo	Llave	Descripción
idactividad	BIGINT(20)	PK	Identificador de la actividad
idusuario	BIGINT(20)		Identificador del usuario
inicio	BIGINT(20)		Fecha de inicio de la actividad (timestamp)
fin	BIGINT(20)		Fecha de finalización
descripcion	TEXT		Descripción de la actividad

virtual_admin

Esta tabla se utiliza para definir los usuarios administradores de Aula Virtual.

Columna	Tipo	Llave	Descripción
idusuario	BIGINT(20)	PK	Identificador de usuario
admin_usuarios	INTEGER(11)		Tiene permitido administrar usuarios? (1=sí, 0=no)
admin_cursos	INTEGER(11)		Tiene permitido administrar cursos? (1=sí, 0=no)
admin_temas	INTEGER(11)		Tiene permitido administrar los temas gráficos de Aula Virtual? (1=sí, 0=no)
inicio	BIGINT(20)		Fecha de inicio (timestamp)
fin	BIGINT(20)		Fecha de caducidad del rol (timestamp)

virtual_lista_general

Esta tabla almacena las participaciones de los usuarios en la lista de discusión general de Aula Virtual.

Columna	Tipo	Llave	Descripción
idparticipacion	BIGINT(20)	PK	Identificador de la participación
idusuario	BIGINT(20)		Identificador del usuario
parent	INTEGER(11)		Participación que se está respondiendo
titulo	TEXT		Título de la participación
mensaje	TEXT		Texto de la participación
fecha	BIGINT(20)		Fecha de la participación (timestamp)

virtual_recurso

Esta tabla permite almacenar enlaces a los recursos que posee un usuario en diferentes repositorios digitales.

Columna	Tipo	Llave	Descripción
idrecurso	BIGINT(20)	PK	Identificador del recurso
idrepositorio	BIGINT(20)		Identificador del repositorio en el cual se encuentra almacenado el recurso
idusuario	BIGINT(20)		Identificador del propietario del recurso
resourceid	TEXT		Identificador del recurso dentro del repositorio
titulo	TEXT		Título del recurso
descripcion	TEXT		Descripción
url	TEXT		Dirección URL donde reside el recurso
file	TEXT		Dirección URL donde reside el archivo comprimido que contiene los archivos que conforman el recurso
lenguaje	VARCHAR(6)		Lenguaje del recurso

virtual_repositorio

Esta tabla almacena los repositorios de objetos de contenido que han sido adicionados por un usuario administrador.

Columna	Tipo	Llave	Descripción
idrepositorio	BIGINT(20)	PK	Identificador del repositorio
nombre	TEXT		Nombre del repositorio
descripcion	TEXT		Descripción del repositorio
wSDL_path	TEXT		Dirección URL de la interfaz de definición de servicios (documento WSDL)
url	TEXT		Dirección de la interfaz de acceso web al repositorio
activo	TINYINT(4)		Determina si se permiten o no nuevas adiciones de recursos al repositorio (1=permitido, 0=no permitido).

virtual_usuario_repositorio

En esta tabla se almacenan los datos de suscripción de los usuarios dentro de los repositorios.

Columna	Tipo	Llave	Descripción
idusuario	BIGINT(20)		Identificador del usuario
idrepositorio	BIGINT(20)		Identificador del repositorio
repemail	VARCHAR(100)		Correo electrónico con el cual se realizó el registro en el repositorio
repkey	TEXT		Llave única de usuario dentro del repositorio

virtual_adl_comments_from_learner (SCORM)

Esta tabla almacena los comentarios de los usuarios de recursos SCORM.

Columna	Tipo	Llave	Descripción
learner_id	VARCHAR(200)		Identificador del usuario
package_id	VARCHAR(200)		Identificador del paquete
sco_id	VARCHAR(200)		Identificador del sco dentro del paquete

comment	TEXT		Comentario
location	TEXT		Localización
timestamp	TEXT		Fecha de adición (timestamp)

virtual_adl_content_package (SCORM)

En esta tabla se almacena la información temporal de la ejecución de un paquete de contenidos SCORM.

Columna	Tipo	Llave	Descripción
package_id	VARCHAR(200)	PK	Identificador del paquete SCORM
description	TEXT		Descripción del paquete
activity_tree	LONGTEXT		Árbol de actividades del paquete de contenidos, representado como un documento serializado.
url	TEXT		Dirección URL en donde se pueden localizar los archivos que conforman el paquete

virtual_error

Esta tabla almacena los códigos de error internos que posee Aula Virtual.

Columna	Tipo	Llave	Descripción
iderror	BIGINT(20)	PK	Identificador del error
descripcion	TEXT		Descripción

virtual_tema

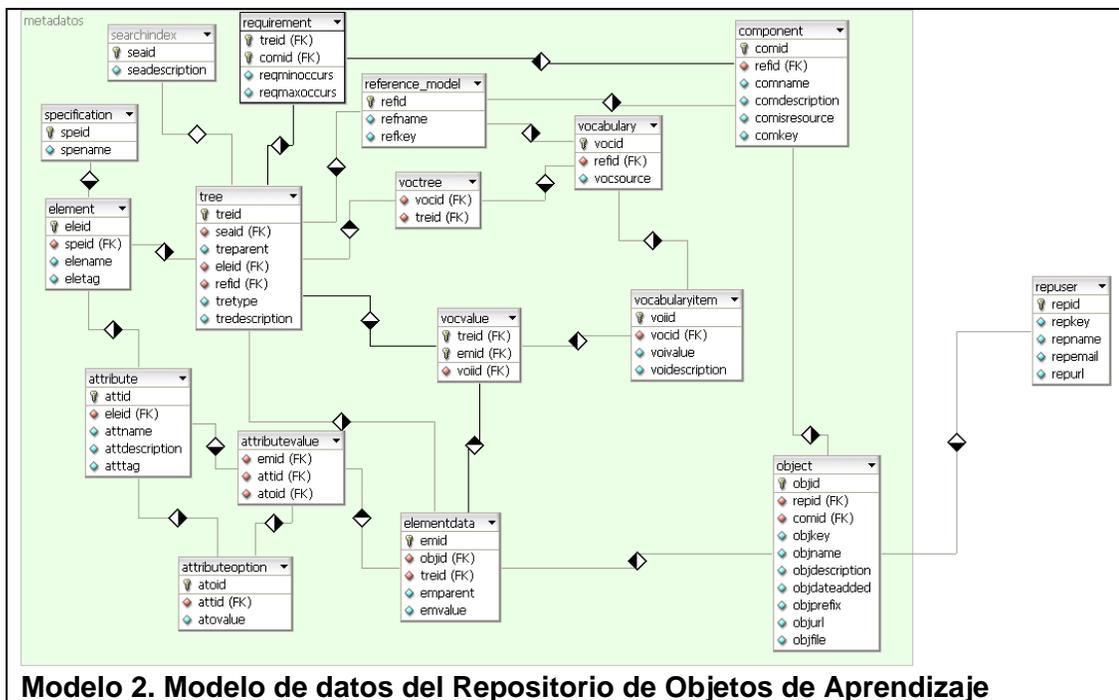
En esta tabla se almacena la información de los temas gráficos disponibles para Aula Virtual.

Columna	Tipo	Llave	Descripción
idtema	BIGINT(20)	PK	Identificador del tema

descripcion	TEXT		Descripción
path	TEXT		Ruta del tema
activo	TINYINT(4)		Determina si el tema está activo (1=activo,

Anexo 3. Modelo de datos del Repositorio de Objetos de Aprendizaje

A continuación se presenta el modelo de datos del repositorio de objetos de aprendizaje, implementado en una base de datos MySQL.



Modelo 2. Modelo de datos del Repositorio de Objetos de Aprendizaje

A continuación se presenta el diccionario del Repositorio de Objetos de Aprendizaje, que describe la estructura de las tablas y los campos.

reference_model

Esta tabla almacena los modelos de referencia que soporta el repositorio.

Campo	Tipo	Llave	Descripción
refid	INTEGER	PK	Identificador del modelo
refname	TEXT		Nombre del modelo de referencia
refkey	VARCHAR(20)		Llave única del modelo de referencia

specification

Esta tabla almacena las diferentes especificaciones que soporta el repositorio.

Campo	Tipo	Llave	Descripción
speid	INTEGER	PK	Identificador de la especificación
sname	TEXT		Nombre de la especificación

element

Esta tabla almacena los elementos que conforman los esquemas de datos XML.

Campo	Tipo	Llave	Descripción
eleid	INTEGER	PK	Identificador del elemento
speid	INTEGER		Identificador de la especificación
elename	TEXT		Nombre del elemento
eletag	TEXT		Etiqueta XML del elemento

tree

Esta tabla almacena la distribución jerárquica de los elementos que conforman una especificación de metadatos.

Campo	Tipo	Llave	Descripción
treid	INTEGER	PK	Identificador del nodo
seaid	INTEGER		Identificador del índice de búsqueda
treparent	INTEGER		Identificador del nodo padre
eleid	INTEGER		Identificado del elemento de la especificación
refid	INTEGER		Identificador del modelo de referencia
tretype	CHAR		Tipo de nodo: <ul style="list-style-type: none">• P: Nodo contenedor• L: Cadena de caracteres con lenguaje• D: Elemento fecha• T: Elemento duración

			• C: Cadena de caracteres sin lenguaje
tredescription	TEXT		Descripción del nodo

elementdata

Esta tabla almacena los valores de los metadatos para un objeto de aprendizaje.

Campo	Tipo	Llave	Descripción
emid	INTEGER	PK	Identificador del registro de metadatos
objid	INTEGER		Identificador del objeto de aprendizaje
treid	INTEGER		Identificador del nodo del árbol
emparent	INTEGER		Identificador del nodo de valores padre
emvalue	TEXT		Valor del nodo de metadatos

attribute

Esta tabla almacena los atributos XML de los diferentes elementos de metadatos.

Campo	Tipo	Llave	Descripción
attid	INTEGER	PK	Identificador del atributo
eleid	INTEGER		Identificador del elemento
attname	TEXT		Nombre del atributo
attdescription	TEXT		Descripción del atributo
atttag	TEXT		Etiqueta XML del traje.

attributevalue

Esta tabla almacena los valores de los atributos, para un determinado registro de metadatos.

Campo	Tipo	Llave	Descripción
emid	INTEGER		Identificador del registro de metadatos
attid	INTEGER		Identificador del atributo

atoid	INTEGER		Identificador del registro de valor
-------	---------	--	-------------------------------------

component

Esta tabla almacena los componentes, que conforman un modelo de referencia.

Campo	Tipo	Llave	Descripción
comid	INTEGER	PK	Identificador del componente
refid	INTEGER		Identificador del modelo de referencia
comname	TEXT		Nombre del componente
comdescription	TEXT		Descripción del componente
comisresource	CHAR(1)		Determina si el componente es un recurso (Y=Sí, N=No).
comkey	VARCHAR(20)		Llave única del componente dentro del repositorio.

requirement

Esta tabla determina los elementos requerimientos de obligatoriedad de los elemento del esquema de datos de un modelo de referencia, de acuerdo con los componentes.

Campo	Tipo	Llave	Descripción
treid	INTEGER	PK	Identificador del nodo de la jerarquía de metadatos
comid	INTEGER	PK	Identificador del componente
reqminoccurs	INTEGER		Mínimo número de ocurrencias permitidas
reqmaxoccurs	INTEGER		Máximo número de ocurrencias permitidas

vocabulary

Esta tabla almacena los diferentes vocabularios que soporta un modelo de referencia.

Campo	Tipo	Llave	Descripción
-------	------	-------	-------------

vocid	INTEGER	PK	Identificador del vocabulario
refid	INTEGER		Identificador del modelo de referencia
vocsource	TEXT		Texto de la fuente del vocabulario (Para los vocabularios LOM, este elemento siempre tiene el valor de LOMv1.0).

vocabularyitem

En esta tabla se almacenan las diferentes opciones para un vocabulario.

Campo	Tipo	Llave	Descripción
voiid	INTEGER	PK	Identificador de la opción
vocid	INTEGER		Identificador del vocabulario
voivalue	TEXT		Valor del vocabulario
voidescription	TEXT		Descripción de la opción

vocvalue

En esta tabla se almacenan los valores de vocabulario de los registros de metadatos que los requieren.

Campo	Tipo	Llave	Descripción
treid	INTEGER	PK	Identificador del nodo del árbol de metadatos
emid	INTEGER	PK	Identificador del registro de metadatos
voiid	INTEGER		Identificador de la opción del vocabulario

attributeoption

Esta tabla almacena las posibles opciones para un atributo de un elemento XML

Campo	Tipo	Llave	Descripción
atoid	INTEGER	PK	Identificador de la opción
attid	INTEGER		Identificador del atributo
atovalue	TEXT		Valor del atributo

voctree

En esta tabla se almacena la información de los nodos del árbol de metadatos que dependen de un vocabulario.

Campo	Tipo	Llave	Descripción
vocid	INTEGER		Identificador del vocabulario
treid	INTEGER		Identificador del nodo en el árbol de metadatos

searchindex

Esta tabla almacena los posibles índices de búsqueda de los elementos de un esquema de metadatos, de acuerdo con IMS DRI.

Campo	Tipo	Llave	Descripción
seaid	INTEGER	PK	Identificador del índice
seadescription	TEXT		Descripción del índice

repuse

Esta tabla almacena la información acerca de los usuarios del repositorio.

Campo	Tipo	Llave	Descripción
repid	BIGINT	PK	Identificador del usuario
repkey	TEXT		Llave única del usuario
repname	TEXT		Nombre
repemail	TEXT		Correo electrónico
repurl	TEXT		Dirección URL de contacto

Anexo 4. Documento de definición de los servicios de intercambio de la plataforma

A continuación se presenta el documento WSDL que define los servicios ofrecidos por la plataforma e intercambio de materiales desarrollada en el presente trabajo de investigación. Este documento puede ser utilizado para obtener acceso a los servicios ofrecidos por la implementación desarrollada, o para crear nuevas implementaciones de repositorios y ambientes virtuales de aprendizaje.

La implementación de la interfaz de servicios fue desarrollada en PHP 5.0, y ha sido probada con clientes implementados en PHP 5.0 y Microsoft C#.NET 2003. Igualmente, parte de esta implementación ha sido tomada como base en el proyecto “SCORM Public-Access Repository”, financiado por Microsoft dentro de su programa Microsoft Research eScience and SciData RFP Awards (2004), y desarrollado por el Grupo de I+D en Tecnologías de la Información, adscrito al Departamento de Sistemas de la Universidad del Cauca, Colombia.

(Espacio en blanco intencional para conservar el formato del documento)

```

<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="http://tempuri.org/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" targetNamespace="http://tempuri.org/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
      <s:element name="addResource">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="key" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="model" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="component" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="name" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="description" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="language" type="s:string"
              nillable="true" />
            <s:element minOccurs="1" maxOccurs="1" name="url" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="launch" nillable="true" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="filename" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="unzip" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="addResourceResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="addResourceResult"
              type="tns:addResourceValue" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="addResourceValue">
        <s:sequence>

```

```
<s:element minOccurs="0" maxOccurs="1" name="ErrorCode" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="ErrorDescription" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="ResourceId" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="Uri" type="s:string" />
</s:sequence>
</s:complexType>

<s:element name="getReferenceModelList">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="key" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="getReferenceModelListResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="getReferenceModelListResult"
        type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="getComponentList">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="key" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="model" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="getComponentListResponse">
  <s:complexType>
    <s:sequence>
```

```
        <s:element minOccurs="0" maxOccurs="1" name="GetComponentListResult"
            type="s:string" />
    </s:sequence>
</s:complexType>
</s:element>
<s:element name="suscribe">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="name" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="email" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="url" type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="suscribeResponse">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="suscribeResult" type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="loadMetadata">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="key" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="resource" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="metadata" type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="loadMetadataResponse">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="loadMetadataResult" type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>
```

```
</s:element>
<s:element name="getMetadata">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="key" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="resource" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="getMetadataResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="getMetadataResult" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="metadataTextQuery">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="key" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="type" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="text" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="since" type="s:string" nillable="true" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="metadataTextQueryResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="metadataTextQueryResult"
        type="tns:ArrayOfQueryResult" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:complexType name="ArrayOfQueryResult">
  <s:sequence>
```

```

    <s:element minOccurs="0" maxOccurs="unbounded" name="QueryResult" nillable="true"
        type="tns:QueryResult" />
</s:sequence>
</s:complexType>
<s:complexType name="QueryResult">
    <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="Resource" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="Type" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="Date" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="Text" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="Name" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="Description" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="Url" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="File" type="s:string" />
    </s:sequence>
</s:complexType>

<!-- La misma implementacion de metadataTextQuery, con la diferencia que regresa un XML -->
<s:element name="metadataQueryXML">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="key" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="type" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="text" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="since" type="s:string" nillable="true" />
            <s:element minOccurs="0" maxOccurs="1" name="owner" type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>

<s:element name="metadataQueryXMLResponse">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="metadataQueryXMLResult"
                type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>

```

```
</s:complexType>
</s:element>

<s:element name="getResourceList">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="key" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="text" type="s:string" nillable="true" />
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="getResourceListResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="getResourceListResult" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>

  <s:element name="getResourceInfo">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="key" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="resource" type="s:string" />
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="getResourceInfoResponse">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="getResourceInfoResult"
          type="tns:resourceInfo" />
      </s:sequence>
    </s:complexType>
  </s:element>

```

```
</s:element>
<s:complexType name="resourceInfo">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="name" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="description" type="s:string" />
  </s:sequence>
</s:complexType>

<s:element name="updateResourceInfo">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="key" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="resource" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="name" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="description" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="updateResourceInfoResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="updateResourceInfoResult"
        type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="isOnline">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="token" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
```

```
<s:element name="isOnlineResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="isOnlineResponseResult"
        type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="isValidReferenceModel">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="key" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="model" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="isValidReferenceModelResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="isValidReferenceModelResult"
        type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>

<s:element name="isValidComponent">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="key" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="model" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="component" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
```

```

    <s:element name="isValidComponentResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="isValidComponentResult"
            type="s:string" />
        </s:sequence>
      </s:complexType>
    </s:element>

  </s:schema>
</wsdl:types>
<!-- inicio de los mensajes -->
<wsdl:message name="addResourceSoapIn">
  <wsdl:part name="parameters" element="tns:addResource" />
</wsdl:message>
<wsdl:message name="addResourceSoapOut">
  <wsdl:part name="parameters" element="tns:addResourceResponse" />
</wsdl:message>
<wsdl:message name="suscribeSoapIn">
  <wsdl:part name="parameters" element="tns:suscribe" />
</wsdl:message>
<wsdl:message name="suscribeSoapOut">
  <wsdl:part name="parameters" element="tns:suscribeResponse" />
</wsdl:message>
<wsdl:message name="getReferenceModelListSoapIn">
  <wsdl:part name="parameters" element="tns:getReferenceModelList" />
</wsdl:message>
<wsdl:message name="getReferenceModelListSoapOut">
  <wsdl:part name="parameters" element="tns:getReferenceModelListResponse" />
</wsdl:message>
<wsdl:message name="getComponentListSoapIn">
  <wsdl:part name="parameters" element="tns:getComponentList" />
</wsdl:message>
<wsdl:message name="getComponentListSoapOut">
  <wsdl:part name="parameters" element="tns:getComponentListResponse" />
</wsdl:message>

```

```
<wsdl:message name="loadMetadataSoapIn">
  <wsdl:part name="parameters" element="tns:loadMetadata" />
</wsdl:message>
<wsdl:message name="loadMetadataSoapOut">
  <wsdl:part name="parameters" element="tns:loadMetadataResponse" />
</wsdl:message>
<wsdl:message name="getMetadataSoapIn">
  <wsdl:part name="parameters" element="tns:getMetadata" />
</wsdl:message>
<wsdl:message name="getMetadataSoapOut">
  <wsdl:part name="parameters" element="tns:getMetadataResponse" />
</wsdl:message>
<wsdl:message name="metadataTextQuerySoapIn">
  <wsdl:part name="parameters" element="tns:metadataTextQuery" />
</wsdl:message>
<wsdl:message name="metadataTextQuerySoapOut">
  <wsdl:part name="parameters" element="tns:metadataTextQueryResponse" />
</wsdl:message>
<wsdl:message name="metadataQueryXMLSoapIn">
  <wsdl:part name="parameters" element="tns:metadataQueryXML" />
</wsdl:message>
<wsdl:message name="metadataQueryXMLSoapOut">
  <wsdl:part name="parameters" element="tns:metadataQueryXMLResponse" />
</wsdl:message>
<wsdl:message name="getResourceListSoapIn">
  <wsdl:part name="parameters" element="tns:getResourceList" />
</wsdl:message>
<wsdl:message name="getResourceListSoapOut">
  <wsdl:part name="parameters" element="tns:getResourceListResponse" />
</wsdl:message>

<wsdl:message name="getResourceInfoSoapIn">
  <wsdl:part name="parameters" element="tns:getResourceInfo" />
</wsdl:message>
<wsdl:message name="getResourceInfoSoapOut">
  <wsdl:part name="parameters" element="tns:getResourceInfoResponse" />
</wsdl:message>
```

```
</wsdl:message>

<wsdl:message name="updateResourceInfoSoapIn">
  <wsdl:part name="parameters" element="tns:updateResourceInfo" />
</wsdl:message>
<wsdl:message name="updateResourceInfoSoapOut">
  <wsdl:part name="parameters" element="tns:updateResourceInfoResponse" />
</wsdl:message>

<wsdl:message name="isOnlineSoapIn">
  <wsdl:part name="parameters" element="tns:isOnline" />
</wsdl:message>
<wsdl:message name="isOnlineSoapOut">
  <wsdl:part name="parameters" element="tns:isOnlineResponse" />
</wsdl:message>

<wsdl:message name="isValidReferenceModelSoapIn">
  <wsdl:part name="parameters" element="tns:isValidReferenceModel" />
</wsdl:message>
<wsdl:message name="isValidReferenceModelSoapOut">
  <wsdl:part name="parameters" element="tns:isValidReferenceModelResponse" />
</wsdl:message>

<wsdl:message name="isValidComponentSoapIn">
  <wsdl:part name="parameters" element="tns:isValidComponent" />
</wsdl:message>
<wsdl:message name="isValidComponentSoapOut">
  <wsdl:part name="parameters" element="tns:isValidComponentResponse" />
</wsdl:message>

<wsdl:portType name="Service1Soap">
  <wsdl:operation name="addResource">
    <wsdl:input message="tns:addResourceSoapIn" />
    <wsdl:output message="tns:addResourceSoapOut" />
  </wsdl:operation>
</wsdl:portType>
```

```
</wsdl:operation>
<wsdl:operation name="suscribe">
  <wsdl:input message="tns:suscribeSoapIn" />
  <wsdl:output message="tns:suscribeSoapOut" />
</wsdl:operation>
<wsdl:operation name="getReferenceModelList">
  <wsdl:input message="tns:getReferenceModelListSoapIn" />
  <wsdl:output message="tns:getReferenceModelListSoapOut" />
</wsdl:operation>
<wsdl:operation name="getComponentList">
  <wsdl:input message="tns:getComponentListSoapIn" />
  <wsdl:output message="tns:getComponentListSoapOut" />
</wsdl:operation>
<wsdl:operation name="loadMetadata">
  <wsdl:input message="tns:loadMetadataSoapIn" />
  <wsdl:output message="tns:loadMetadataSoapOut" />
</wsdl:operation>
<wsdl:operation name="getMetadata">
  <wsdl:input message="tns:getMetadataSoapIn" />
  <wsdl:output message="tns:getMetadataSoapOut" />
</wsdl:operation>
<wsdl:operation name="metadataTextQuery">
  <wsdl:input message="tns:metadataTextQuerySoapIn" />
  <wsdl:output message="tns:metadataTextQuerySoapOut" />
</wsdl:operation>

<wsdl:operation name="metadataQueryXML">
  <wsdl:input message="tns:metadataQueryXMLSoapIn" />
  <wsdl:output message="tns:metadataQueryXMLSoapOut" />
</wsdl:operation>

<wsdl:operation name="getResourceList">
  <wsdl:input message="tns:getResourceListSoapIn" />
  <wsdl:output message="tns:getResourceListSoapOut" />
</wsdl:operation>
```

```
<wsdl:operation name="getResourceInfo">
  <wsdl:input message="tns:getResourceInfoSoapIn" />
  <wsdl:output message="tns:getResourceInfoSoapOut" />
</wsdl:operation>

<wsdl:operation name="updateResourceInfo">
  <wsdl:input message="tns:updateResourceInfoSoapIn" />
  <wsdl:output message="tns:updateResourceInfoSoapOut" />
</wsdl:operation>

<wsdl:operation name="isOnline">
  <wsdl:input message="tns:isOnlineSoapIn" />
  <wsdl:output message="tns:isOnlineSoapOut" />
</wsdl:operation>

<wsdl:operation name="isValidReferenceModel">
  <wsdl:input message="tns:isValidReferenceModelSoapIn" />
  <wsdl:output message="tns:isValidReferenceModelSoapOut" />
</wsdl:operation>

<wsdl:operation name="isValidComponent">
  <wsdl:input message="tns:isValidComponentSoapIn" />
  <wsdl:output message="tns:isValidComponentSoapOut" />
</wsdl:operation>

</wsdl:portType>
<wsdl:binding name="Service1Soap" type="tns:Service1Soap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
  <wsdl:operation name="addResource">
    <soap:operation soapAction="http://tempuri.org/addResource" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
</wsdl:service>
```

```
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="suscribe">
  <soap:operation soapAction="http://tempuri.org/suscribe" style="document" />
  <wsdl:input>
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="getReferenceModelList">
  <soap:operation soapAction="http://tempuri.org/getReferenceModelList" style="document" />
  <wsdl:input>
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="GetComponentList">
  <soap:operation soapAction="http://tempuri.org/getComponentList" style="document" />
  <wsdl:input>
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="loadMetadata">
  <soap:operation soapAction="http://tempuri.org/loadMetadata" style="document" />
  <wsdl:input>
```

```
<soap:body use="literal" />
</wsdl:input>
<wsdl:output>
  <soap:body use="literal" />
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="getMetadata">
  <soap:operation soapAction="http://tempuri.org/getMetadata" style="document" />
  <wsdl:input>
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="metadataTextQuery">
  <soap:operation soapAction="http://tempuri.org/metadataTextQuery" style="document" />
  <wsdl:input>
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="metadataQueryXML">
  <soap:operation soapAction="http://tempuri.org/metadataQueryXML" style="document" />
  <wsdl:input>
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>
```

```
<wsdl:operation name="getResourceList">
  <soap:operation soapAction="http://tempuri.org/getResourceList" style="document" />
  <wsdl:input>
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="getResourceInfo">
  <soap:operation soapAction="http://tempuri.org/getResourceInfo" style="document" />
  <wsdl:input>
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="updateResourceInfo">
  <soap:operation soapAction="http://tempuri.org/updateResourceInfo" style="document" />
  <wsdl:input>
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="isOnline">
  <soap:operation soapAction="http://tempuri.org/isOnline" style="document" />
  <wsdl:input>
```

```
<soap:body use="literal" />
</wsdl:input>
<wsdl:output>
  <soap:body use="literal" />
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="isValidReferenceModel">
  <soap:operation soapAction="http://tempuri.org/isValidReferenceModel" style="document" />
  <wsdl:input>
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="isValidComponent">
  <soap:operation soapAction="http://tempuri.org/isValidComponent" style="document" />
  <wsdl:input>
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>

</wsdl:binding>
<wsdl:service name="Service1">
  <documentation xmlns="http://schemas.xmlsoap.org/wsdl/" />
  <wsdl:port name="Service1Soap" binding="tns:Service1Soap">
    <soap:address location="http://tic.uis.edu.co/aula/repository/RepositoryService.php" />
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

