Herramienta de software libre para la construcción de contenido SCORM: un caso de estudio

Francisco Javier Díaz[1], Alejandra Schiavoni[1]
Nidia Banchero[1]

[1] LINTI – Laboratorio de Investigación en Nuevas Tecnologías Informáticas
Calle 50 y 120, 2do Piso
La Plata, Argentina
jdiaz@unlp.edu.ar, ales@info.unlp.edu.ar, n_banchero@yahoo.com.ar

Resumen

En los últimos años, el aprendizaje electrónico ha cobrado vital importancia, debido al extendido uso de las IT. Por ello, han surgido un conjunto de estándares tendientes a desarrollar contenido educativo de alta calidad. La tarea de creación de contenido SCORM, requiere de una serie de pasos que consisten en la creación de los objetos de aprendizaje, y en la incorporación de metadatos y funciones de comunicación con el LMS. A través de esta comunicación es posible registrar la información del desempeño del alumno en el seguimiento del curso. Sin embargo, incorporar esta comunicación no representa una tarea sencilla, ya que es necesario conocer detalles de implementación del estándar. En este artículo se propone una herramienta Open Source cuyo objetivo es simplificar la incorporación de funciones que implementen la comunicación entre los objetos de aprendizaje y el LMS. La herramienta consiste en extender la funcionalidad de un editor HTML, para que en el momento de creación de una página el autor tenga la posibilidad de agregar componentes predefinidos orientados a la comunicación. La funcionalidad de la herramienta se muestra en el desarrollo de un caso concreto.

1. Introducción

Desde hace unos años, el proceso de aprendizaje ha variado considerablemente hacia lo que se conoce como e-learning (electronic learning), debido al extendido uso de las tecnologías de la Información y la Comunicación que soportan esta nueva forma de educación, también llamada *educación a distancia*. Por esta razón, diversas organizaciones internacionales vienen trabajando para lograr un conjunto de estándares cuyo objetivo es el desarrollo de sistemas de e-learning, herramientas y metodologías que combinadas producen e-Educación de alta calidad [1]. Los estándares más difundidos hoy en el mercado son: IEEE LTSC LOM [http://ltsc.ieee.org/wg12/], AICC [http://www.aicc.org] y ADL SCORM [http://www.adlnet.org/], aunque también existen otros como Dublín Core [http://dublincore.org] y EML [http://eml.ou.nl/eml-ou-nl.htm]. En la evolución de estos estándares, se produjo un proceso de convergencia hacia ADL SCORM, el cual integra IMS, LOM y AICC y que hoy en día aparece como el estándar internacional más conocido.
La posibilidad de disponer de material educativo reusable, durable, accesible e interoperable, y que pueda, de esta manera, ser utilizado desde cualquier plataforma de aprendizaje o LMS (Learning Management System) representa un hecho muy prometedor. Sin embargo, habiendo transcurrido varios años desde la aparición de los estándares mencionados, la creación de objetos de aprendizaje que cumplan las

normas, sigue siendo una tarea costosa y que requiere ciertos conocimientos técnicos específicos. La generación de contenido reutilizable agrega complejidad a la mera generación de contenido [2]. Por esta razón, es que se ha retardado considerablemente la imposición de e-learning como modalidad de aprendizaje universal cuando se habla de las nuevas tendencias en educación. Al momento de usar el estándar SCORM, existen herramientas que facilitan la creación y validación de los objetos, pero que no cubren todos los aspectos. Quienes intentan crear su propio material encuentran diversas dificultades, ya que deben conocer en detalle las características del estándar, hecho que comúnmente no sucede.
En particular, implementar la comunicación entre el LMS y el contenido no representa una tarea sencilla, ya que se debe conocer tanto el lenguaje JavaScript como el modelo de datos con el que trabaja el estándar para poder lograr el intercambio de información. Este intercambio resulta de gran utilidad si se desea registrar datos del alumno y su evolución durante el seguimiento del curso.
Este artículo describe en detalle los pasos a seguir en la creación y uso de objetos de aprendizaje y propone una herramienta basada en entornos de software libre que tiene como objetivo facilitar uno de los aspectos presentes en este proceso. La funcionalidad de esta herramienta consiste en simplificar la incorporación de funciones que implementen la comunicación entre los objetos de aprendizaje y el LMS. En las próximas secciones se muestran las etapas del proceso de creación de contenido SCORM, sus dificultades y la funcionalidad de la herramienta propuesta. Además, se describe un caso práctico de desarrollo de un curso utilizando la herramienta, y cómo se visualiza en un LMS como Moodle.

2. Creación de contenido SCORM

El estándar SCORM enfoca todos sus esfuerzos en la creación de material educativo que sea reusable, interoperable, durable y accesible independientemente del sistema que se utilice para su administración y visualización [3]. Para ello propone el uso de *objetos de aprendizaje compartidos* (SCOs, Shareable Content Objects) denominados *recursos* (assets), compuestos de material educativo y que pueden incluir objetivos, autoevaluaciones, simulaciones, etc. Estos objetos pueden ser entregados en un entorno de ejecución SCORM, eventualmente provisto por un LMS o desde un repositorio de objetos de aprendizaje capaz de dar cumplimiento a los requerimientos SCORM. Además, el uso de metadatos permite identificar y ubicar el material educativo, por ejemplo en repositorios de contenidos en línea.
SCORM hace una clara distinción entre el rol del LMS y el rol del contenido en sí mismo. El hecho de confundir estos roles es una de las principales causas de la falta de interoperabilidad y reusabilidad [4]. Por ejemplo, al usar SCORM toda la información relativa al estudiante es almacenada en el LMS y no dentro del material educativo.
La creación de contenido compatible con SCORM es un proceso que lleva una serie de etapas, según lo menciona la guía de LSAL Carnegie Mellon [3]. En primer lugar, la tarea es identificar y diseñar los SCO. Es importante tener en cuenta que una forma de asegurar la integridad instruccional del contenido SCORM, es que cada SCO represente una unidad de aprendizaje independiente.
Si estamos trabajando con un curso creado con anterioridad y queremos convertirlo al estándar SCORM, es necesario hacer un proceso de reingeniería. El primer paso es estructurar el contenido armando una jerarquía compuesta por módulos, donde cada módulo, a su vez, está compuesto por lecciones. El siguiente paso es identificar las audiencias potenciales, para luego determinar cada uno de los SCOs que integrarán el material. En el artículo “Standardizing educative content with SCORM: a case study” se describe cómo se convirtió a SCORM un curso diseñado dentro del Plan de Capacitación en el uso de las TICs del Ministerio de Educación de la Nación y que fue dictado por la Facultad de Informática de la Universidad Nacional de La Plata [5].

Una vez definidos los SCOs, es importante la incorporación de metadatos, que permite que los objetos sean localizados en forma eficiente. El esquema de metadatos que usa SCORM está basado en el estándar LOM (Learning Object Metadata), e incluye nueve categorías de campos divididos en subcategorías. La elección de los metadatos a incorporar depende del tipo y volumen del contenido que se está creando.
Existen, además, otros aspectos a considerar como son la implementación del secuenciamiento y navegación del contenido implementados a través de un conjunto de reglas, y la creación de tests o evaluaciones dentro de los objetos SCORM , tareas que resultan bastante difíciles para autores no especializados [2] [6].
Uno de los puntos más interesantes del estándar SCORM es que permite llevar a cabo la comunicación entre los objetos de aprendizaje y el LMS. A través de esta comunicación es posible registrar la información relativa al desempeño del alumno en el seguimiento del curso. En las siguientes secciones profundizaremos este aspecto junto con las características de la herramienta propuesta y mostraremos el desarrollo de un caso concreto.
La necesidad de contar con metadatos, empaquetamiento y un secuenciamiento que pueda ser interpretado sin ambigüedades, lleva a imponer una norma y lograr su implementación correcta en los distintos LMSs. La norma SCORM se propone estos objetivos, pero tropieza también con dificultades de adopción por los autores debido a la complejidad de los metadatos, la comunicación con el LMS y los cambios de una versión a otra [2].
En SCORM, la fase de desarrollo requiere experiencia técnica adicional para cumplir las pautas de conformidad del modelo de referencia. Los programadores y desarrolladores deben tener los conocimientos básicos de la tecnología requerida para aplicar SCORM, basada en las guias Aggregation Model, Run-time Environment, y Sequencing y Navigation para los Objetos de Aprendizaje. Ellos deben ser capaces de estructurar los datos apropiadamente y agregar metadatos a las organizaciones, agregaciones, actividades, SCOs y assets.[7].

3. Comunicación entre el contenido y LMS

Como mencionamos anteriormente una de las grandes ventajas del modelo SCORM es que el contenido, representado a través de los objetos de aprendizaje, puede comunicar información sobre el alumno a cualquier sistema LMS compatible con SCORM. La especificación SCORM determina exactamente cuáles son las piezas de información que se pueden recuperar y actualizar: identificación del alumno, nombre, puntuación en tests, tiempo empleado en cada objeto de aprendizaje y sus preferencias de visualización.
La comunicación se realiza mediante una interfaz que estandariza los protocolos de comunicación proporcionando métodos para que el LMS pueda conseguir el estado actual (inicializado, finalizado, etc.) de los objetos y para el envío de datos entre ambos.
La utilización de una interfaz (API, Application Program Interface) proporciona una forma estandarizada para que los contenidos se comuniquen con el LMS, aunque la implementación de esta comunicación es transparente para el desarrollador de los contenidos. La forma en la que esté implementada la API no es importante para los desarrolladores, pero todos deben usar la interfaz externa que ofrece la funcionalidad de la API. Ésta esconde los detalles de implementación a los contenidos permitiendo con ello la reutilización e interoperabilidad de los mismos.
El adaptador de la API (API Adapter) es el software que expone las diferentes funciones de la API y está basada en JavaScript. Este adaptador hace posible la comunicación entre los SCO´s y el LMS, pues es el medio que se les ofrece a los SCO´s para conversar con el LMS.

Con este adaptador se pretende conseguir una independencia entre los contenidos y el entorno de ejecución. De esta manera, los encargados de la implementación de entornos de enseñanza, no tienen más que tener en cuenta las funciones que ofrece la API, y los creadores de contenidos no se tienen que preocupar por el tipo de sistema en el que serán mostrados o el tipo de plataforma. La aparición de este adaptador supone la separación total entre contenidos y sistemas.
Para establecer la comunicación, el estándar SCORM utiliza, además de la API, un modelo de datos que define el flujo de datos intercambiado entre el entorno LMS y el contenido.
El propósito de establecer un modelo de datos común es asegurarse que la información sobre el SCO pueda ser seguida por diferentes LMSs. Si, por ejemplo, se determina que seguir la puntuación de un alumno es un requerimiento del sistema, entonces es necesario establecer una vía común en el contenido para informar al LMS de las puntuaciones. Si los SCO usan su propio sistema de puntuaciones, los sistemas de aprendizaje no sabrían como recibir, almacenar o procesar la información.

Hay un número de modelos de datos bajo desarrollo en varias comunidades y organizaciones estándar. El modelo de datos del entorno de ejecución de SCORM deriva directamente del modelo de datos de AICC CMI. Los elementos del modelo de datos están divididos en 2 categorías: obligatorios y opcionales. La guía AICC CMI001 entrega recomendaciones y lineamientos a seguir para lograr la interoperabilidad entre LMSs y especifica cuáles de los elementos de datos son obligatorios y cuáles opcionales. Los elementos de datos obligatorios deben ser admitidos por nuestro LMS. Además el LMS también puede proporcionar implementación para admitir algunos o todos los elementos del modelo de datos opcionales.
Según lo expuesto anteriormente para poder llevar a cabo esta comunicación es necesario conocer detalles de implementación del estándar SCORM, como el RTE Management, los requerimientos de comunicación expuestos por la API y los requerimientos de administración y comportamiento del modelo de datos. El objetivo de la herramienta propuesta es facilitar la incorporación de las funciones de comunicación entre los SCOs y el LMS utilizado.

4. Herramienta para facilitar la comunicación con el LMS

La tarea de crear contenido SCORM enriquecido con toda la funcionalidad provista por este estándar, requiere, por un lado entender las características e implementación del estándar en sí mismo, y por otro lado conocer un conjunto de tecnologías que incluyen XML, HTML y JavaScript.
Siguiendo los pasos de la guía de LSAL Carnegie Mellon [3] descriptos con anterioridad, resulta sencilla la identificación de los SCOs que integrarán el curso. Además, toda persona que haya construido un curso SCORM, habrá podido observar que utilizando una herramienta para tal fin como ReloadEditor [http://www.reload.ac.uk/], fácilmente es posible importar los objetos que formarán parte del curso (documentos HTML, imágenes, archivos multimedia, etc.) y especificar cuál será la estructura del curso y sus metadatos básicos, obteniendo de esta forma un paquete SCORM. Este paquete puede ser visualizado con un LMS compatible con SCORM, como por ejemplo Moodle [http://moodle.org].
Si bien la construcción y visualización de un paquete SCORM son tareas simples, resueltas hoy por herramientas Open Source como las mencionadas, una de las mayores dificultades está dada en la incorporación de las funciones JavaScript que permiten que el contenido envíe o tome información del LMS. La complejidad de estas funciones crece a medida que crece el número y el detalle de la información que se desea enviar.
Para la implementación de esta herramienta se evaluaron distintas alternativas. Por un lado, se analizó el editor HTML embebido en Moodle, que es un editor gráfico WYSIWYG. Es un editor muy simple que permite únicamente ingresar texto plano con

tags HTML. Además de componer el texto directamente en el editor, también permite copiar y pegar texto con formatos y estructuras desde documentos generados en otro procesador de textos, insertar imágenes, tablas y enlaces. Con estas características representa un método cómodo y directo de componer una página web sin necesidad de conocer y dominar el lenguaje. La desventaja que presenta para nuestro proyecto es que no está permitida la escritura de scripts, y éstos son eliminados automáticamente al momento de almacenar la página. A consecuencia de estos resultados, el editor incluido en el Moodle no fue utilizado y se decidió trabajar sobre un editor HTML Open Source, que pudiera ser modificado para agregarle la funcionalidad deseada. Se eligió el editor FCKEditor [http://www.fckeditor.net/], que tiene todas las características básicas de esta clase de editores, ya que ofrece funciones para el formateo de texto, gestión de imágenes, inserción de enlaces, manejo de XHTML, etc. Además, soporta el uso de ASP, ASP.NET, PHP, Java y JavaScript. La versión actual es una versión estable, que puede integrarse fácilmente en cualquier aplicación existente.
La incorporación de la funcionalidad mencionada consiste en modificar la barra superior de herramientas agregándole nuevos botones que permitan insertar el código JavaScript en la página que se está editando.

5. Funcionalidad de la herramienta FCKScorm

Existen en la actualidad algunas herramientas que facilitan este proceso, pero son herramientas propietarias y que necesitan determinados requerimientos para su uso. El objetivo de nuestro proyecto es obtener una herramienta que cumpla con esta funcionalidad bajo la filosofía Open Source. La idea es extender un editor de código HTML de modo que, en el momento mismo de la creación o edición de una página sea posible agregarle las funciones JavaScript necesarias para la comunicación entre el SCO (al que pertenece dicha página) y el LMS. Esta funcionalidad será transparente para un usuario sin conocimientos de programación JavaScript, e inclusive sin conocimientos del lenguaje HTML. De esta forma se le brinda al usuario la posibilidad de incorporarle a las páginas componentes predefinidos orientados a la comunicación.
Los métodos disponibles en principio son los métodos básicos de Sesión y Transferencia de datos, implementados por la API. Con los primeros es posible iniciar y finalizar la sesión de comunicación entre un SCO y el LMS a través de la instancia de la API. Los últimos son utilizados para intercambiar valores del modelo de datos, en este caso los resultados de las evaluaciones.
Para poder plasmar nuestra propuesta en una aplicación real, se construyó una aplicación Web que permite al usuario generar la estructura del curso, crear páginas con o sin el agregado de funciones específicas para la comunicación con un sistema LMS y la posibilidad de importar páginas generadas fuera de la aplicación.
Con el objetivo de acotar el amplio conjunto de funciones SCORM provistas por la API, la aplicación permite diseñar cursos que responden al modelo de cursos propuestos por el CDTC (Centro de Difusão de Tecnologia e Conhecimento) [http://www.iti.gov.br/twiki/bin/view/Swlivre/Cdtc] a través del acuerdo de cooperación entre el ITI (Instituto Nacional de Tecnologia da Informação), Brasil [http://www.iti.gov.br/] y la ONTI (Oficina Nacional de Tecnologías de Información) [http://www.sgp.gov.ar/contenidos/onti/onti.html]. En el marco de este acuerdo la Universidad Nacional de La Plata está participando activamente en el Foro de Responsables Informáticos, donde la ONTI coordina las actividades. Estos cursos abarcan una amplia gama de temas y son utilizados actualmente en forma masiva. La estructura comprende unidades de teoría y una evaluación.
La aplicación se diseñó para ser ejecutada en un servidor Web donde se almacenará toda la estructura del curso y las páginas generadas, con el objetivo de que pudieran ser accedidas desde cualquier máquina conectada al servidor.

Recursos generados con el editor Reload Editor
Editor Personalizado

Con este diseño fue necesario brindarle al usuario la funcionalidad de exportar todo el curso generado con la aplicación fuera del servidor y de esta manera poder crear un paquete SCORM con una herramienta como ReloadEditor.

Moodle

En la figura 1 se muestra un esquema con los pasos a seguir en la construcción de un paquete SCORM; desde la creación de las páginas HTML con nuestro editor hasta la visualización con alguna herramienta LMS.

SCO

Páginas creadas con el editor

SCO

Paquete de SCO

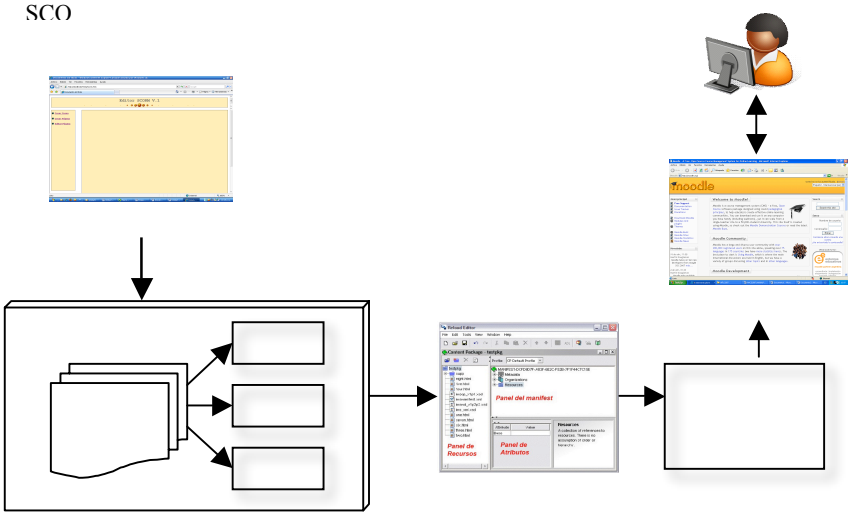

Figura 1. Esquema de creación de un paquete SCORM.

Las funciones provistas por la aplicación comprenden:

- Crear curso: Permite generar la estructura del curso.
- Crear página: Permite crear páginas con o sin el agregado de funciones específicas para la comunicación con un sistema LMS.
- Editar página: Permite abrir páginas ya generadas para poder editarlas.
- Exportar curso: Permite exportar el curso generado fuera del servidor.

6. Caso de estudio: Curso "UML" de CDTC

Como caso práctico de este proyecto se tomó como base el curso UML del proyecto CDTC mencionado anteriormente y se lo construyó utilizando la herramienta implementada.

El contenido del curso original se distribuye en cuatro lecciones: Introducción, Conceptos Iniciales, Diagramas de Estructuras y Diagramas de Comportamiento. En la figura 2 es posible ver la pantalla inicial con el índice del curso a partir del cual se puede acceder a cada lección y a la evaluación final.

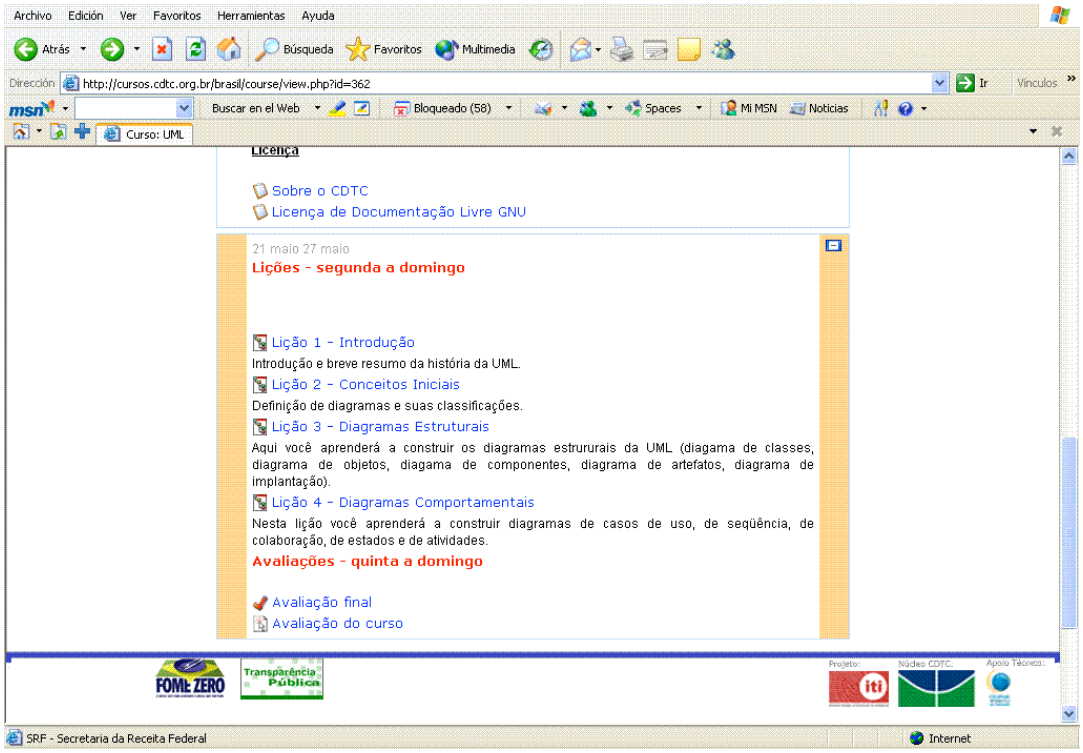

Figura 2. Estructura del curso original UML – ITI Brasil

Partiendo de este curso y utilizando la aplicación desarrollada, mostraremos que es posible reestructurarlo de manera de seguir las normas del estándar SCORM.

La restructuración consistirá en tomar las lecciones del curso existente y crear para cada una, dos Objetos de Aprendizaje (OA) auto-contenidos e independientes del contexto. Uno de los OA corresponderá al contenido propiamente dicho de la lección y el otro a la evaluación de la misma.

Las funciones de comunicación del curso con el LMS son incorporadas en forma automática a través del editor.

Esta forma de construir los cursos permite llegar al mismo producto final que el original, sin agregarle dificultad a la etapa de construcción, pues el usuario "creador" no depende de gente con conocimientos de programación para incorporales un mejor seguimiento del alumno en el curso ya que es posible evaluar cómo fue su actividad, el tiempo total de navegación del curso, la página en que abandonó la navegación, el estado de navegación para cada lección (completa o incompleta) y toda la información relativa a las evaluaciones. Al mismo tiempo que se obtiene material reusable y fácil de mantener.

Todos los cursos se dividen en unidades. En este caso cada lección representará una unidad. Dentro de cada unidad existen dos secciones, la sección Teórica y la sección Práctica, con sus respectivas páginas.

Cada página del curso es creada o editada por el editor FCKeditor SCORM V.1 con el fin de que puedan ejecutar las funciones correspondientes propuestas por ADL, en el archivo SCORMGenericLogic.js para la comunicación con el LMS.

Una vez que el usuario finaliza la creación del curso con el editor deberá proceder a la creación del paquete SCORM y luego a su navegación con un LMS que soporte SCORM.

6.1 Pasos para el armado del curso con FCKScorm

Al acceder a la aplicación se tiene un menú con las opciones de: crear un curso nuevo, crear o editar una página y exportar el curso. A continuación describiremos cada una de ellas.

Crear Curso: se visualizará la ventana que se muestra en la figura 3, solicitándonos algunos parámetros para crear en el servidor la estructura del curso. Los parámetros solicitados son:

- Nombre del curso. En nuestro caso UML.
- Cantidad de unidades que forman nuestro curso. En nuestro caso 4.

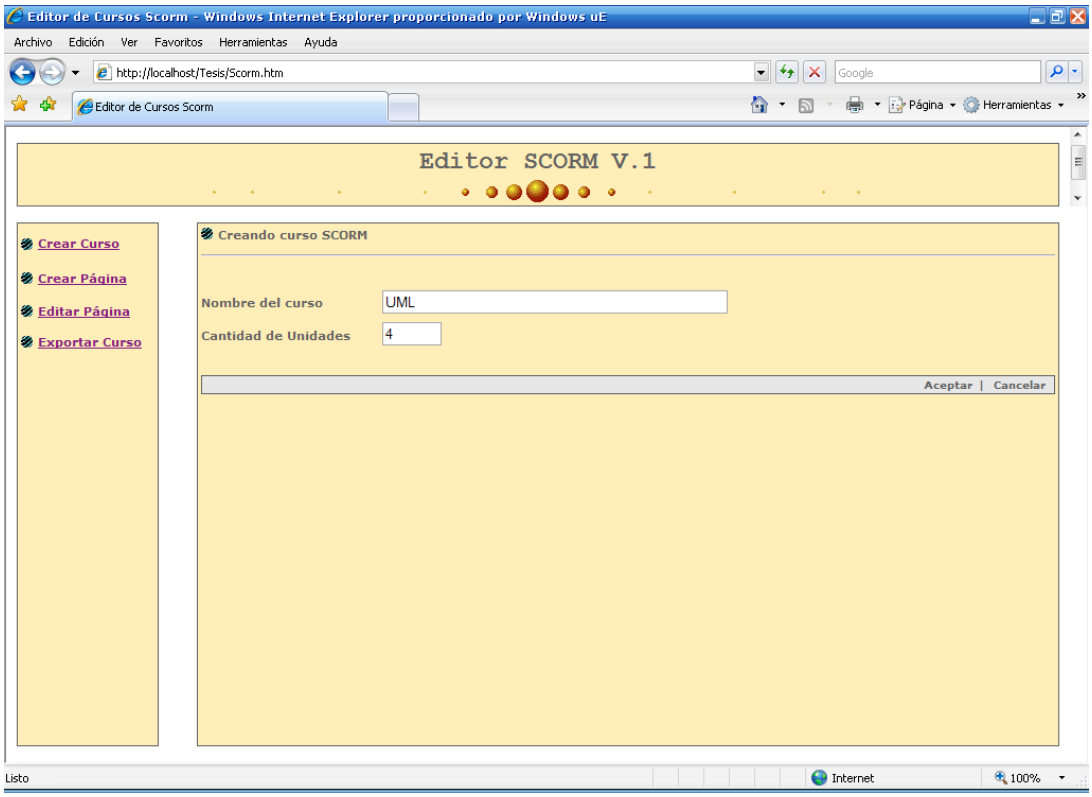

Figura 3. Pantalla de Crear Curso

Al presionar el botón “Aceptar”, automáticamente se crea en el servidor la estructura del curso como se muestra es la figura 3.a:

Figura 3.a. Estructura del curso UML

Al mismo tiempo que se crea la estructura del curso, la aplicación copia automáticamente en cada una de las carpetas "Teoría" de cada unidad el archivo SCORMGenericLogic.js y en las carpetas "Practicas" copia los archivos SCORMGenericLogic.js y SCORMObjectiveLogic.js, que contienen las funciones propuestas por ADL para la comunicación con el LMS y para el manejo de objetivos, respectivamente.

Al generar las carpetas Teoría y Práctica, en cada unidad, la aplicación copia los archivos dummypage.htm, page0.htm y sco.htm desde una ubicación fija en el servidor. Como ya se explicó, todos estos archivos son necesarios tanto para formar un SCO como al momento de generar el paquete SCORM y visualizarlo con algún LMS que lo interprete.

Si el proceso de creación del curso fue exitoso se le presenta al usuario una pantalla donde se le indica *"Se ha creado satisfactoriamente el curso UML"*, en caso contrario se indica *"No se ha podido crear el curso UML".*

Crear Página: una vez creada la estructura del curso debemos crear las páginas de contenidos y evaluaciones para cada unidad. Al seleccionar desde el menú esta opción se presentará al usuario una pantalla solicitando algunos parámetros y luego se abrirá el editor FCKeditor adaptado para comenzar a crear cada una de las páginas.

Los parámetros solicitados son:

- Nombre del curso. En nuestro caso UML.
- Número de unidad al que pertenece la página. En nuestro caso pertenece a la Unidad 1
- Número de página dentro de la unidad. En nuestro caso corresponde a la página 1 de la Unidad 1.
- Seleccionar el tipo de página a crear, es decir, si es una página teórica o de evaluación.

A continuación detallamos por separado la creación de una página teórica y una página de evaluación.

a.- Crear página teórica: Como siguiente paso debemos crear, es decir, rediseñar las páginas teóricas de cada unidad del curso UML Brasil con nuestro editor para permitir la navegación entre las páginas, como se muestra en la figura 4. De esta manera, se podrá registrar el seguimiento del alumno,.

Tipo de página

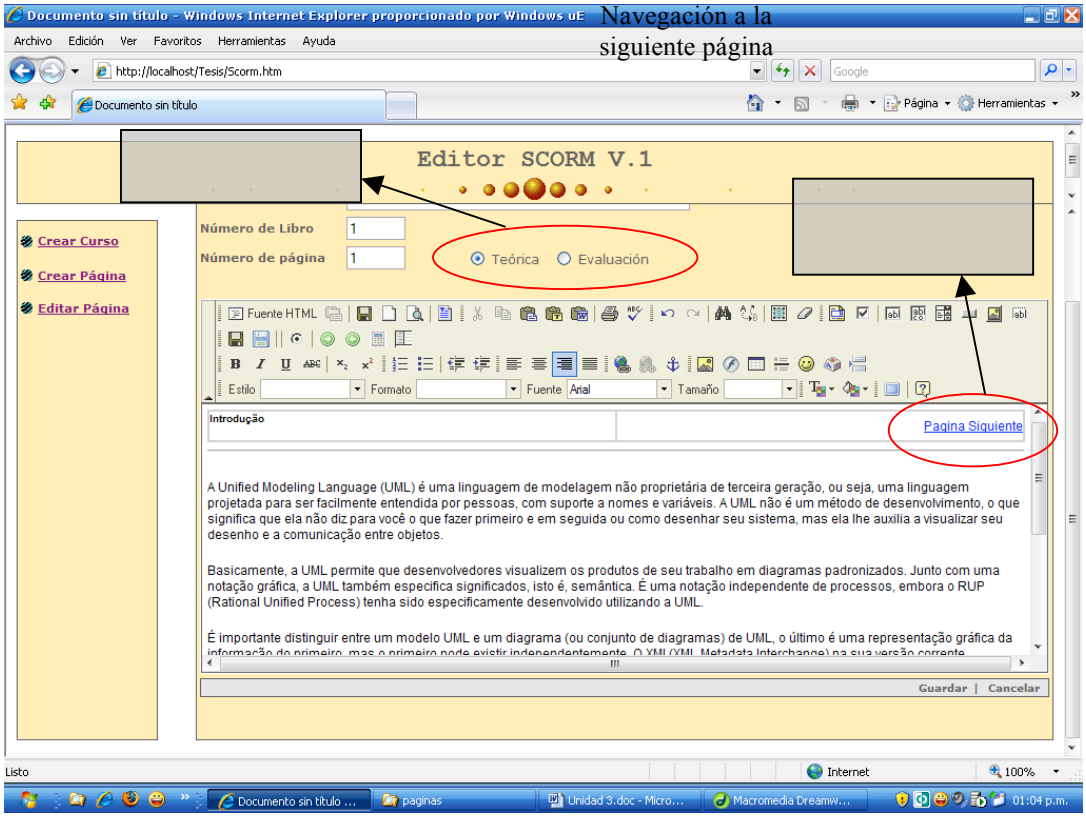

Figura 4. Crear Página

Para esto seleccionamos de la barra de herramientas la opción “Página Siguiente”. Se abrirá la pantalla para la inserción del link a la página siguiente, como se visualiza en la figura 5, solicitando los parámetros: descripción del link, en nuestro caso “Página Siguiente” y número de unidad al que pertenece la página, en este caso la número 1.

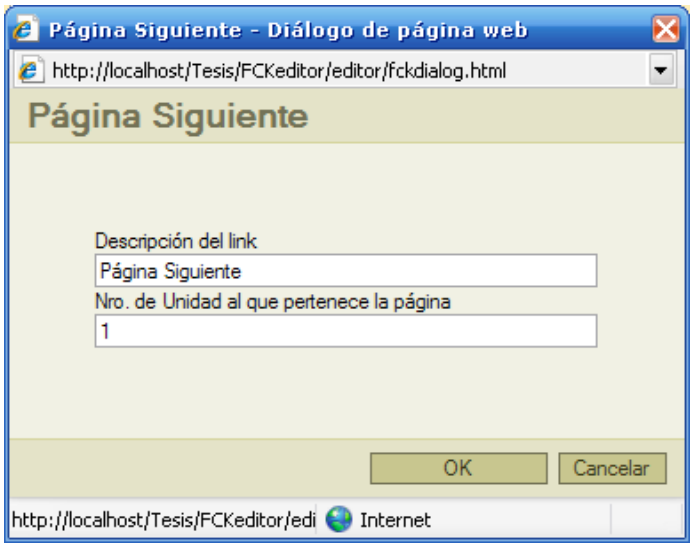

Figura 5. Pantalla correspondiente al botón “PáginaSiguiente” del editor adaptado.

Tener en cuenta que la página descripta en este ejemplo es la primera de la unidad por lo tanto solamente tiene navegación a la página siguiente, en las páginas sucesivas deberíamos agregar también la navegación a la página

anterior con la opción del editor “Página Anterior”. Excepto en la última página que solamente tendrá navegación a la página anterior.

Debemos aclarar que no es la única forma de rearmar las páginas originales pues el usuario “creador” podría diseñarla con otro criterio. Lo más importante es que a cada una de las páginas del curso original le agreguemos los botones “PaginaSiguiente” y “PaginaAnterior” para lograr obtener luego la navegación del alumno por el curso.

Una vez creada la página, seleccionamos “Guardar”. La página será creada en el servidor con el nombre “unidad1_pag1.html” en el path “/UML/unidad1/teoria/”.

Si la creación fue exitosa se visualizará el resultado final de la página creada. En caso contrario se presentará al usuario el texto “La página no se pudo crear.”

a.- Crear página de evaluación: Para rediseñar las páginas de evaluación de cada unidad del curso UML con nuestro editor debemos armar un cuestionario con múltiples opciones y al pie de la página agregar el botón de “Calcular Nota”. Escribimos el texto correspondiente a la página de evaluación de forma tal que quede cada pregunta con sus posibles respuestas. Cada respuesta deberá ser agregada a la página como un “Radio Button”. Para esto seleccionamos de la barra de herramientas la opción “Radio Button”. Se abrirá la pantalla para la inserción de “Radio Buton”, como se visualiza en la figura 6, solicitando los parámetros: nombre del botón, en nuestro caso “Respuesta1”; valor, en nuestro caso las respuestas correctas tendrán un valor de 100 y las incorrectas de 0; el campo “Seleccionado” no debe quedar tildado; el campo “Soporta SCORM” debe quedar tildado y agregamos el valor del último parámetro “Nro. Pregunta” que en este caso es la 1.

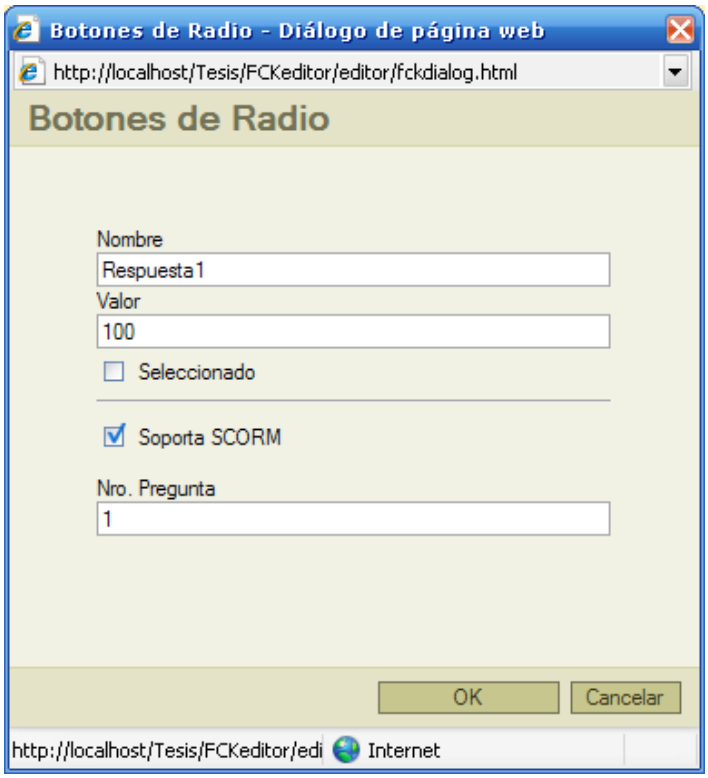

Figura 36. Pantalla correspondiente al botón “Radio button” del editor adaptado.

Al presionar el botón de “OK” automáticamente agrega el “Radio Button” correspondiente con la llamada a la función JavaScript detallada anteriormente. A continuación nos posicionamos al lado del “Radio Button” que se visualiza en la página que estamos creando y escribimos la respuesta correspondiente a ese radio.
Una vez finalizado el cuestionario agregamos el botón “Calcular Nota” al final del cuestionario. Para esto seleccionamos de la barra de herramientas la opción “Calcular Nota”. Se abrirá la pantalla para la inserción del botón, como se visualiza en la figura 7, solicitando los parámetros: nombre, es el nombre que se le asigna a la propiedad name del botón, en nuestro caso “Calcular” y texto, es la descripción que se visualiza en el botón, en nuestro caso “Calcular Nota”.

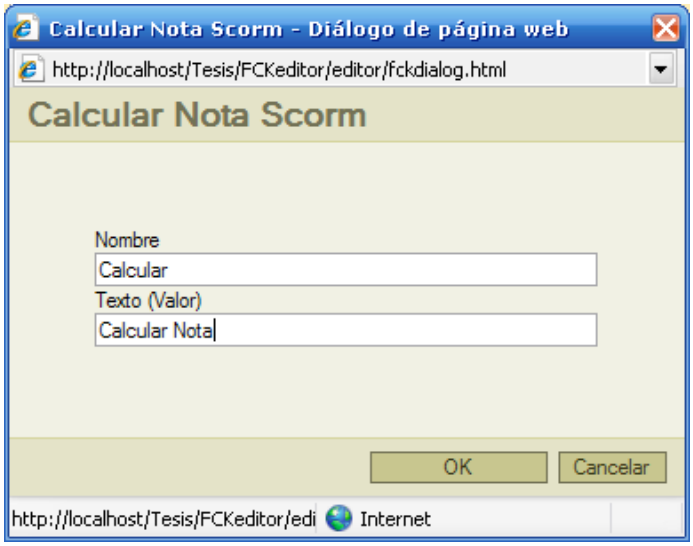

Figura 7. Pantalla correspondiente al botón “Calcular Nota” del editor adaptado.

Al presionar el botón de “OK” automáticamente agrega el botón “Calcular Nota” correspondiente con la llamada a la función JavaScript detallada anteriormente.
Una vez creada la página, seleccionamos “Guardar”. La página será creada en el servidor con el nombre “evaluacion1.html” en el path “/UML/unidad1/practicas/”.
Si la creación fue exitosa se visualizará el resultado final de la página creada, como muestra en la figura 8. En caso contrario se presentará al usuario el texto “La página no se pudo crear.”

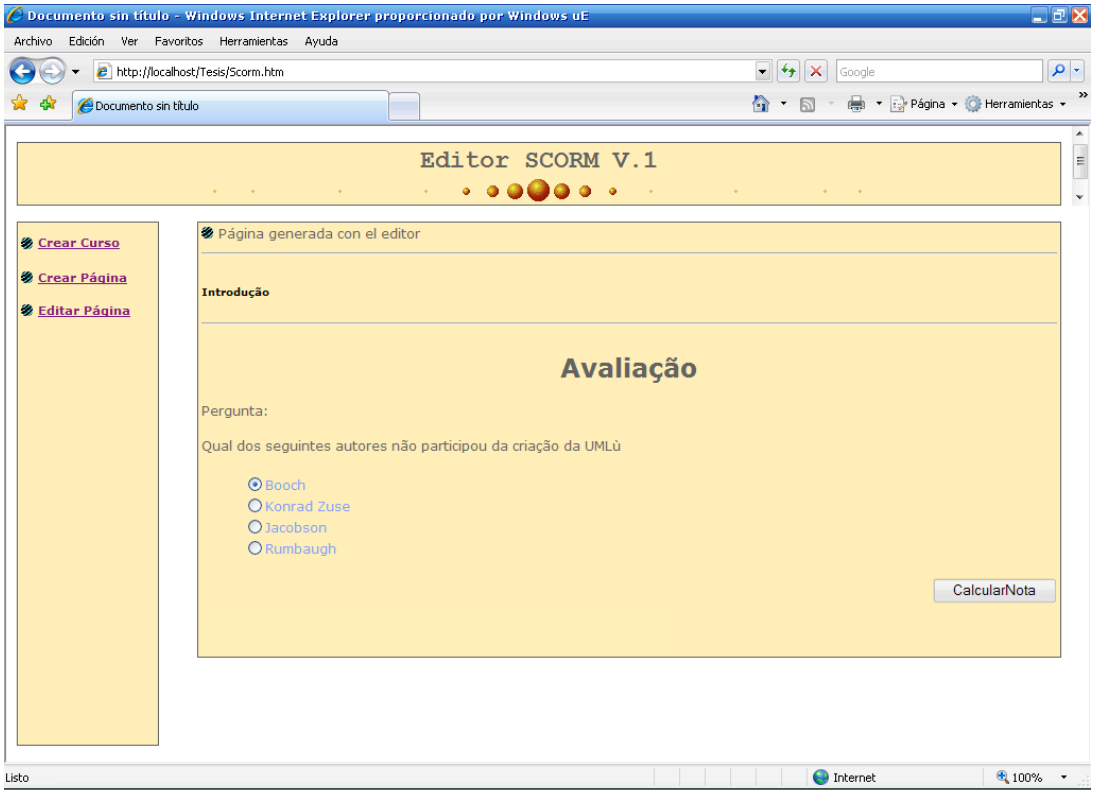

Figura 38. Visualización de una página creada con “FCKScorm”

Editar Página: esta opción permite abrir páginas existentes en la máquina local como se explicó anteriormente. Al seleccionar la opción, se abre una pantalla solicitando que el usuario seleccione la página a editar ingresando la ubicación (path). Una vez seleccionada deberá presionar el botón “Cargar Página” y automáticamente se visualizará la página seleccionada en una pantalla con la misma funcionalidad que en la opción “Crear Página”, pero en este caso el contenido de la página seleccionada se podrá visualizar en el editor FCKeditor para su edición.

Exportar Curso: esta opción permite al usuario obtener en su máquina algún proyecto creado en el servidor. Al seleccionar la opción se abre una pantalla solicitando: el nombre del curso, en nuestro caso UML; la cantidad de unidades, en nuestro caso 4; y la ubicación (path) donde quiere guardar el proyecto en su máquina. Al aceptar los datos ingresados la aplicación genera un archivo “.zip”, es decir, un archivo comprimido con el nombre del curso. Este archivo contiene toda la estructura del curso generado por el usuario y el formato que es usado en una herramienta de creación de paquetes SCORM como el Reload Editor.

Con los pasos descriptos hasta ahora el usuario podrá crear sus cursos con componentes de comunicación SCORM. Luego deberá crear el paquete SCORM utilizando Reload Editor.

6.2 Armado del paquete SCORM con Reload Editor

Una vez creadas las páginas html con el editor extendido “FCKScorm”, se utilizó la herramienta Reload Editor para construir el paquete SCORM.

Al abrir Reload Editor seleccionamos la opción de “Crear Paquete SCORM” en el cual seleccionamos el curso UML que hemos creado y automáticamente se presentará una pantalla con la estructura del mismo como se visualiza en la figura 9.

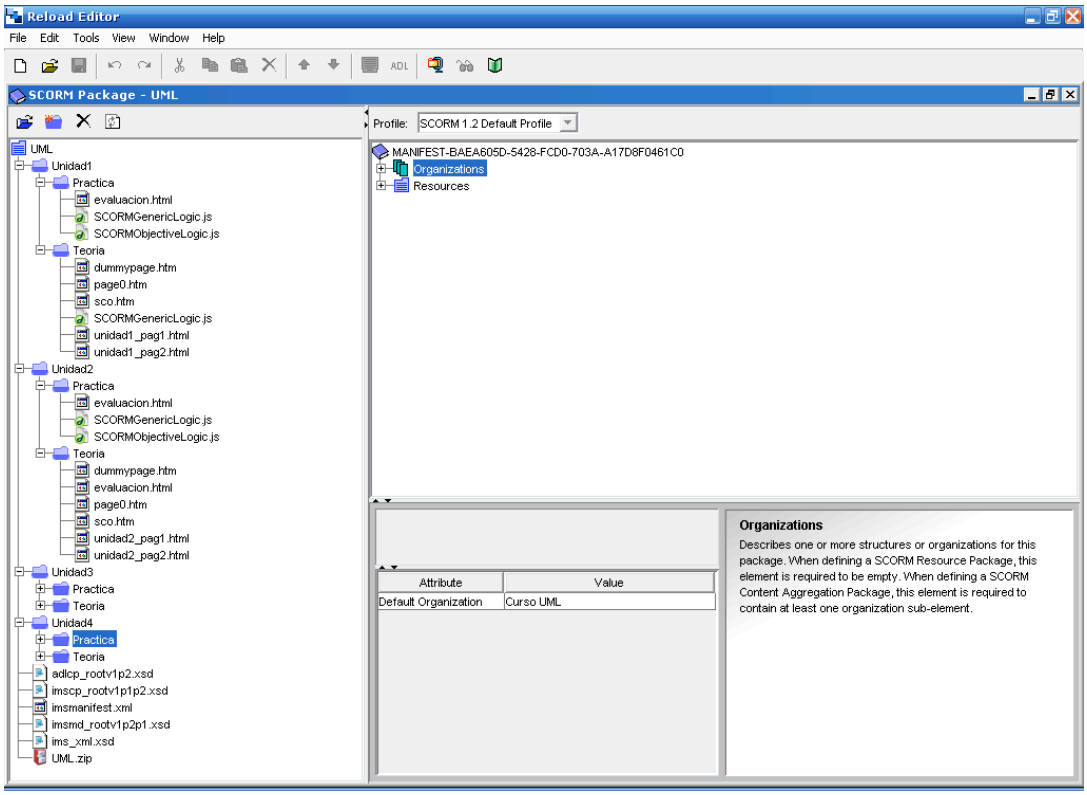

Figura 40. Estructura del curso UML visualizado con Reload Editor

En el margen izquierdo podemos ver la distribución de los html y javaScripts creados. En el margen derecho debemos crear las organizaciones para nuestro curso. Este curso contiene una sola organización para el contenido. Esta organización está dividida en cuatro unidades: Introdução, Conceitos Iniciais, Diagramas Estruturais y Diagramas Comportamentais. Luego dentro de cada sección tenemos Lição y Avaliação con su respectivo acceso a cada módulo. En el margen derecho en la parte inferior podemos ver los recursos utilizados. Para cada unidad tenemos el sco correspondiente a Introdução, Conceitos Iniciais, Diagramas Estruturais y Diagramas Comportamentais, como así también los sco’s correspondiente a las Avaliações. Cada uno de ellos con sus assets.
Una vez finalizado el proceso de definición de las organizaciones y los recursos, utilizamos la función “Zip Content Package” para crear el paquete SCORM del curso.

6.2 Visualización del curso SCORM con Moodle

Como último paso subamos el paquete SCORM creado al Moodle para mostrar cómo el curso UML puede navegarse en un LMS que soporte SCORM.
En un curso de Moodle podemos añadir distintos tipos de actividades. Y SCORM es una más de las que admite este entorno virtual. Lo primero que necesitamos es tener nuestro paquete SCORM en el sistema. Por esto hace falta subir el archivo comprimido a la carpeta de archivos del curso donde queremos tener la actividad SCORM.

Una vez subido el archivo vamos a la opción “Agregar Actividad SCORM”. Con esto se nos presentará una pantalla en la cual debemos seleccionar el archivo comprimido que cargamos anteriormente y completar la información sobre la actividad que estamos creando.
Nuestro curso se presenta como se muestra en la figura 10, aparece el índice y a partir de acá podemos comenzar a navegarlo.

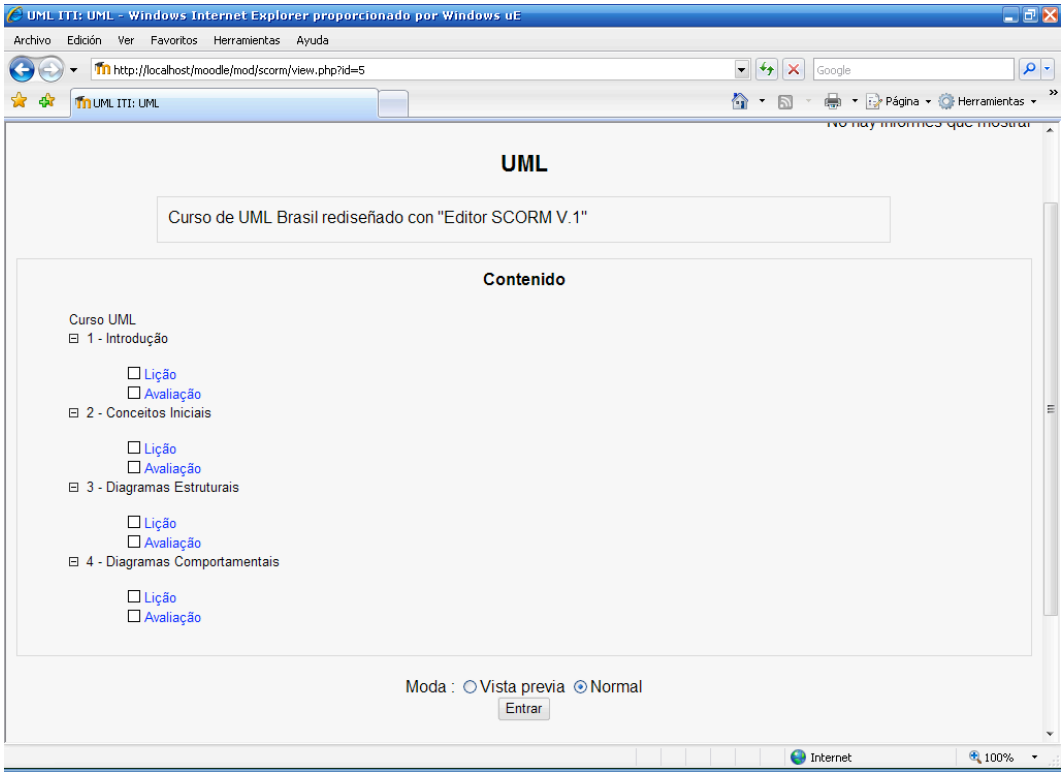

Figura 10. Visualización del curso UML en Moodle

Al seleccionar la unidad “Introdução” y luego la opción “Lição”, vamos a navegar la primer unidad y realizar la evaluación correspondiente.
Si salimos del curso e intentamos navegarlo nuevamente se presenta el índice con algunos cambios como se observa en la figura 11. Vemos que se indica con diferentes íconos el estado de navegación del mismo. Por ejemplo en la unidad 1 “Introdução - Lição” se muestra una luna amarilla, esto significa que se navegó toda la opción “Introdução - Lição” propuesta. En cambio en la opción “Introdução - Avalaição” se muestra un icono de una flecha con lo cual se indica que se comenzó la navegación pero no se culminó.
Podemos ingresar a la sección de informes de la actividad SCORM, y acceder a algún alumno para ver su actividad, esto se muestra en la figura 12.

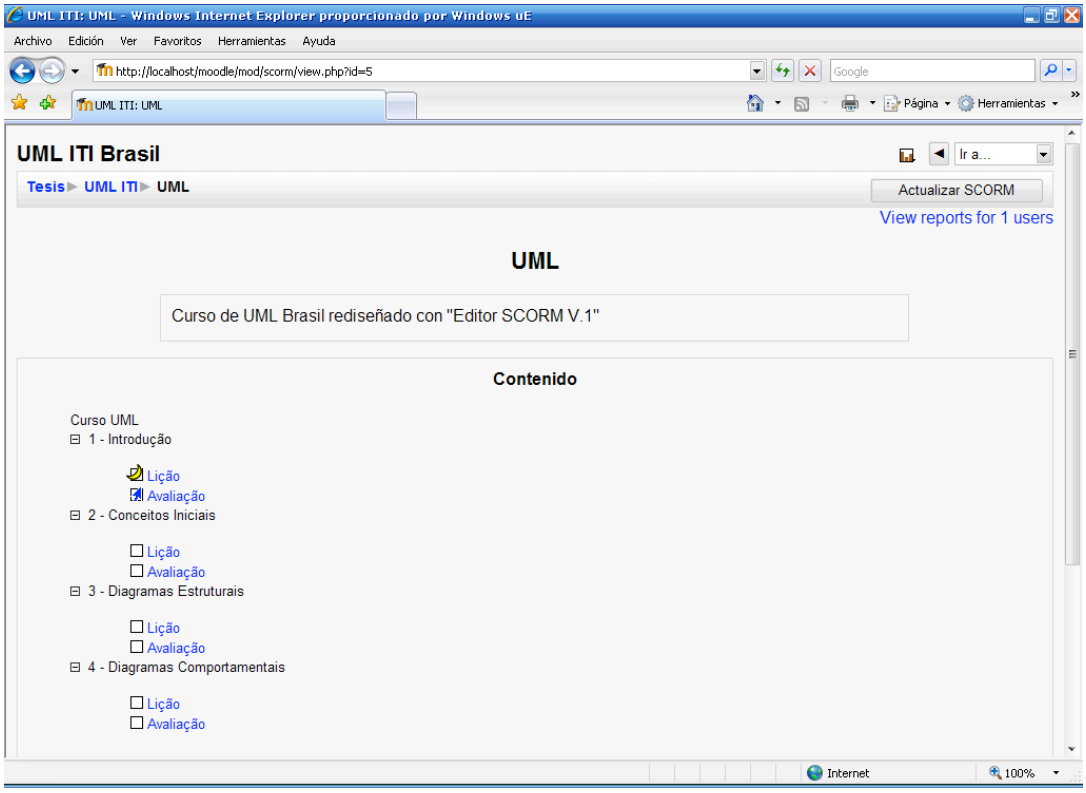

Figura 11. Índice del Curso con el estado de la navegación.

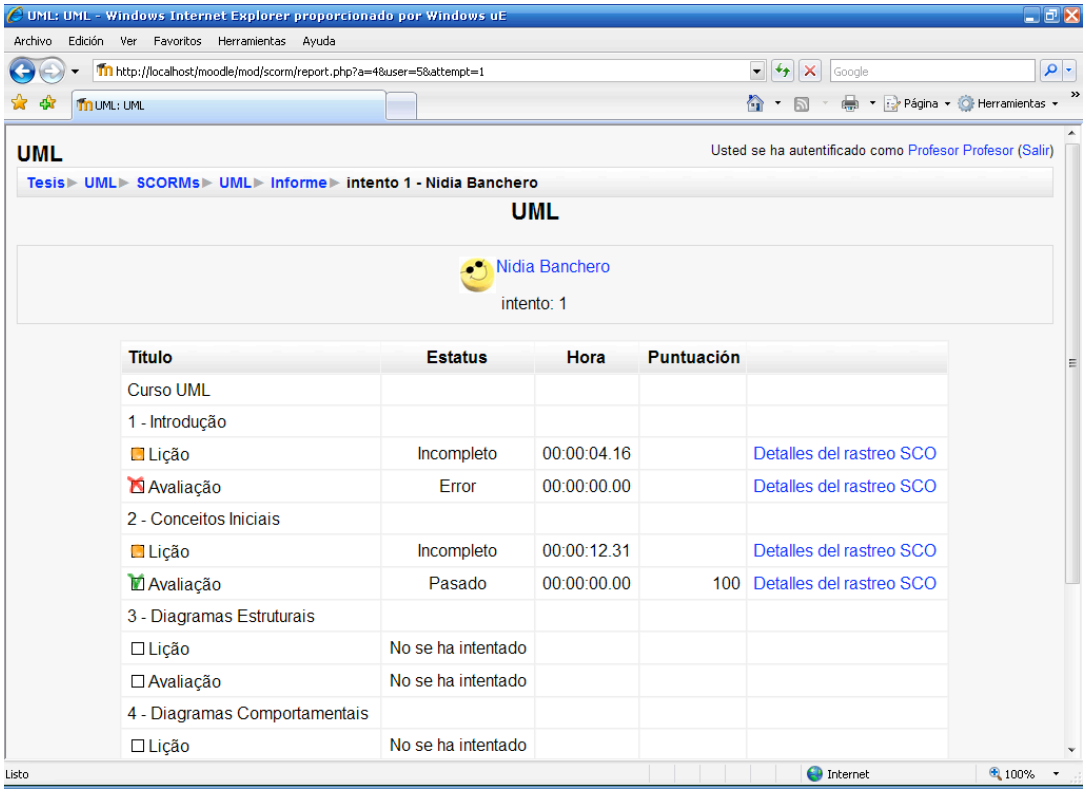

Figura 12. Información de la navegación de un alumno.

Se presenta al profesor una planilla con la información de la navegación donde se indica para cada unidad:

- el estado. Si el tema de la unidad son páginas navegables se indica: Completo, Incompleto y No se ha intentado. En el caso de ser una evaluación se indica: “Error”, si no se logró, “Pasado” si la respuesta fue correcta y “No se ha intentado” si no se navego.
- hora: tiempo total de navegación para el tema.
- puntuación en el caso de ser una evaluación se presenta la puntuación obtenida.
- un link al detalle del rastreo SCO.

Si seleccionamos el link “Detalle del rastreo SCO” podemos ver información más detallada de la navegación del alumno.

Conclusiones

Actualmente la construcción de material educativo que respete un estándar, es de vital importancia para la generación de recursos reusables e interoperables. La herramienta implementada tiene como objetivo facilitar la etapa del proceso de armado de cursos compatibles con SCORM, relacionada con la incorporación de funciones de comunicación a los objetos de aprendizaje.
La aplicación final responde a un modelo de cursos específico de amplia difusión y uso en Brasil, propuestos por el CDTC (Centro de Difusão de Tecnologia e Conhecimento) a través del acuerdo de cooperación con el ITI (Instituto Nacional de Tecnologia da Informação). A partir de este modelo se seleccionó la funcionalidad a incorporar, dentro del abanico de propuestas de la API SCORM.
Una vez seleccionado el modelo de curso y planteada la aplicación se generó un caso de prueba, rearmando un curso tomado del sitio de referencia (ITI) y construyendo las mismas páginas con nuestro editor. Para ello, fue necesario utilizar, además, el software ReloadEditor que nos permitiera construir el paquete SCORM y también el entorno de aprendizaje Moodle para ver como se comportaba el curso generado en un LMS.
Con la herramienta obtenida se favoreció la generación de cursos SCORM, permitiendo de esta manera, que el contenido resulte reusables. Además tiene la ventaja de ser open source, lo que representa una contribución significativa porque no existen actualmente herramientas de estas características bajo esta filosofía. Esta herramienta representa una contribución concreta para el desarrollo de contenido educativo en entornos de software libre, ya que permite generar recursos que respetan el estándar SCORM, lo cual no está soportado por las herramientas actuales.

Referencias

[1] Radoslav, A.; Ganchev, I.; O’Droma, M. (2005). Content Metadata Application and Packaging Service (CMAPS) – Innovative Framework for Producing SCORM-compliant e-Learning Content. *Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies (ICALT´05)*, Kaohsiung, Taiwan, July 5, 2005 pp. 274-278.

[2] González Barbone, V. (2006). Creación de Cuestionarios en Objetos de Aprendizaje Reutilizables. *Proceedings of Congreso Regional de Telecomunicaciones, 1.* MVD Telcom. Montevideo, Uruguay.

[3] Carnegie Mellon (2004). *Best practices Guide for Content Developers.* Carnegie Mellon, Learning Systems Architecture Lab. 1st. edition including updates to Section 7, dated 2004-04-05. http://www.lsal.cmu.edu/lsal/expertise/projects/developersguide/

[4] Forth, S.; Childs, E. (2003). *White Paper on e-Learning Specifications and Standards*, Recombo Inc, Innovative Training Solutions Inc.

[5] Díaz, J.; Schiavoni, A.; Osorio, A.; Amadeo, P. (2006). Standardizing educative content with SCORM: a case study, *Proceedings of 17th International Conference of Society for Information Technology and Teacher Education (SITE)*, Association for the Advancement of Computing in Education (AACE), Orlando, Florida, March 20th, 2006.

[6] Jun-Ming Su (2005). An Object based Authoring Tool for Creating SCORM Compliant Course. *Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05)*, March 28th, 2005, Vol. 1, pp 209-214.

[7] Pisel, K.; Lindsey, A. (2004). Developing SCORM-Compliant Media-Rich Graduate-Level Distance Education: A Case Study of Best Practices, *Proceedings of the 20th Annual Conference on Distance Teaching and Learning*, August 4-6, 2004, Madison, Wisconsin, Monona Terrace Convention Center. Available online at http://www.uwex.edu/disted/conference/Resource_library/proceedings/04_1333.pdf.