

# Extendiendo un editor de software libre para soportar contenido SCORM

Javier Díaz<sup>1</sup>, Alejandra Schiavoni<sup>1</sup>  
Nidia Banchero<sup>1</sup>

<sup>1</sup> LINTI – Laboratorio de Investigación en Nuevas Tecnologías Informáticas  
Calle 50 y 120, 2do Piso  
La Plata, Argentina  
[jdiaz@unlp.edu.ar](mailto:jdiaz@unlp.edu.ar), [ales@info.unlp.edu.ar](mailto:ales@info.unlp.edu.ar), [n\\_banchero@yahoo.com.ar](mailto:n_banchero@yahoo.com.ar)

**Resumen.** La educación ha tenido un proceso de evolución a la par de las nuevas tecnologías que fueron surgiendo estos últimos años. La aparición de los numerosos sistemas y recursos educativos, ha llevado a establecer estándares que permitan su uso eficiente. La utilización de SCORM en la creación de contenido educativo requiere una serie de pasos y el conocimiento de los detalles de implementación del estándar, para la identificación de los OA y la incorporación de las funciones de comunicación con el LMS dentro de ellos. Este artículo propone una herramienta basada en Software Libre para facilitar la tarea de construcción de material educativo compatible con SCORM que cuente con componentes de comunicación. La herramienta extiende la funcionalidad de un editor HTML permitiendo que se agreguen a las páginas las funciones de interacción con el LMS.

**Keywords:** Objetos de aprendizaje, SCORM, Comunicación con LMS

## 1 Introducción

El proceso educativo fue evolucionando a la par del desarrollo de las tecnologías de la información. La educación a distancia o lo que también se conoce como e-learning es hoy en día una técnica de enseñanza-aprendizaje muy usada. 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]. En la evolución de los diferentes estándares surgidos, se produjo un proceso de convergencia hacia ADL SCORM (Sharable Content Object Reference Model) [2], el cual integra IMS [3], IEEE LTSC LOM [4] y AICC [5] y que hoy en día aparece como el estándar internacional más conocido.

La construcción de cursos basados en un estándar de e-learning resulta de suma importancia para la generación de nuevos tipos de material educativo más robustos y poderosos [6]. Sin embargo, habiendo transcurrido varios años desde la aparición de

los estándares, 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 [7].

En la etapa de construcción del material educativo, normalmente se utilizan editores HTML. Estos editores, ya sean versiones comerciales o de software libre, si bien permiten crear de manera sencilla un curso de aprendizaje, no brindan la posibilidad de generar contenido educativo que respete el estándar SCORM.

La mayor complejidad en el proceso de construcción de contenido educativo compatible con SCORM, está dada en aprovechar la funcionalidad de comunicación entre el LMS (Learning Management Systems) y el contenido a través de la utilización de las funciones JavaScript. Esta comunicación resulta de gran utilidad cuando se desea registrar la actividad del alumno dentro del curso. La incorporación de estas funciones requiere conocer los detalles de implementación y el modelo de datos del estándar.

Este artículo describe una herramienta que facilita la creación de contenido SCORM disminuyendo el nivel de complejidad del proceso de agregar metadatos y funciones de comunicación a las páginas HTML que forman parte de un curso. De esta manera, es posible generar contenido portable entre cualquier sistema LMS compatible con SCORM.

Para lograr el objetivo mostramos cómo se puede extender un editor de código HTML (Open Source), para que permita agregar las funciones JavaScript utilizadas en la comunicación entre un paquete SCORM y un LMS en el momento de la creación o edición de una página.

## **2 Comunicación entre objetos de aprendizaje y LMS utilizando 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 [8]. Para ello propone el uso de *objetos de aprendizaje compartidos* (SCOs, Shareable Content Objects) que pueden incluir objetivos, autoevaluaciones, simulaciones, etc. Un objeto de aprendizaje tiene un claro propósito educativo y es una estructura autónoma que contiene un objetivo general y actividades de aprendizaje.

Un LMS es un entorno de ejecución capaz de mantener información sobre el usuario, lanzar objetos educativos y comunicarse con ellos, interpretando instrucciones sobre secuenciación entre distintos objetos. En este sentido, 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 [9]. Por ejemplo, al usar SCORM toda la información relativa al estudiante es almacenada en el LMS y no dentro del material educativo.

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.

En el modelo SCORM, es el contenido el que inicia y termina la comunicación informándolo al LMS. Para ello, existen dos funciones de la API que gestionan la inicialización y finalización de un SCO, que son LMSInitialize("") y LMSFinish("") y su uso es obligatorio. El resto de las funciones de la API son usadas para la transferencia de datos desde y hacia el LMS y no son obligatorias. Su finalidad es comunicar datos tales como el nombre del alumno, la puntuación obtenida en una evaluación o el tiempo empleado en el curso.

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 garantizar que la información sobre el SCO pueda ser interpretada por diferentes LMS's de manera uniforme. Por ejemplo, si 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.

La fase de desarrollo requiere experiencia técnica para cumplir las pautas de conformidad del modelo de referencia, es decir que los programadores deben tener los conocimientos básicos de la tecnología requerida para aplicar SCORM. Normalmente, la persona que construye los contenidos de un curso, debe agregar en forma manual las llamadas a estas funciones Javascript, editando el código HTML. En el artículo "*Standardizing Educative Content with SCORM: A Case Study*", se muestra en forma clara la construcción de un curso bajo el estándar SCORM y la información intercambiada con el LMS [10]. Con la herramienta propuesta estas invocaciones son agregadas a las páginas que se está creando, en forma automática y transparente al usuario.

### **3 Funcionalidad de la herramienta**

Si bien la construcción y visualización de un paquete SCORM son tareas relativamente simples, resueltas hoy por herramientas Open Source tales como ReloadEditor [11] y Moodle [12], como mencionamos anteriormente 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.

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.

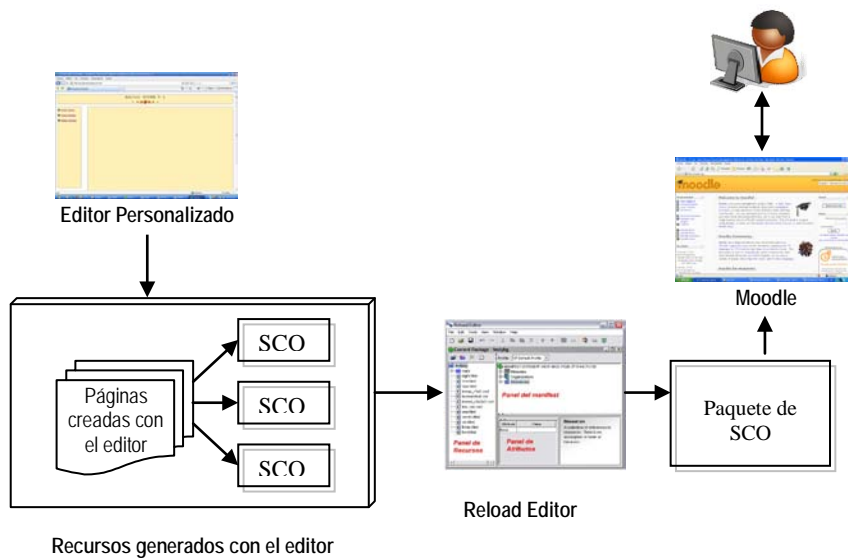
Para poder plasmar nuestra propuesta en una aplicación real, se construyó una aplicación Web que permite al usuario generar la estructura del curso, crear páginas con o sin el agregado de funciones específicas para la comunicación con un sistema LMS y la posibilidad de importar páginas generadas fuera 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) [13] a través del acuerdo de cooperación con el ITI (Instituto Nacional de Tecnologia da Informação), Brasil [14]. 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.

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.

En la Fig. 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.



**Fig 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.

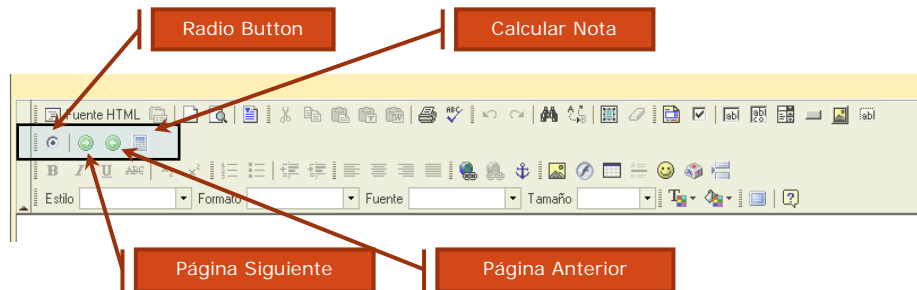
#### 4 Implementación de FCKeditor para SCORM

Para poder construir esta herramienta en primer lugar se adaptó el editor FCKeditor [15] de forma tal que permita construir fácilmente cursos de enseñanza que se adapten a la estructura “Teoría – Evaluación”, respetando el modelo de curso seleccionado.

Para las páginas que forman parte de la Teoría del curso, el usuario puede utilizar los botones “Página Anterior” y “Página Siguiente” para agregarle las funciones que permiten registrar la navegación del alumno en el LMS. En tanto que para la página de evaluación, el usuario puede utilizar los botones “Radio Button” para armar el cuestionario de selección múltiple y “Calcular Nota” para agregarle las funciones que permiten registrar los resultados de las evaluaciones en el LMS.

Para adaptar el editor FCKeditor a nuestras necesidades se definió una nueva barra icónica para el menú, como se muestra en la Fig. 2, donde:

- Se mantuvieron las funciones básicas de cualquier editor HTML.
- Se agregaron nuevos botones que permiten la incorporación de las funciones de navegación dentro de las páginas y las de comunicación de un curso SCORM con el LMS.



**Fig. 2.** Barrar icónica SCORM del FCKEditor

En la construcción de la nueva barra icónica se realizaron las siguientes tareas:

- Se definió una nueva barra icónica bajo el nombre SCORM en el archivo de configuración (fckconfig.js).
- Se definieron las imágenes correspondientes a cada nuevo botón estableciendo un skin “SCORM” que respeta la estructura de archivos de FCKeditor.
- Se definieron los tooltip correspondientes a cada botón en el archivo de lenguaje (es.js).

El archivo **fckeditorcode.js** es el más importante del editor FCKeditor porque en él se implementan las distintas funcionalidades de cada uno de los botones de la barra que se muestra al instanciarlo.

Las funciones del editor que fueron modificadas son:

- La función “FCKToolbarItems.GetItem” que asocia a cada botón de la barra su correspondiente imagen y ToolTip, para que instancie a la función “FCKToolbarButton” para cada nuevo botón.
- La función “FCKCommands.GetCommand”, que es ejecutada cuando se presiona un botón de la barra icónica, para agregar la funcionalidad de los botones nuevos correspondientes a SCORM.

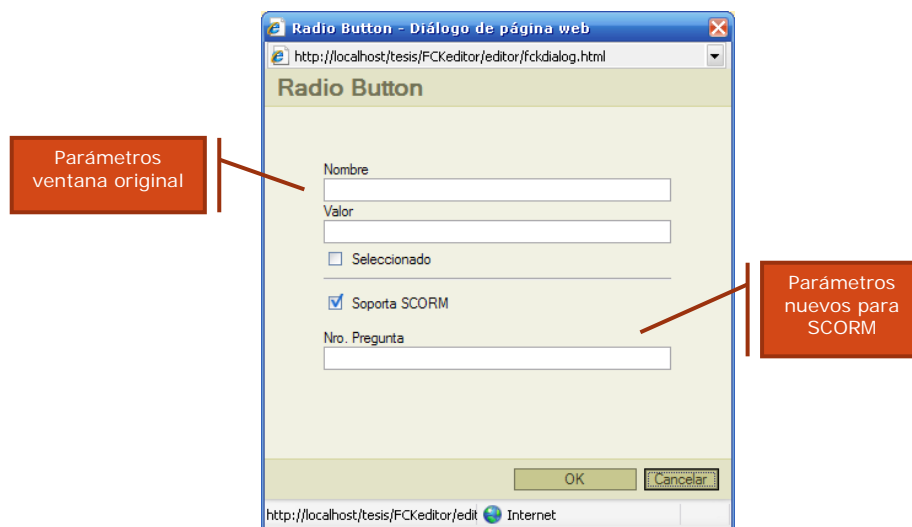
Los eventos que disparan los botones con funcionalidad SCORM incorporados, abren ventanas de diálogo solicitando ciertos parámetros requeridos para la invocación de la función JavaScript correspondiente. Después del ingreso de los datos internamente se genera el tag HTML correspondiente al botón presionado, con la llamada a la función de la API para registrar la comunicación con el LMS. Este tag es insertado en la página que se está generando con el editor.

#### 4.1 Funcionalidades de los botones correspondientes a SCORM

**Botón “Radio Button”**. Este botón se puede utilizar para agregar tanto un botón de radio “normal” como uno con contenido SCORM.

Para nuestro modelo de curso es posible usar este botón para las páginas de evaluación, en donde el usuario debe seleccionar la opción SCORM al crear un cuestionario con preguntas y posibles valores de respuesta para cada caso. De esta manera, el SCO de la evaluación puede comunicar los resultados obtenidos por el alumno al sistema LMS.

Al presionar este botón desde la barra del editor, se abre una ventana de diálogo solicitando ciertos parámetros como se visualiza en la Fig. 3.



**Fig. 3.** Ventana de diálogo del botón “Radio Button”

Esta ventana fue redefinida para soportar funcionalidad SCORM. Es una página HTML que recibe ciertos parámetros y en el caso de que soporte SCORM dispara la llamada a la función correspondiente de la API SCORM. .

A continuación se muestra el código generado internamente:

➤ Radio button normal:

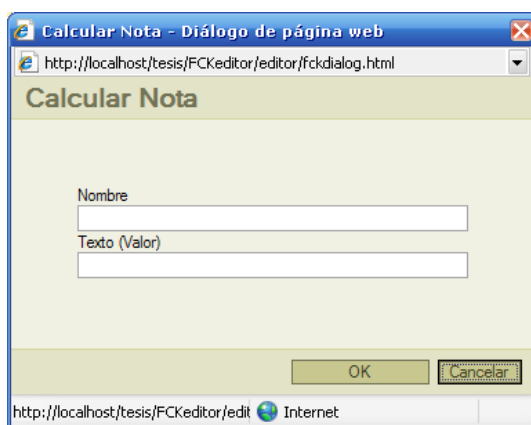
```
<p><input type="radio" name="Repuesta1" value="0" />  
Repuesta 1</p>
```

➤ Radio button con contenido Scorm:

```
<p><input type="radio" name="Repuesta2" value="0"  
onclick="JavaScript:EvalMultipleChoiceItem(this,1)" />  
Respuesta 2</p>
```

**Botón “Calcular Nota”.** Este botón se puede utilizar cuando se desea registrar en el LMS la nota obtenida por el alumno en la evaluación. En nuestro modelo de curso, este botón es utilizado únicamente en las páginas correspondientes a la sección Evaluación.

Al presionar este botón desde la barra del editor se abre una ventana de diálogo solicitando ciertos parámetros como se visualiza en la Fig. 4.



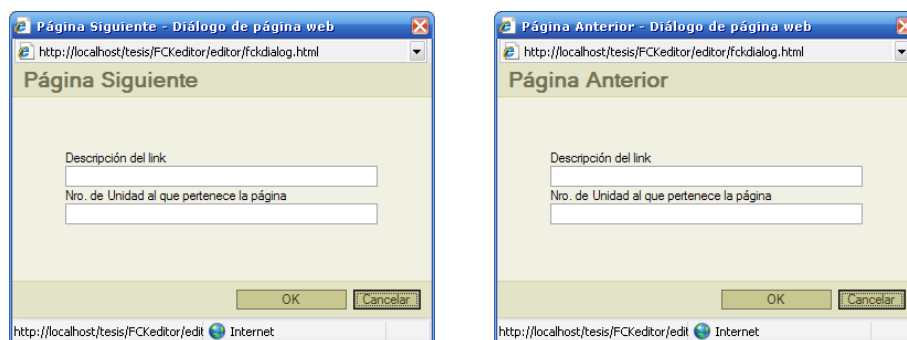
**Fig. 4.** Ventana de diálogo del botón “Calcular Nota”

Al aceptar los datos ingresados se invoca a la función JavaScript que internamente genera el siguiente código:

```
<input type="button" name="Calcular" value="Calcular Nota" onclick="ShowSCORMstatus()" />
```

**Botones “Página Siguiente” y “Página Anterior”.** Estos botones permiten navegar las páginas que componen una unidad registrando la secuencia de navegación en el LMS. En nuestro modelo de curso, estos botones son utilizados únicamente en las páginas correspondientes a la sección Teoría.

Al presionar este botón desde la barra del editor se abre una ventana de diálogo solicitando ciertos parámetros como se visualiza en la Fig. 5.





### Figura 5. Ventanas de diálogo de los botones de navegación

Al aceptar los datos ingresados se invoca a la función JavaScript que internamente genera el siguiente código:

```
<a href="JavaScript:window.parent.PreviousPage(1)">
    Página Anterior</a>

<a href="JavaScript:window.parent.NextPage(1)">
    Página Siguiete</a>
```

## 5 Conclusiones

La posibilidad de disponer de material instruccional compatible con SCORM y que haga uso de toda la funcionalidad provista por el estándar requiere de una serie de pasos que involucran el conocimiento detallado de la implementación del estándar y de un conjunto de tecnologías que resultan difíciles para autores no especializados. La herramienta propuesta tiene como objetivo facilitar una de las etapas del proceso de creación de contenido SCORM, relacionada con la implementación de la comunicación entre los objetos de aprendizaje y el LMS. Está basada en software libre, lo que representa un aporte significativo, dada la escasa existencia de esta clase de herramientas bajo esta filosofía. Debido a que existe una brecha entre la definición del modelo SCORM y el proceso global de desarrollo de cursos por parte de docentes que construyen material educativo, el uso de esta herramienta puede resultar una ayuda considerable para que puedan utilizar el estándar SCORM más fácilmente. La herramienta fue diseñada respondiendo a un modelo de cursos de amplia difusión y uso en Brasil, lo que permite una mayor contribución en el acuerdo de cooperación entre ambos países. Esta herramienta sirve como punto de partida en el proceso de atacar las tareas que resultan más complejas al momento de la construcción de material educativo compatible con SCORM. Siguiendo con esta línea de investigación, las próximas etapas consisten en adaptar y ampliar la herramienta implementada para un universo mayor de cursos con el objetivo de propiciar la construcción y uso de contenido e-learning.

## Referencias

1. Radoslav, A., Ganchev, I., O'Droma, M.: Content Metadata Application and Packaging Service (CMAPS) – Innovative Framework for Producing SCORM-compliant e-Learning Content. In: Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies (ICALT'05), Kaohsiung, Taiwan, July 5, 2005 pp. 274-278 (2005).
2. ADL SCORM – Advanced Distributed Learning, Sharable Content Object Reference Model, <http://www.adlnet.org/>
3. IMS Global Learning Consortium, <http://www.imsglobal.org/learningdesign/>
4. IEEE LTSC LOM Learning Object Metadata, <http://ltsc.ieee.org/wg12/>
5. AICC Aviation Industry CBT Committee, <http://www.aicc.org>

6. Jones, E.: Implications of SCORM and Emerging E-learning Standards On Engineering Education. In: Proceedings of the 2002 ASEE Gulf-Southwest Annual Conference. The University of Louisiana at Lafayette, March 20 – 22, 2002. Available online at <http://falcon.tamucc.edu/~ejones/papers/ASEE02.pdf> (2002).
7. González Barbone, V.: Creación de Cuestionarios en Objetos de Aprendizaje Reutilizables. In: Proceedings of Congreso Regional de Telecomunicaciones, 1. MVD Telcom. Montevideo, Uruguay (2006).
8. Carnegie Mellon: Best practices Guide for Content Developers. Learning Systems Architecture Lab at Carnegie Mellon University. 1st. edition including updates to Section 7, dated 2004-04-05. (2004).
9. Forth, S., Childs, E.: White Paper on e-Learning Specifications and Standards, Recombo Inc, Innovative Training Solutions Inc. (2003)
10. Díaz, J., Schiavoni, A., Osorio, A., Amadeo, P.: Standardizing Educative Content with SCORM: A Case Study. In: Proceedings of Society for Information Technology and Teacher Education International Conference, pp. 318-323, C. Crawford et al. (Eds.) Chesapeake, VA: AACE, Orlando, Florida, USA, March 19 (2006).
11. RELOAD Reusable eLearning Object Authoring & Delivery, <http://www.reload.ac.uk/>
12. Moodle, <http://moodle.org>
13. CDTC Centro de Difusão de Tecnologia e Conhecimento <http://www.iti.gov.br/twiki/bin/view/Swlivre/Cdte>
14. ITI Instituto Nacional de Tecnologia da Informação, <http://www.iti.gov.br/>
15. FCKeditor, <http://www.fckeditor.net/>