

# CREACIÓN DINÁMICA DE EDICIONES DOCUMENTALES CON REPOSITORIO DE DATOS EN LENGUAJE XML

Daniel Garzón Rubio<sup>1</sup> y Benjamín Ramos Álvarez<sup>2</sup>

<sup>1</sup> Grupo de Seguridad de las TI, Universidad Carlos III de Madrid  
[dgarzon@inf.uc3m.es](mailto:dgarzon@inf.uc3m.es)

<sup>2</sup> Grupo de Seguridad de las TI, Universidad Carlos III de Madrid  
[benja1@inf.uc3m.es](mailto:benja1@inf.uc3m.es)

**ABSTRACT:** Actualmente, las empresas, Universidades, Escuelas de Negocios... en cada uno de sus congresos, jornadas, etc. suelen crear ediciones en soporte electrónico con la descripción de las mismas, detalles de los ponentes, contenidos de las ponencias, etc. En el artículo se presenta una propuesta para automatizar el proceso de creación de ediciones. Se pretende que desde una aplicación maestra sea relativamente fácil la creación de ediciones. La Herramienta dinámica de creación de ediciones por medio de archivos de configuración permite generar cada nueva aplicación con el fin de ser distribuida en soporte físico tipo CD, pendrive... Dado el cariz multimedia de la aplicación, la tecnología de desarrollo utilizada ha sido una herramienta de autor. Buscando por un lado la máxima portabilidad y por otro la capacidad de comunicación con otras herramientas y bases de datos, los archivos de configuración se han especificado en el lenguaje universal de marcado XML, actuando éstos como repositorio de datos.

**Keywords:** multimedia, XML, estandarización, e-learning, lenguaje de marcado.

## 1. INTRODUCCIÓN

El presente trabajo de investigación y su representación final en una herramienta, pretende resolver primero conceptualmente y posteriormente mediante una implementación, el problema de automatización de un proceso de difusión de documentación, acercando la tecnología y la posibilitando a usuarios no tecnólogos el desarrollo de aplicaciones de edición y repositorio documental.

El objetivo principal del proyecto es automatizar el proceso de creación de CDs de ediciones. De manera que desde una aplicación maestra sea sencillo el crear cada una de estas ediciones. Cada una de estas ediciones consistirá, entonces, en la integración de los contenidos de un evento sobre una base (aplicación) estándar.

El concepto evento es una abstracción de repositorio de información. Así, todo tipo de actos en los que se presente información a un usuario final como por ejemplo, congresos, simposios, jornadas, etc. o situaciones en las que se desee entregar, recopilar, almacenar de manera estructurada documentación, estarían englobados en el término abstracto evento.

Otro de los objetivos es que el presente desarrollo sirva como gestor de información de manera asíncrona y distributable en soporte físico. De esta forma, se evitan las necesidades de servidores y recursos de red para su almacenamiento. La herramienta se convierte así en un modo de

estructurar y presentar información, a modo de gestor documental *off-line*.

Con esta arquitectura se pretende disponer de una aplicación maestra (Meta-aplicación) que sirva como soporte para crear tantas ediciones como se desee, en las que únicamente cambien los contenidos, y en la que se mantengan fijos la estructura, diseño base, navegación e interactividad.

La personalización de cada una de las ediciones viene dada por las descripciones de la información en forma de texto, imágenes, animaciones y otros recursos documentales en formatos compatibles como PDF, PPT, DOC, XLS, etc.

El resto del documento se estructura de la siguiente forma. En la Sección 2 se revisa a alto nivel (visión del editor) la manera en la que se generan las ediciones partiendo de la aplicación máster. En la sección 3 se especifican cuáles han sido los principales requisitos de usuario que se han tenido en cuenta en el desarrollo. La sección 4 muestra a grandes rasgos el diseño de la aplicación, en cuanto a estructura y modo de navegación sobre la información. En la sección 5 se examinan de manera general, los componentes y módulos de los que consta la aplicación, es decir, la arquitectura sobre la que se ha realizado el desarrollo. La sección 6 revisa los aspectos relacionados con el diseño, implementación y uso de los archivos de configuración realizados sobre XML. Finalmente, la sección 7 cierra el artículo con las conclusiones obtenidas.

## 2. ESQUEMA GENERAL

La aplicación que se desarrolla en este proyecto es de naturaleza dinámica, por lo que requiere que exista una sistemática relativamente sencilla para que la configuración de cada ediciones que se deseen crear sea relativamente simple.

El objetivo con esta arquitectura es tener una aplicación maestra (Meta-aplicación) que sirva como soporte para crear tantas ediciones como se desee, en las que únicamente cambien los contenidos, y en la que se mantengan fijos la estructura, el diseño base, la navegación y la interactividad.

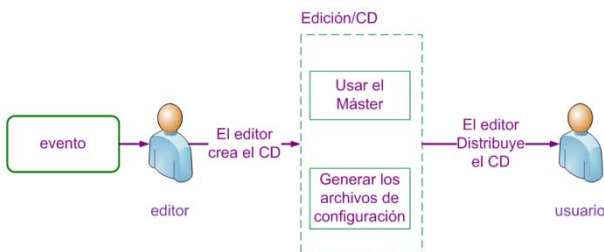


Fig. 1. Proceso de creación de una edición

De esta forma se dispone de una aplicación que con la que pueden generarse múltiples ediciones distintas. Por ejemplo una edición para un congreso de cirugía cardiovascular con tres jornadas con 20 ponentes cada una, o una edición para un curso de fotografía sonido y vídeo.

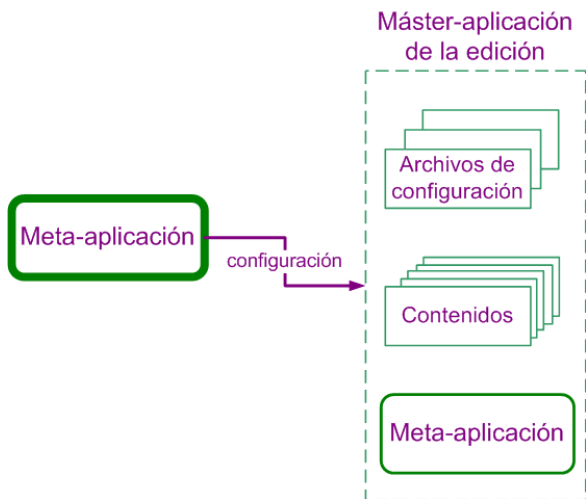


Fig. 2. Esquema general de creación de una edición

Para poder crear una edición se dispone de la meta-aplicación, que permitirá la inclusión de los contenidos y generará los archivos de configuración necesarios.

En estos archivos, que son fácilmente editables, se deberán introducir los contenidos y las rutas de acceso a los mismos. Posteriormente serán recogidos y reconocidos por la aplicación para poder ser mostrados por pantalla.

## 3. PREMISAS DE DESARROLLO

En cuanto al estudio y desarrollo de la implementación, se ha buscado la máxima adaptación a un perfil de editor no tecnológico, de manera que las operaciones a realizar para crear cada una de las ediciones finales fueran simples. Así, se ha intentado seguir una línea de diseño y ejecución fundamentada en las siguientes premisas:

- **Mantenibilidad:** Que se desarrolle por componentes y que éstos sean fácilmente integrables y actualizables.
- **Usabilidad:** Que el producto sea fácil de manejar, intuitivo y amigable para los usuarios que deben utilizarlo. De esta manera, el diseño y las recomendaciones de usabilidad en vigor han sido tenidas en cuenta en durante el desarrollo.
- **Distribución:** Que por medio del desarrollo modular (componentes implementados) se consiga que la aplicación sea fácilmente configurable.

## 4. ESPECIFICACIONES Y DESARROLLO

Para el modelado completo del sistema, tanto en el diseño del software como en la arquitectura se ha utilizado una metodología representada en el método formal UML. Así la especificación y diseño se ha visto representadas por:

- **Modelo de casos de uso.** Que representa la visión del usuario de la aplicación. Y que ha sido desarrollada junto al experto del dominio.
- **Modelo estructural estático.** Muestra una visión de la estructura interna del sistema en términos de las entidades que lo integran y las relaciones existentes entre ellas (ver apartado 3), capturando el vocabulario del dominio de aplicación.
- **Modelo de comportamiento.** Describe la evolución de los posibles estrados que atraviesa el sistema y las interacciones que tienen lugar entre sus elementos. Su representación en el proyecto ha sido mediante diagramas de interacción de colaboración y secuencia; diagramas de de estado y de actividad.
- **Modelo estructural de implementación.** Por último y a más bajo nivel, este modelo permite especificar los aspectos de implementación del sistema. Se han realizado los diagramas de componentes y los diagramas de despliegue.

## 5. DISEÑO DE LA APLICACIÓN

La complejidad del modelo y su extensión, no permiten su exposición en el presente artículo, por lo que únicamente se muestra la estructura jerárquica de clases, como representación conceptual del modelo. Ésta se basa en la manera en la que conceptualmente son desagregados los contenidos que se asocian a cada evento. En la Fig. 3 se ilustra esta jerarquía de clases (estructura estática).

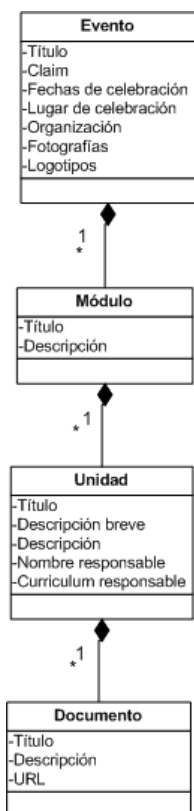


Fig. 3. Jerarquía de información de la aplicación

Los atributos de cada unidad de información serían los siguientes:

- Evento, compuesto por: título del evento, subtítulo, fechas de celebración, lugar de celebración, detalles de la entidad organizadora, fotografías generales del curso, que se usarán en la presentación y en los créditos, logotipos de las entidades organizadoras y patrocinadoras, que se usarán en la presentación y en los créditos, etc.
- Módulo, compuesto por: título del módulo, explicación de los contenidos del módulo y unidades de las que se compone.
- Unidad, compuesta por: título de la unidad, explicación detallada de los contenidos y objetivos de la unidad, persona responsable de la unidad, notas acerca del cargo o currículum del responsable, etc.

- Documento, compuesto por: título o nombre del recurso descripción del archivo o recurso y URL de enlace al recurso.

Por otro lado, el diseño se completa con una estructura de navegación lineal que relaciona los nodos de información del diagrama de clases con las pantallas de la aplicación (véase la Fig. 4).

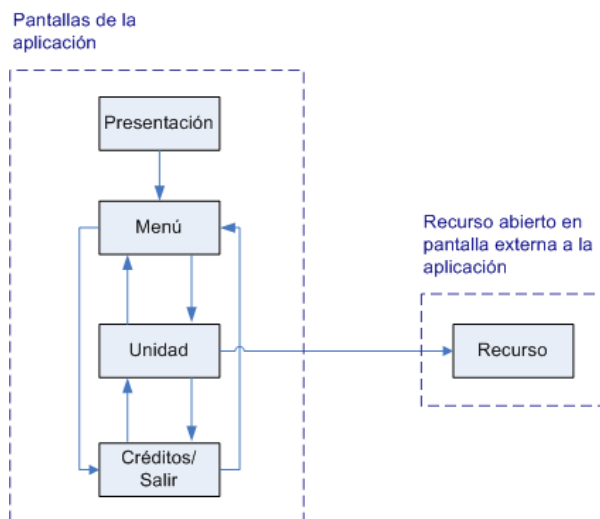


Fig. 4. Jerarquía de navegación de la aplicación

## 6. ARQUITECTURA DE DESARROLLO

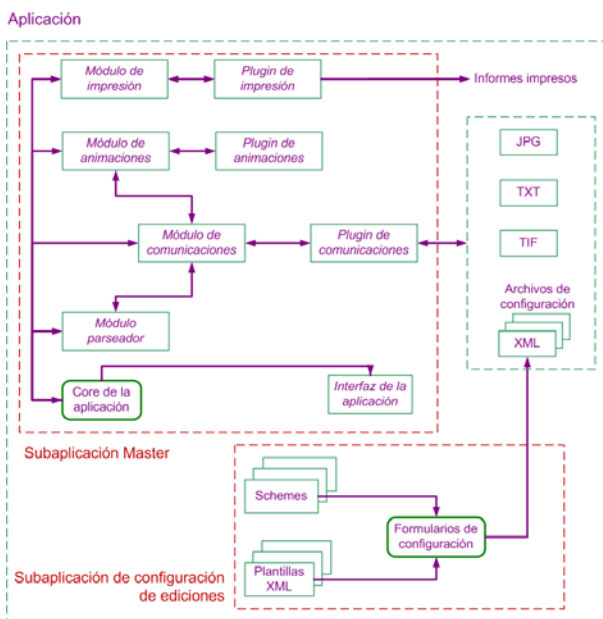
La arquitectura define los elementos principales de la aplicación y sus interrelaciones, sean éstas de comunicación interna por paso de mensajes o externas. De esta manera, se diferencian dos componentes que serán mostrados por separado al editor.

El primero de los componentes es el denominado *Subaplicación Master*, cuyas características principales son:

- Elemento considerado como *core* (corazón) de la herramienta.
- El componente es único, cerrado y siempre el mismo para cualquier edición o CD que el editor cree.
- Físicamente está integrado por un archivo ejecutable, librerías y otros archivos en los que se integran sus módulos.
- Los módulos que componen la *Subaplicación Master* son: Módulo de impresión, Plugin de impresión, Módulo de animaciones, Plugin de animaciones, Módulo de comunicaciones, Plugin de comunicaciones, Módulo parseador e Interfaz de la aplicación.

Por otro lado, se encuentra el segundo de los componentes de la arquitectura, la *Subaplicación de configuración de ediciones*:

- Dicho elemento consiste en unos formularios Excel desde los cuales se generan los archivos de configuración XML que la *Subaplicación Master* utiliza para interpretar y mostrar los contenidos del evento.
- Este componente es abierto y es el que el editor utiliza para introducir los contenidos de los eventos: información general del evento, módulos que lo componen, unidades, etc.
- Los módulos que componen la *Subaplicación de configuración de ediciones* son: Schemas SGML, Plantillas de configuración XML y los Formularios de configuración Excel para el editor.



**Fig. 5.** Esquema de la arquitectura de la aplicación con la identificación de sus componentes y módulos

Los componentes de la arquitectura, y en concreto los módulos que los integran, se encargan de funciones específicas relacionadas con la comunicación, interpretación y presentación de los contenidos, utilizando para ello la información extraída de los archivos de configuración XML generados por el editor que integran la aplicación máster (edición).

## 7. DISEÑO FUNCIONAL DE CONFIGURACIÓN

Para la definición de las estructuras necesarias para el desarrollo de los archivos de configuración se ha escogido el lenguaje de marcado XML (simplificación de SGML), considerado un estándar para el intercambio de información estructurada. El objetivo de referencia para su elección, ha sido la combinación de fiabilidad, seguridad, facilidad y difusión.

Cada tipo de archivo de configuración ha sido diseñado en función de un esquema específico de referencia DTD.

### Configuración de una edición con la generación de los archivos XML

Cada aplicación que se genera requiere de sus propios archivos XML. Existen dos tipos de archivos XML que cada aplicación puede utilizar; uno para la configuración general de la aplicación y otro tipo que sirve para almacenar los contenidos de cada uno de los módulos de que se compone.

Las características del archivo principal de configuración son:

- Es el archivo de configuración principal de cada edición que se cree.
- Su nombre es *principal.xml*.
- Únicamente existe uno por edición creada.
- Su ubicación es el directorio raíz de la edición creada.
- Almacena datos generales de la aplicación como el título del evento, los módulos que componen la aplicación, etc.

De igual manera, la creación de una edición contará con los archivos de configuración de módulos, cuyas características son:

- Es el archivo de configuración de cada módulo de que consta la edición.
- Su nombre es *nombre\_del\_modulo.xml*.
- Existe uno por cada módulo de que conste la edición.
- Su ubicación es el directorio raíz de la edición creada.
- Almacena datos generales de los módulos de la edición como el título del módulo, las unidades que componen el módulo, los archivos de contenido de cada unidad, etc.

### Automatización de la creación de los archivos de configuración mediante formularios

La creación de los archivos XML es un proceso no excesivamente complicado pero sí algo tedioso; además, exige el conocimiento detallado de las etiquetas y de algún editor de archivos de este tipo. Es decir, que exige cierta destreza tecnológica. Con el fin de cumplir con los requisitos de usabilidad, y buscando la reducción de tiempos y costes en el proceso de creación por

parte del editor, se ha buscado una metodología en la que este editor (sin grandes conocimientos tecnológicos) sea capaz de crear estos archivos de manera semiautomática.

Para ello se han diseñado unos formularios de introducción de datos, desde los que el editor los puede crear fácilmente, de forma que con sólo introducir la información en los campos adecuados, se generen los XML que después serán comprensibles para la aplicación maestra.

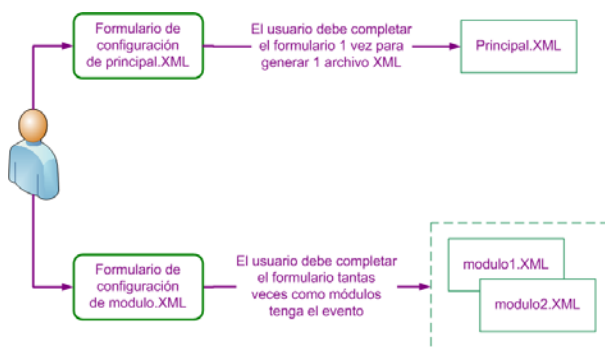


Fig. 6. Generación de archivos XML por el editor

Por otro lado, se ha buscado, como interfaz de introducción de información para la generación de los archivos XML, una aplicación que fuera de difusión masiva, de forma que cualquier editor pudiera realizar estas tareas de una manera rápida y accesible. De todas las estudiadas, y en función de los resultados obtenidos en los diseños y test aplicados, ha sido seleccionado Microsoft Excel (difusión, comodidad de uso, conocimientos exigidos y usabilidad).

De esta manera, se han creado dos formularios que el editor utilizará para la generación de los archivos de configuración XML necesarios para cada edición a crear. Éstos son:

- Un formulario, *plantilla\_modulo.xls*, es para la creación de *principal.xml*, el archivo de configuración de la aplicación.
- Otro formulario, *plantilla\_modulo.xls*, para la creación de los xml de los módulos.

### Estructura del proceso de *parseo* de los archivos de configuración

Uno de los puntos clave en el desarrollo lo ha constituido el diseño de la metodología de análisis de las secuencias de etiquetas de los XML y la identificación de las estructuras gramaticales definidas según los DTD. Este análisis de sintaxis es realizado por el Módulo *parseador*.

El Módulo *parseador*, integrado en la *Subaplicación Master*, se comunica con el *Core* y con el Módulo de comunicaciones. Después de

recibir la información de configuración de los archivos XML, la interpreta y gestiona para todas las funciones que la requieren, siempre pasando por el *Core*.

En la Fig. 7 puede observarse la secuencia completa de llamadas del proceso de *parseo* de los archivos de configuración XML.

Este proceso de comunicación se realiza por paso de mensajes encapsulados, manteniendo estructura de marcado por etiquetas.

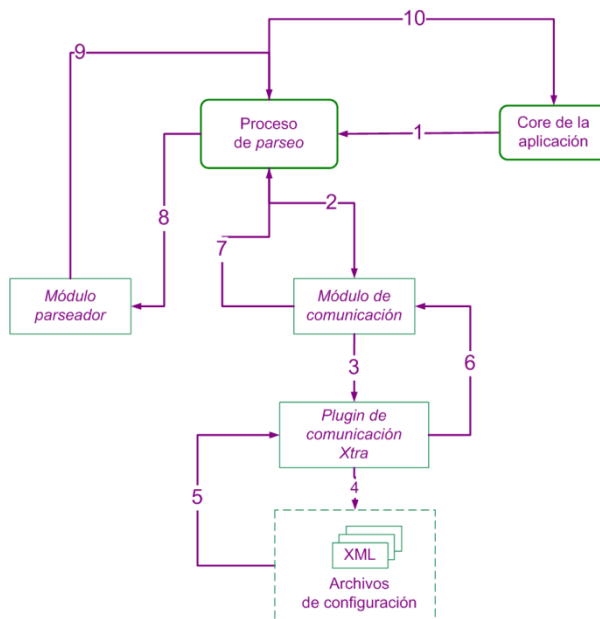


Fig. 7. Secuencia de llamadas del proceso de *parseo*

La secuencia mostrada transforma la entrada de texto en una estructura de datos tipo arborescente definida por las marcas (etiquetas) XML que será interpretable y procesable para el *core*. La construcción del árbol según el patrón especificado por el *Document Type Definition*, se ajusta a la jerarquía de información anteriormente citada, y que en definitiva será comprensible a alto nivel para el usuario y a bajo nivel para el *core* de la aplicación.

## 8. CONCLUSIONES

Después de la ejecución, se ha comprobado el cumplimiento de los objetivos. La herramienta dinámica de creación de CD's de ediciones puede suponer ciertas ventajas a las entidades que deseen difundir una información más o menos estándar acerca de ciertos eventos. De esta manera y según información recopilada en algunas de las aplicaciones ya plausibles, se presentan las siguientes ventajas:



## Automatización

El proceso se convierte en repetitivo una vez que se ha realizado la primera aplicación. Así, se consigue en unas horas, realizar un CD que suponía días de trabajo.

Por ejemplo, para uno de los test realizados, el desarrollo consumió 6 horas desde la fase de concepción, estudio y organización de la información. Todo ello sin conocimientos previos de la herramienta. Este tiempo comparado con las más de 60 horas que hubieran sido necesarias en un desarrollo a medida nos da una idea de la repercusión del presente trabajo.

## Simplicidad

Otra de las características destacables es la simplicidad de realización de una edición. Si bien es verdad que para la primera edición, es normal encontrarse con ciertas dificultades para entender los conceptos, una vez comprendido esto, el proceso se convierte en algo sencillo y repetitivo.

## Reducción de costes

Después de haber revisado en algunos test la inversión en horas o subcontrataciones que necesarias para hacer un producto a medida, frente a las necesarias haciéndolo por medio de esta herramienta; se llega a la conclusión de que puede producirse un ahorro de costes de mínimo el 60% de inversión.

## Mejora de la difusión

El hecho de disponer de la herramienta y poder realizar una edición o CD en cualquier momento y para eventos "menores" facilita la difusión de estos eventos. Mejorando la imagen y difusión de ellos.

Otra de las virtudes que se ha podido extraer, y que no estaba en un principio prevista, convirtiéndose en un efecto colateral positivo, ha sido la ampliación del concepto de evento. Que ahora se expande a visiones más empresariales. Por ejemplo alguna de las empresas que lo han testado, lo han visto y utilizado como:

- Gestor documental.
- Difusión de Memorias económicas, de calidad, etc.
- Herramienta de catálogo comercial.
- Otros.

Por otro lado, el proyecto ha sido desarrollado con vistas y en un entorno de empresa por lo que se puede contar ya con algunos ejemplos aplicados

en organizaciones y que son muestra de la efectividad de la herramienta.

## Futuro

Hablando de futuro, es relativamente fácil, poder escalar este proyecto incluyendo nuevos módulos o aplicando mayor versatilidad. Puede destacarse por ejemplo:

- Aumentar el número de componentes para aplicar mayores funcionalidades. Esta faceta se facilita por el diseño modular inicialmente concebido.
- Incluir módulos para la configuración del entorno e interfaz de la aplicación.
- Desarrollar un módulo para migración automática a formato web
- Otros.

## Bibliografía

- [1] England, E., Finney, A. (2002), *Managing Multimedia. Project Management for Web and Convergent Media*. Book 1: *Technical Issues*. Addison Wesley.
- [2] de Jong, T., Sarti, L. (1994), *Design and production of multimedia and simulation-based learning material*. Holland: Kluwer Academic.
- [3] Steinmetz, R. (2002) *Multimedia Fundamentals: media coding and content processing*. IMSC Press Multimedia Series. Prentice Hall, Inc.
- [4] Marquès Graells, P. (2000) *Metodología para la creación de materiales formativos multimedia*. Comunicación Educativa y Nuevas Tecnologías. Barcelona: Praxis
- [5] Dewar, M. (2003) *Identifying and Brokering Mathematical Web Services*. Web Services Journal Volume 3, Issue 8 (August 2003), pp 44-46.
- [6] Walmsley, P. (2001) *Definitive XML Schema*. Web Prentice Hall PTR
- [7] Kirsanov, D. (2004) *XSLT 2.0 Web Development*. Web Prentice Hall PTR.
- [8] Brogden, W. (21-jun-2003) *An XML Fragment Reader*. XML.com.
- [9] Colmenar, A. (2005) *Diseño y Desarrollo Multimedia: Herramientas de Autor*. RA-MA
- [10] Aedo, I. (1998) *Evaluación de sistemas hipermedia orientados al aprendizaje*. Educación y Tecnologías de la Comunicación. pp 161-173. Universidad de Oviedo.
- [11] Fallside, C. (2004) *XML Schema Part 0: Primer Second Edition*. W3C World Wide Web Consortium.
- [12] Mendelsohn, N. (2004) *XML Schema Part 1: Structures Second Edition*. W3C World Wide Web Consortium.
- [13] Biron, P. (2004) *XML Schema Part 2: Datatypes Second Edition*. W3C World Wide Web Consortium.
- [14] Alessi, S. (2000) *Multimedia for Learning: Methods and Development*. Allyn & Bacon.
- [15] Garrand, T. (2006) *Writing for Multimedia and the Web, Third Edition: A Practical Guide to Content Development for Interactive Media*. Focal Press.