

DESARROLLO DE APLICACIONES MULTIMEDIA INTERACTIVAS

Consuelo Belloch Ortí
Unidad de Tecnología Educativa (UTE). Universidad de Valencia

INTRODUCCIÓN

¿Cómo podemos desarrollar aplicaciones multimedia interactivas? ¿Se precisa disponer de conocimientos avanzados de informática y programación? ¿Qué sentido tiene que desarrollemos este tipo de aplicaciones?



AGENTES IMPLICADOS EN EL DESARROLLO DE APLICACIONES MULTIMEDIA

El desarrollo de programas de TIC aplicados a la educación son producto del trabajo realizado por equipos interdisciplinarios, integrados por profesionales de muy distintos campos. No obstante, existen diferentes niveles de complejidad en el desarrollo de material TIC aplicado a la educación.

Para la realización de aplicaciones multimedia interactivas debe afrontarse con un equipo interdisciplinario, en el que participan al menos tres profesionales: experto en el contenido del curso, el experto en el diseño de instrucción y el técnico programador.

Sin embargo, en ocasiones dada la facilidad de uso de los lenguajes de autor, los especialistas en educación (pedagogos, educadores, formadores, etc.) y logopedia pueden llevar a cabo el diseño y realización aplicaciones multimedia, aunque en estos casos necesitan disponer al menos de unos conocimientos mínimos sobre informática. El propio profesor, utilizando programas abiertos para la creación de materiales TIC, puede desarrollar un material sencillo para implementarlo en su actividad profesional como apoyo de la misma. No obstante, para la realización de programas de una cierta amplitud y complejidad Insa y Morata (1998) plantean la necesidad de un equipo multidisciplinario compuesto por los siguientes profesionales:

- **Director del proyecto.**- Encargado de coordinar todos los recursos y acciones del proyecto: análisis de las necesidades, diseño de las líneas generales, presupuestos, responsables, etapas y tiempos, etc.
- **Pedagogos.**- Encargados de realizar el diseño pedagógico del programa: contenidos, objetivos, metodología, recursos didácticos, evaluación, etc.
- **Especialistas en la materia del programa.**- Aportan los contenidos temáticos y el material didáctico multimedia, realizando las indicaciones didácticas concretas de cada materia.
- **Guionista.**- Realiza el diseño de la aplicación por escrito, organizando y secuencializando todos los elementos y recursos que intervienen.
- **Expertos en informática.**- Son los encargados de realizar el programa informático a partir de las indicaciones de los otros miembros del equipo.

- **Expertos en multimedia.**- Cuya finalidad será realizar y/o crear los recursos multimedia necesarios para el programa: imágenes, animaciones, videos, sonidos, etc.
- **Personas colaboradoras en el control de calidad.**- Estas personas, que no forman parte del equipo de desarrollo del programa, realizan el control de calidad del mismo, pueden ser tanto expertos sobre el tema como usuarios potenciales.

En general, para conseguir un software educativo de calidad, es necesario que en el desarrollo del mismo participen tanto técnicos informáticos como pedagogos y educadores, de modo que, la colaboración y el entendimiento entre los técnicos y los especialistas en educación será determinante para conseguir que los programadores informáticos lleven a cabo un programa atendiendo a los requerimientos de los especialistas en educación.

LENGUAJES Y SISTEMAS PARA EL DESARROLLO DE SOFTWARE EDUCATIVO

El software educativo ha sido desarrollado atendiendo a los lenguajes y programas que existían y sus posibilidades en el proceso educativo. La evolución que se ha seguido es:

- 1º Lenguajes de programación
- 2º Lenguajes de autor
- 3º Sistemas de autor

Esta evolución viene marcada atendiendo a dos aspectos: lograr entornos más amigables que faciliten el desarrollo del software educativo y, además, adaptar los lenguajes a las necesidades y características propias del software educativo.

Lenguajes de Programación

En un principio los lenguajes de programación tales como Basic, que evolucionó hasta el Visual Basic, permitían mediante comandos indicar las acciones que se deseaban que realizase el programa. Basic y sus posteriores versiones, no disponen de comandos u objetos específicos del mundo educativo, sino que es el desarrollador del software quien utiliza los comandos generales para conseguir un software con propósito, contenido y apariencia marcadamente educativa.

Los lenguajes de programación tales como C+, Basic, Pascal, etc. requieren del desarrollador de software educativo, unos conocimientos amplios en estos lenguajes y en el ámbito informático en general, por lo que no suelen ser utilizados por los educadores para la realización de software educativo, pero si por equipos interdisciplinarios en donde participe un técnico informático.

En las últimas versiones de estos lenguajes, aunque utilizan un entorno más sencillo y basado en objetos, siguen requiriendo una gran preparación específica en los comandos y sentencias propias, así como en los procedimientos y sistemas generales de programación.

Creemos que debemos hacer una mención especial al lenguaje utilizado para desarrollar páginas web, el lenguaje "HTML" (HyperText Markup Language), puesto que actualmente es uno de los lenguajes más utilizados por los profesores, su popularidad se debe principalmente a su facilidad de uso para

desarrollar páginas, y a las potencialidades educativas que tienen estas páginas. El lenguaje HTML, desarrollado por Tim Berners-Lee en 1991, parte del uso de comandos o etiquetas (tags) que deben incorporarse atendiendo a ciertas reglas. Este lenguaje nos permite programar las páginas web aprovechando sus diferentes posibilidades (inclusión de imágenes y/o animaciones, realización de enlaces, etc.). Las diferentes versiones que se han realizado en el lenguaje HTML han posibilitado un aumento de sus potencialidades tanto a nivel general como educativo, en este sentido creemos conveniente reseñar que la última versión HTML 4.0, ha mejorado considerablemente la posibilidad de diseñar páginas web accesibles para todos, aportando soluciones que permiten adaptar este material a las personas con discapacidades.

Lenguajes de Autor

Los lenguajes de autor suponen un gran avance sobre los lenguajes de programación, puesto que disponen de comandos específicos para aquellas tareas propias del proceso de instrucción, tales como el análisis de respuestas, recopilación de información sobre los resultados y procesos realizados por el estudiante, etc. Sin embargo, estos lenguajes siguen requiriendo un conocimiento sobre programación, ya que siguen utilizando comandos y sentencias que es necesario conocer para indicar al ordenador lo que debe realizar.

Un ejemplo de lenguaje de autor es el lenguaje Tutor, cuya primera versión fue desarrollada en la Universidad de Illinois en 1959, por un equipo multidisciplinar, con el fin de facilitar que los profesores pudieran crear su propio material educativo, este lenguaje posibilitó un desarrollo importante de la EAO.

Sistemas de autor

Los sistemas de autor son programas que, trabajan bajo Windows u otro sistema operativo, utilizan menús, iconos y plantillas de herramientas que permiten desarrollar software educativo sin necesidad de conocer y memorizar la sintaxis propia de los lenguajes de programación y de los lenguajes de autor. Además, estos sistemas dan respuesta a la mayoría de las necesidades que tiene un profesional de la educación cuando desea realizar un programa de EAO o multimedia. Los Sistemas de autor son, por tanto, programas muy potentes que permiten realizar distintos tipos de programas educativos (tutoriales, simuladores, aplicaciones multimedia, etc.) de una forma bastante sencilla. La mayoría de ellos, actualmente dispone, además de sentencias y comandos propios de un lenguaje de autor, con el fin de permitir realizar tareas más complejas.

Sistemas de autor para el desarrollo de software educativo: **ToolBook, Authorware**

ToolBook es una herramienta de programación que utiliza el lenguaje "OpenScript", de fácil comprensión y gran potencia. Permite la construcción de forma rápida y sencilla de aplicaciones que combinan los elementos típicos de las aplicaciones windows (ventanas, menús, iconos, etc.) con recursos multimedia. Una de las grandes ventajas de este sistema de autor es la posibilidad que brinda para construir interface complejos, en cuanto a su potencialidades y recursos, de una forma sencilla.

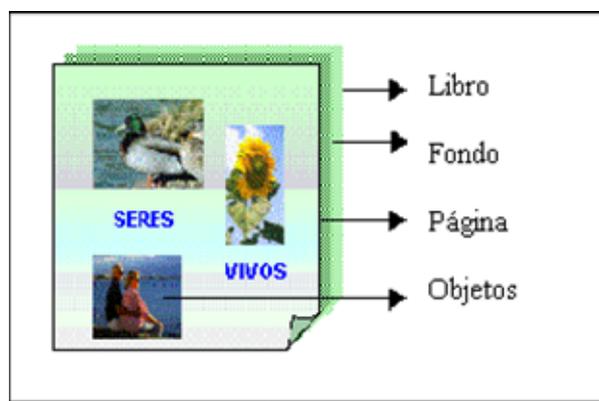
Con ToolBook podemos trabajar en dos niveles o modalidades:

Nivel de autor: Permite construir la aplicación multimedia.

Nivel lector: Permite utilizar la aplicación multimedia.

A nivel de autor, la construcción de una aplicación con ToolBook se basa, como su nombre indica, en construir un libro, diseñando la páginas que lo forman, creando y modificando los objetos incluidos en las páginas y escribiendo el programa con el lenguaje OpenScript. Los elementos utilizados por ToolBook para la creación de aplicaciones, pueden ser organizados jerárquicamente:

Libro (Book) – Fondo (Background) – Página (Foreground) – Objetos



Los **libros de ToolBook**, constituyen por si mismos una aplicación como por ejemplo una lección o unidad didáctica y permiten reunir varios libros conformando una aplicación global en la que se incluyen diferentes aplicaciones o libros, de este modo, por ejemplo podríamos reunir todas las unidades didácticas de una determinada materia o curso de modo que para el estudiante se presentasen como una única aplicación.

El **fondo** es utilizado para establecer elementos comunes que se comparten en todas o un grupo de páginas del libro (fondos, imágenes, botones, etc.), esta capacidad permite dar una coherencia en la presentación de las páginas de una libro ToolBook, en relación en relación a la finalidad o contenido de las mismas.

Las **páginas de ToolBook** se corresponden con las pantallas diseñadas para cada libro, en cada una de ellas se mostrará una información diferente, para ello se utiliza una programación basada en objetos.

Por tanto, podemos decir que los **objetos** son los elementos que se incorporan en una página (imágenes, botones, cuadros, animaciones, campos de texto, etc.). La funcionalidad de los objetos viene determinada por las propiedades de los mismos y por las acciones que pueden realizarse a partir de

ellos. Para determinar estas acciones ToolBook utiliza el lenguaje OpenScript. Existe la posibilidad de crear grupos de objetos, cuando se precisa que un conjunto de objetos se comporten de un mismo modo, esto es respondan conjuntamente a una acción del usuario.

Las últimas versiones de ToolBook han supuesto un giro en esta herramienta de autor hacia la enseñanza basada en Web (WBT: Web Based Training). De este modo el sistema permite distribuir las aplicaciones en formato CD-ROM, o desde la red (Intranet o Internet), exportando el libro a formato HTML.

Programas educativos abiertos

Algunos autores, han creado programas educativos abiertos, que si bien no disponen de todas las potencialidades de los sistemas de autor, pueden ser utilizados por los docentes para la elaboración de aplicaciones multimedia sencillas.

Los programas educativos son abiertos cuando permiten que el profesor realice cambios en el contenido y secuencia de las actividades propuestas por el programa. La cantidad e importancia de las adaptaciones que puede realizar el profesor dependerá del programa educativo, en algunos de ellos, el profesor sólo tiene una mínima posibilidad de cambio, por ejemplo: eliminar una determinada actividad, mientras que otros programas son totalmente abiertos pudiendo el docente realizar todas las modificaciones y adaptaciones que considere oportunas.

Los programas educativos abiertos, generalmente, permiten que el docente elabore las actividades del programa e indique la secuencia que debe seguirse por parte del alumno. Estos programas tienen gran acogida por los profesores ya que, al no ser programas cerrados, los profesores tienen libertad para realizar cambios o realizar un conjunto de actividades totalmente nuevas y adaptadas a sus necesidades en el aula.

Un ejemplo de este tipo de programas es el JCLIC con el que cada profesor diseña y programa las actividades que realizarán con posterioridad sus alumnos. En este tipo de programas es muy importante, que los profesores no necesiten grandes conocimientos informáticos para llevar a cabo sus actividades. El programa JCLIC de Francesc Busquets es un programa gratuito, puesto a disposición de los profesionales por el Programa de Informática Educativa (PIE) de la Generalitat de Cataluña, al que se puede acceder por la Web en la dirección de la Xarxa Telemática Educativa de Cataluña: <http://clic.xtec.net/es/jclic/index.htm> .

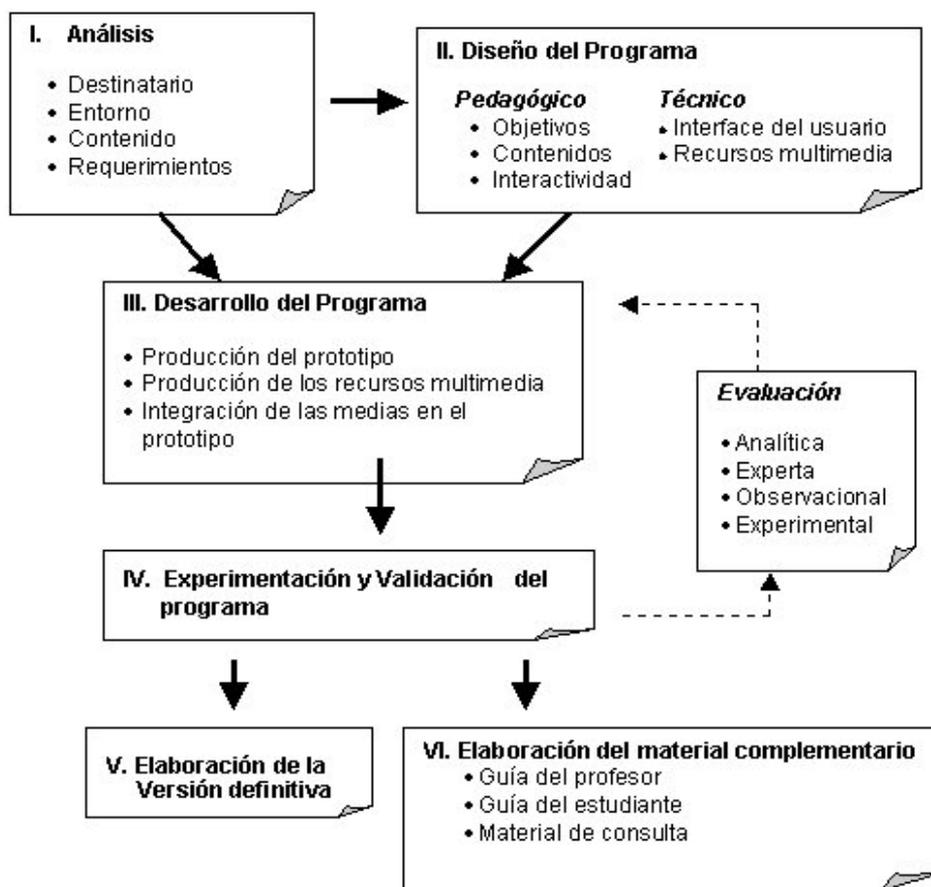
FASES EN EL DESARROLLO DE APLICACIONES MULTIMEDIA INTERACTIVAS

El proceso que se sigue para desarrollar software educativo, consta de varias fases o etapas, interdependientes:

1. Análisis
2. Diseño del Programa

3. Desarrollo del Programa
4. Experimentación y Validación del Programa
5. Realización de la Versión definitiva del programa
6. Elaboración del material complementario

En el cuadro siguiente se muestra un esquema de las diferentes fases que se siguen en la producción de aplicaciones TIC para la educación, especificando las tareas fundamentales llevadas a cabo en cada una de ellas.



Fases del desarrollo de software educativo.

A continuación exponemos brevemente la finalidad de cada una de estas fases.

Fase 1. Análisis

Esta fase tiene por finalidad recabar información sobre determinadas características de los usuarios, metodología, hardware, etc. que no formando parte integra del diseño de la aplicación, es necesario tener presente siempre para que la aplicación sea adecuada.

Dentro de esta primera fase, se pretende analizar aspectos tales como:

- **Características de los usuarios.**- Debemos conocer la edad, nivel sociocultural, conocimientos previos sobre el tema, conocimientos que tienen sobre las herramientas informáticas y en concreto sobre la EAO, motivación de los usuarios para la utilización del programa informático, etc.
- **Características del entorno de aprendizaje.**- Es necesario realizar un análisis de las condiciones espacio-temporales en las que el programa se va a utilizar por parte de los usuarios. Así, deberemos conocer el entorno físico al que se destina el programa (domicilio particular, centro escolar, empresa, centro de formación, etc.). También, la modalidad temporal con la que se trabajará, indicando el tiempo necesario para el aprendizaje, el tiempo de que van a disponer los usuarios para el mismo y la modalidad temporal de trabajo, esto es si son los usuarios los que establecerán las sesiones de trabajo de una forma libre, o si se implementará con sesiones y horario fijo de trabajo.
- **Análisis del contenido.**- A partir de los análisis previos, se realizará una primera aproximación al contenido del programa, indicando el tema a tratar, la profundidad de los contenidos que se va a realizar, y una descripción global del temario.
- **Requerimientos técnicos.**- Es necesario especificar los requerimientos o características que debe cumplir el hardware y el software para que el programa pueda ser utilizado. Si el programa ha sido solicitado por un centro o empresa, los requerimientos técnicos deberán ajustarse al equipo informático con el que cuenta el centro o empresa.

Esta primera fase está muy relacionada con la siguiente fase del programa, formando los planteamientos sobre los que se asienta la fase del Diseño.

Fase 2. Diseño del programa

En esta fase se precisa la colaboración de los expertos en pedagogía e informática, de modo que el primero elabore el diseño pedagógico del curso y el segundo el diseño técnico, planteando la viabilidad de las propuestas pedagógicas y el lenguaje o sistema de autor que sería conveniente utilizar.

El **diseño pedagógico** del curso llevará a establecer:

- **Las líneas pedagógicas** del curso, el modelo de aprendizaje en que se va a basar la aplicación, la finalidad y objetivo general de la misma.
- **Diseño de contenidos** en el que se establecerán los objetivos específicos de la aplicación (conceptuales, procedimentales y actitudinales) y las unidades didácticas adecuadas para la consecución de estos objetivos. Para cada una de estas unidades didácticas se establecerá: el contenido temático, las actividades, la evaluación, las fuentes documentales y los recursos multimedia que se van a utilizar.
- **Interactividad** del programa, para ello se confeccionará el mapa de navegación en el que se indicarán gráficamente los itinerarios que el usuario puede seguir en el programa cuando busca información, señalando los vínculos entre pantallas y los vínculos entre eventos.

El contenido del **diseño técnico** dependerá del diseño pedagógico, de los requerimientos técnicos y de los lenguajes o sistemas utilizados para desarrollar la aplicación. Los aspectos que se trabajan en el diseño técnico son:

- Los **elementos multimedia**, esto es el material audiovisual que va a implementarse en la aplicación. Su diseño tendrá siempre presente el resultado final que se desea obtener.
- El **Interface del usuario**. Para algunos autores como D.Insa y R.Morata el interface del usuario es "el espacio de trabajo, es decir la pantalla del ordenador, en el que el usuario encontrará los elementos que le van a permitir interactuar con el sistema informático" (1998: 96). Sin embargo, esta definición es algo restrictiva ya que únicamente atiende al interface gráfico de comunicación, a este habría que añadir, como indica Alcantud (1999) todos aquellos elementos del ordenador que el usuario utiliza para relacionarse con el programa (pantalla, ratón, teclado, etc.). Esta distinción es de gran importancia ya que nos permite tener una visión más amplia sobre el modo de comunicación entre usuario y programa.

Fase 3. Desarrollo del programa

Consiste en la realización de una versión inicial del programa. Para ello, se pueden seguir los siguientes pasos:

- **Desarrollo del prototipo**, que suele consistir en el desarrollo de una unidad o módulo completo de un curso (contenido, actividades, evaluación, etc.). Si el programa educativo es más sencillo el prototipo puede ampliarse a los contenidos de la versión definitiva del programa. El formato del prototipo deberá ajustarse al formato final que se desee realizar de la aplicación.
- **Elaboración de los recursos multimedia**, seleccionando las características que deben tener los mismo para que formen parte del programa y se ajusten a los requisitos pedagógicos e instructivos. Para ello se utilizara el software y hardware específico para la creación de los multimedia.
- **Integración de los recursos multimedia** en la versión inicial del programa.

Fase 4. Experimentación y validación del programa

Consiste básicamente en realizar una evaluación de los diferentes aspectos del prototipo, analizando la calidad de los mismos y su adecuación. La evaluación a realizar será una formativa, para comprobar que todos los elementos del programa funcionen correctamente, y si no es así, realizar las modificaciones oportunas sobre el prototipo del programa.

Para D.Insa y R.Morata la evaluación formativa permite:

- *identificar y subsanar las dificultades que puedan plantearse o surgir a lo largo del proceso de implementación del sistema;*
- *comprobar la integración operativa de los elementos de la aplicación y del usuario con el sistema;*
- *verificar la consecución progresiva de los objetivos y finalidades de la aplicación."* (1998: 201)

Con el fin de controlar la calidad del programa, como indica Alcantud (1999), pueden realizarse diferentes tipos de evaluación:

- **La evaluación analítica.**- Descripción de todos los componentes del programa, y predicción de su uso y utilidad por parte del usuario.
- **La evaluación experta.**- En la que actúan como jueces expertos tanto en el tema como en TIC aplicada a la educación, analizando el mismo e identificando problemas para que se subsanen o proponiendo mejoras en el mismo.
- **La evaluación por observación.**- Consiste en observar como los usuarios del programa interactúan con él, para reunir información sobre la conducta de los usuarios y posibles mejoras en el prototipo.
- **Evaluación experimental.**- Consiste en realizar un control experimental de determinadas variables del prototipo, comprobando los efectos de los cambios en la variable sobre la actividad y rendimiento de los usuarios.

Los problemas encontrados permitirán depurar el prototipo, realizando la mejora del mismo, posteriormente se realizará otra vez la evaluación del prototipo. Esta actividad se realizará de forma circular de modo que se seguirá validando hasta que se considere que el prototipo a superado el control de calidad tanto a nivel técnico como pedagógico o formativo.

En cualquier caso, estos elementos de evaluación-validación no pueden entenderse como separados de las estrategias y criterios que se emplean para la evaluación en general de este tipo de programas. La alternativa que parece más razonable, aunque rara vez se plantea así de forma explícita, es la utilización de los mismos criterios y estrategias en ambos casos, siendo aquí un ensayo más reducido y concentrado en los errores y desajustes más importantes.

Fase 5. Realización de la versión definitiva del programa

La realización de la versión definitiva del programa se produce cuando se ha tomado la decisión por parte del equipo que trabaja en el desarrollo de la aplicación, de que esta cumple los requisitos de calidad. Se producirá de este modo el master que permitirá realizar las copias o duplicaciones del programa.

Fase 6. Elaboración del material complementario

Para finalizar, una vez producido el programa, se elaborará el material complementario que acompañará al mismo, tal como la guía del profesor y la del alumno, en las que se incluirán principalmente, aspectos pedagógicos y de contenido. Además, también se realizará un manual de consulta que contendrá información técnica y de uso del programa.

Aunque la mayoría de los autores plantean esta fase secuenciada respecto al desarrollo general y su validación este no debiera ser un esquema universal y requeriría una reflexión al respecto. La primera justificación para esta estrategia es que lo relevante es la aplicación propiamente dicha y los materiales restantes son secundarios, lo que no es una alternativo muy sólida en general y, particularmente, desde

una perspectiva educativa. Parece más razonable concebir el conjunto de componentes como algo integrado en la unidad del programa que siga los mismos procesos de construcción y revisión. Desde el punto de vista educativo, además, no parece adecuado concebir las guías y manuales de los usuarios como algo menos relevante que la aplicación propiamente dicha, con lo que su integración en un único proceso de desarrollo debería ser el estándar de actuación.

La única justificación sostenible es que la elaboración de este material depende en gran medida de los resultados del proceso de revisión-validación del conjunto de la aplicación. Unificar el proceso de desarrollo supone, pues un incremento sustancial de los costes de desarrollo que no siempre se respaldan desde una concepción de “mercado” y de la “relevancia” que se les atribuye.

PARA SABER MÁS

Alcantud, F. (1999) (ed.) Teleformación: Diseño para todos. Valencia: Servei de Publicacions de la Universitat de Valencia Estudi General.

Insa, D. y Morata, R. (1998) Multimedia e Internet: las nuevas tecnologías aplicadas en la educación. Madrid: Paraninfo.

Clares, J. (2005) Propuesta de diseño pedagógico para la elaboración de un programa educativo multimedia.

http://www.quadernsdigitals.net/index.php?accionMenu=hemeroteca.VisualizaArticuloIU.visualiza&articulo_id=8880

Sampedro, A., Granda, E., Miláns del Bosch, M., Lupiáñez, F., Martínez Nistal, A. y Pérez Herrero, H..

Aspectos implicados en la elaboración de material educativo multimedia (MEM) para el fomento de la motivación empresarial. [http://www.aces-andalucia.org/documentos/aspectos implicados en elaboraci3n de material multimedia.pdf](http://www.aces-andalucia.org/documentos/aspectos%20implicados%20en%20elaboraci3n%20de%20material%20multimedia.pdf)

Rodríguez Illera, J.L. Diseño y producción de software educativo. Universidad de Barcelona.

<http://www.quadernsdigitals.net/articuloquaderns.asp?IdArticle=3562>

Sánchez Rodríguez, J. (2003) Producción de aplicaciones multimedia por docentes. Revista Pixel-bit, nº 21. <http://www.sav.us.es/pixelbit/articulos/n21/n21art/art2109.htm>

Anzulovich, G. Metodología para trabajar multimedia. <http://tecno.unsl.edu.ar/multimedia/10.pdf>

EDUTEKA. Cómo elaborar una WebQuest de calidad o realmente efectiva.

<http://www.eduteka.org/WebQuestLineamientos.php>

EDUTEKA. Consejos de los expertos para realizar presentaciones efectivas.

<http://www.eduteka.org/GuiaPresentaciones.php3>