

Telos

ISSN: 1317-0570

wileidys.artigas@urbe.edu

Universidad Privada Dr. Rafael Beloso Chacín
Venezuela

Quintero, Hugo; Portillo, Lisbeth; Luque, Rafael; González, Marisela
Desarrollo de software educativo: una propuesta metodológica
Telos, vol. 7, núm. 3, septiembre-diciembre, 2005, pp. 383-396
Universidad Privada Dr. Rafael Beloso Chacín
Maracaibo, Venezuela

Disponible en: <http://www.redalyc.org/articulo.oa?id=99318837004>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal

Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto



Desarrollo de software educativo: una propuesta metodológica*

Development of Educative Software: A Methodological Proposal

*Hugo Quintero***, *Lisbeth Portillo****
*Rafael Luque***** y *Marisela González******

Resumen

En este trabajo se propone una metodología para el diseño de software en educación, la cual reúne tanto los elementos educativos como los de análisis y diseño de sistemas, integrando las etapas de realización de software con las teorías de aprendizaje. Esta se ha puesto en práctica en los software denominados “Aprendiendo Física con Experimentos” y “Conociendo algunos físicos, los cuales están en proceso de validación de usuarios finales, quienes solicitan el material para ponerlo en práctica en su aula, a un 81,2% le pareció fácil la utilización del mismo, el 84% se sintió cómodo con la utilización del mismo.

Palabras clave: Software educativo, metodología, elaboración de software.

Recibido: Mayo 2005 • Aceptado: Agosto 2005

- * Proyecto adscrito al Centro de Estudios Matemáticos y Físicos de la Facultad de Humanidades y Educación de la Universidad del Zulia.
- ** Dr. en Ciencias Gerenciales de la URBE. Magíster en Informática Educativa. Profesor Asociado de la Facultad de Humanidades y Educación de la Universidad del Zulia. E-mail: hquinter @luz.ve
- *** Magister Scientiarium en Ciencias de la Comunicación. Mención Nuevas Tecnologías de la Información de LUZ Profesor de la Facultad de Humanidades y Educación de la Universidad del Zulia. E-mail: liportil@luz.ve
- **** Magister Scientiarium en Matemáticas Aplicadas a la Ingeniería, de LUZ Profesor Agregado de la Facultad de Humanidades y Educación de la Universidad del Zulia E-mail: relamix@mixmail.com, rluque@luz.ve
- ***** Magister Scientiarium en Ciencias Aplicadas, área Física de LUZ. Profesor Titular de la Facultad de Humanidades y Educación de la Universidad del Zulia. E-mail: cacoveluz@yahoo.com

Abstract

This paper proposes a method for the design of software that would serve education, and which unites educational elements as well as system design and analysis by integrating the stages of development of software with learning theories. The proposal has been put into practice through the software "Learning physics with experiments" and "Getting to know some physicist". These programs are in the process of validation by ultimate users who acquire them for use in classroom (81,2%) and who find it easy to use the software. 84% of users evaluated the system as user-friendly.

Key words: Educational software, methodology, software development.

Introducción

El avance de la tecnología en los últimos años es incalculable, y se observa en los diversos campos del desarrollo industrial y social. Entre estos avances se encuentra el computador con aplicaciones en diversas áreas de trabajo, logrando ser parte importante para la automatización de procesos que manualmente serían largos y tediosos. Además de ello, se comenzó a utilizar el computador como herramienta instruccional, pero pronto se descubrió que era capaz de administrar instrucción, surgiendo los sistemas de autoaprendizaje asistido por tal herramienta.

Luego se generalizó el uso de este tipo de sistema naciendo modalidades como el adiestramiento basado en el computador (ABC), instrucción asistida por el computador (IAC), y el Software Educativo. En Venezuela, actualmente existen muchas empresas e instituciones educativas que vienen adoptando esta modalidad de instrucción, como recurso novedoso para la formación de sus empleados y estudiantes, considerando así la tecnología computacional como el medio para abordar la educación en distintos niveles y áreas del conocimiento, por lo que la relación que existe entre computación y educación puede ser estudiada desde tres dimensiones distintas, como son las que consideran la computación como: objeto de estudio, medio de enseñanza-aprendizaje y herramienta de trabajo en educación, Quintero (1999).

1. Software educativo

Se puede definir un Software educativo como el conjunto de programas educativos y programas didácticos creados con la finalidad específica de ser utilizados para facilitar los procesos de enseñanza y de aprendizaje, caracterizándose por ser interactivo porque contesta inmediatamente las acciones de los estudiantes y permite un diálogo e intercambio de informaciones entre el computador y los estudiantes; permite individualizar el trabajo de los estudiantes ya que se adapta al ritmo de trabajo de cada uno y puede adaptar sus actividades según las actuaciones de los alumnos; fácil de usar porque los conocimientos informáticos necesarios para utilizar la mayoría de estos programas son mínimos, aunque cada programa tiene unas reglas de funcionamiento que es necesario conocer. Amaya (1999).

Peley (1999), en su trabajo *Software Educativo en el área numérica sobre polinomios*, plantea que la educación como proceso vital y permanente, que va más allá de la escolaridad, está ligada a la informática debido a la transformación que ha venido experimentando esta última. Esta herramienta ha dejado de utilizarse solamente como una técnica para resolver problemas de procesamiento de datos convirtiéndose en un elemento estratégico para el desarrollo de las potencialidades en los aprendices, expandiendo sus capacidades y ayudando a entender el mundo desde su propia perspectiva.

La vida profesional producto del desarrollo de aprendizajes e inmerso en un mundo de cambio constante, halla en el software educativo una herramienta útil para tener acceso a información pertinente, y así lograr el contacto con recursos propios de la informática que forman parte de redes virtuales, permitiendo participar en eventos que no son sincrónicos en el tiempo ni en el espacio, de manera que cada usuario puede administrar su tiempo de estudio y genere su propio aprendizaje.

Según Contreras (2003), un software educativo es aquel que cumple con las normas de ser lúdico, innovador, expresivo, motivador, instructivo e informativo y de esa manera ser un medio didáctico, que permita cambios positivos dentro de una clase de estudio, a través del cual se puede individualizar el trabajo, potenciando el aprendizaje.

Para Sánchez (2000), el software educativo es aquel material de aprendizaje especialmente diseñado para ser utilizado con un computador en los procesos de enseñanza y aprendizaje.

Plantea Quintero (1999), que un software educativo es una herramienta didáctica de alto rendimiento que ayuda al estudiante a adquirir conocimientos, habilidades y destrezas en áreas previamente planificadas.

2. El computador como medio educativo

El grupo de investigación y Desarrollo de Informática Educativa de la Universidad de los Andes (1997) citado en Quintero 1999, propone que los materiales educativos computarizados tienen sentido ya que aprovechan las características del computador como medio, para crear ambientes educativos que generan aprendizajes en áreas específicas del conocimiento. En este sentido, cabe destacar las características que diferencian al computador de otro medio para aprender:

1. El computador está en capacidad de almacenar, procesar y presentar información multimedia (sonido, imágenes, video) en forma interactiva, de modo que es posible crear contextos para el aprendizaje planteándose una relación dialogal con el nivel de concreción o abstracción necesario, bajo el control del diseñador o el usuario, según convenga.
2. El computador puede comportarse con distintos niveles de inteligencia adquirida, para lo cual un material educativo computarizado muestra similitud mayor o menor en su ejecución, al comportamiento racional de un

humano. La inteligencia no es un atributo dicotómico (existe o no), sino una cualidad que puede mostrar distintos niveles de desarrollo, es decir, un ejercitador (programa de ejercicios) planteando distintos grados de inteligencia: en su forma más básica sólo mostrara al aprendiz si su respuesta es o no correcta, pero gradualmente puede hacer otras cosas, tales como adaptar los ejercicios dependiendo de las características y desempeño del aprendiz, dar explicaciones o sugerencias derivadas del proceso, o incluso, resolver ejercicios que propone el estudiante.

3. El computador hace posibles distintos grados de interactividad, entre la máquina y el usuario, en virtud de la cual el usuario en su condición como aprendiz está en control de lo que sucede, dentro de los condicionantes del micromundo en que se lleva a cabo la acción, es decir, el software educativo.
4. Es capaz de hacer conexiones y articulaciones con los medios y recursos para el aprendizaje, permitiendo así la creación de ambientes cooperativos de aprendizajes, aprovechando las cualidades únicas de otros medios (transmisivos, experienciales, interactivos) y la creación de ambientes educativos multimedia. Es decir, el computador permite articular multimedia interactiva dentro de ambientes multimediales educativos en los que el profesor juega un rol preponderante.

Estas características llevan hoy día a discusiones donde se da prioridad al sentido de lo que se puede hacer en informática con respecto al apoyo que pueda brindar esta herramienta. Pues, lo que se puede hacer en informática sólo está limitado por la imaginación del diseñador.

3. Metodologías aplicada para el desarrollo del software educativo

Según Carrión et al. (2003), en el desarrollo de un software educativo se deben considerar dos elementos fundamentales: la estructura y la teoría de aprendizaje. La estructura tiene que ver con los modelos de elaboración de software en ingeniería, planteando etapas de construcción como es el análisis, diseño, desarrollo, evaluación e implementación, considerando el enfoque dirigido hacia la producción del conocimiento por parte del usuario final que en este caso es el estudiante.

El otro elemento es la teoría de aprendizaje, entre las que destacan las siguientes: El constructivismo, teoría centrada en el desarrollo de estrategias de aprendizajes de manera que permitan profundizar en la flexibilidad cognitiva y desarrollo de hipertextos. El conductivismo donde se usan principios de enseñanza programada: descomposición de la información en unidades, diseño de actividades que requieren respuestas del estudiante y planificación del esfuerzo. El cognitivismo, que tiene como base la condición de aprendizaje del estudiante tanto en las fases internas (motivación, comprensión, retención, recuerdo, generalización, ejecución y retroalimentación), como en fases externas (eventos de las instrucciones externas al individuo e instrucciones a cada proceso de aprendizaje).

Por otro lado, el CIED (1997) propone una metodología para el desarrollo de proyectos multimedia y Software Educativo denominada ABC, en la cual se realizan las siguientes etapas o fases para el diseño del sistema, estas son:

1. **Estudio Preliminar:** En esta etapa se realiza un pequeño estudio de las necesidades y requerimientos del sistema y la integración de un equipo interdisciplinario necesario para el desarrollo del mismo, considerando la pertinencia en función de sus aplicaciones.
2. **Diseño Instruccional:** Es la etapa para realizar el diseño instruccional, el cual está constituido por los siguientes pasos:
 - **Identificación de la población a adiestrar:** Se identifica la población para la cual está dirigido el sistema, sus características generales.
 - **Delimitación del curso:** Se delimita el contenido del diseño en cuanto a lo que se tratará como contenido en el área de conocimiento necesario.
 - **Objetivos instruccionales:** Estos objetivos reunirán la conducta esperada del aprendiz (usuario) y el patrón de rendimiento donde se especifican los contenidos a considerar durante el proceso de instrucción, propuesto en el software.
 - **Evaluación:** A partir de la misma, se miden los conocimientos adquiridos en los conceptos y ejercicios que se proponen en el software, considerando si su finalidad principal es sumativa o formativa.
 - **Utilización de los medios:** El punto clave es lograr que en cada segmento de aprendizaje el niño utilice todos los sentidos y pueda realizar alguna actividad que lo motive a utilizar el computador de manera que alcance el mayor porcentaje de los conocimientos impartidos por el software.
3. **Fase del desarrollo de guiones:** Durante esta etapa se realizan los mapas de navegación y guiones de producción, los cuales contienen las guías y el texto y la descripción de todas las características utilizadas en las pantallas, para determinar los niveles de interacción del usuario con el software.
4. **Fase de construcción:** Durante esta etapa se plasma lo planificado anteriormente en un programa de computación, siguiendo las directrices definidas en la etapa del diseño instruccional. Durante esta etapa se utilizan principios básicos, en psicología del color y técnica de procesamiento de textos, para construir la interfaz gráfica con el usuario y los formatos de las pantallas. De igual forma también se definirán los tipos de botones así como la barra de navegación.
5. **Fase de Validación:** Una vez concluida la etapa de construcción del sistema se realizarán una serie de pruebas y validaciones dirigida a los siguientes puntos:
 - Validación del contenido.
 - Validación del sistema como programa de computación.
 - Validación del sistema en el área Instruccional.

De igual manera, Cataldi et al. (2003) conciben la metodología para diseñar software educativo como un modelo que consta de siete etapas integradoras:

1. **Proceso de selección del modelo de ciclo de vida:** Es la etapa para seleccionar un modelo de ciclo de vida acorde con la teoría educativa elegida.
2. **Proceso de iniciación, planificación y estimación del proyecto:** Se debe establecer la matriz de actividades considerando la teoría educativa elegida.
3. **Proceso de exploración de conceptos:** Dentro de esta etapa, se deben en primer lugar identificar las necesidades educativas, luego formular posibles soluciones potenciales y que a su vez sean compatibles.
4. **Proceso de asignación del sistema:** Para ello es necesario en primer lugar: Definir las funcionalidades del programa y luego desarrollar la arquitectura del programa con base en la teoría educativa elegida.
5. **Proceso de análisis de los requisitos de software:** Durante esta etapa se debe definir el tipo de programa a desarrollar y el tipo de interactividad para poder luego integrar requisitos educativos y de software.
6. **Proceso de diseño:** Esta es la etapa donde se diseñan los menús, las pantallas y su organización, los tipos de iconos a usar, se seleccionan efectos (sonido, vídeo, etc.) y textos. Se realizan los guiones y se definen los criterios de navegación, las actividades (búsqueda, ejercitación, etc.), el tipo de módulos (problemas, evaluación, etc.) y los tipos de ayudas didácticas (errores, mensajes).
7. **Proceso de documentación técnica.** Es la última etapa de este proceso, constituye el momento en el cual se somete a la validación interdisciplinaria el mismo y producto de estas, se realizan los ajustes necesarios con base en estas validaciones.

Otra metodología es la presentada por Galvís (1994), la cual es bastante completa y es una buena guía para el desarrollo del software. En esencia se conservan los grandes pasos o etapas de un proceso sistemático para desarrollo de estos materiales, dándole particular énfasis a: la solidez del análisis como punto de partida; el dominio de teorías sustantivas sobre el aprendizaje y la comunicación humana; la evaluación permanente y bajo criterios predefinidos a lo largo de todas las etapas del proceso como medio de perfeccionamiento continuo del material; la documentación adecuada y suficiente de lo que se realiza en cada etapa, como base para el mantenimiento que requerirá el material a lo largo de su vida útil.

Por ello, esta metodología contiene las siguientes etapas:

1. Análisis de necesidades educativas

En esta etapa se analizan los problemas o situaciones problemáticas existentes, sus causas, posibles soluciones y cual de estas últimas son aplicables y pueden generar los mejores resultados. Es en este momento se deben responder las siguientes interrogantes: ¿cómo identificar las necesidades o los problemas existentes?, ¿Qué criterios usar para llegar a decidir si amerita una solución computarizada?

¿Con base en qué, decidir si se necesita un Material Educativo Computarizado (MEC) y qué tipo de MEC conviene que sea, para satisfacer una necesidad dada?

Por lo anterior, es necesario centrarse en : la consulta a fuentes de información apropiadas e identificación de problemas, analizar las posibles causas, alternativas y establecer cual es el papel que jugará el computador en la solución del mismo.

2. Selección o planeación del desarrollo de MEC

Durante esta etapa se identifican los posibles paquetes que puedan satisfacer las necesidades planteadas. Es imprescindible someterlos al ciclo de revisión y prueba de MEC, dentro de la cual es indispensable tener acceso a una copia documentada de cada MEC, realizar una revisión de expertos en contenido, metodología e informática. Los primeros, para garantizar que efectivamente corresponde al contenido y objetivos de interés. Los expertos en metodología para verificar que el tratamiento didáctico es consistente con las estrategias de enseñanza/aprendizaje que es aplicable a la población objeto y al logro de tales objetivos. Los expertos en informática para verificar que dicho MEC se puede ejecutar en la clase de equipos de que dispondrán los alumnos y que hace uso eficiente de los recursos computacionales disponibles.

1. Ciclos para la selección o el desarrollo de MECs

Galvis (1994) plantea que el punto de partida es la identificación de necesidades educativas reales que conviene atender con material educativo computarizado. Dependiendo del resultado final de esta etapa, se procede en el sentido contrario al avance de las manecillas del reloj, cuando se trata de seleccionar un MEC; pero en el mismo sentido del avance del avance de las manecillas, si conviene efectuar su desarrollo.

En cualquiera de los dos ciclos, una vez que se dispone de un MEC, se requiere evaluarlo con un grupo piloto de alumnos que pertenezca a la población objeto, bajo las condiciones para las cuales está diseñado. Esta es la base para decidir si el MEC debe llevarse a la práctica en gran escala, o para echar pie atrás, rediseñarlo, ajustarlo o desecharlo. Durante su implementación también es importante que se evalúe el MEC, de modo que se pueda establecer la efectividad real del material, éste es el sentido de la prueba de campo educativo que es deseable que el MEC cumpla para satisfacer la necesidad.

A partir de los resultados del análisis, es conveniente hacer explícitos los datos que caracterizan el:

- **Diseño de comunicación**

La zona de comunicación en la que se maneja la interacción entre usuario y programa se denomina interfaz. Para especificarla, es importante determinar cómo se comunicará el usuario con el programa, en el momento de la entrada y en el momento de la salida (interfaces de entrada y salida).

- **Diseño computacional**

Con base en las necesidades se establece cuales funciones debe cumplir el MEC en apoyo de sus usuarios. Cuando el usuario es el alumno, un MEC puede brindarle la posibilidad de controlar la secuencia, el ritmo, la cantidad de ejercicios, de abandonar y de reiniciar. Por otra parte, al profesor, un MEC puede ofrecerle la posibilidad de editar los ejercicios o las explicaciones, cumplimiento de cada una de las funciones de apoyo definidas para el MEC por tipo de usuario.

Se recomienda realizar la programación modular, por tipo de usuario, y mediante refinamiento a pasos, de manera que haya niveles sucesivos de especificidad hasta que se llegue finalmente al detalle que hace operacional cada uno de los módulos que incluye el MEC. La estructura lógica deberá ser la base para formular el programa principal y cada uno de los procedimientos que requiere el MEC.

- **Desarrollo de MEC**

Una vez que se dispone de un diseño debidamente documentado es posible llevar a cabo su desarrollo en el tipo de computador seleccionado, usando herramientas de trabajo que permitan, a los expertos cumplir con las metas en términos de tiempo y de calidad de MEC.

- **Prueba piloto de MEC**

Con la prueba piloto se pretende ayudar a la depuración del MEC a partir de su utilización por una muestra representativa de los tipos destinatarios para los que se hizo y la consiguiente evaluación formativa. Para llevarla a cabo apropiadamente se requiere preparación, administración y análisis de resultados en función de buscar evidencia para saber si el MEC está o no cumpliendo con la misión para la cual fue seleccionada o desarrollada.

- **Prueba de campo de MEC**

La prueba de campo es la oportunidad de comprobar, en la vida real, que el MEC que se había probado solo a nivel experimental, cumple los objetivos para los cuales fue desarrollado.

4. Propuesta metodológica

Partiendo de las metodologías antes expuestas y luego de la experiencia acumulada en la elaboración de software, se propone una metódica para abordar, de manera mas concreta, con una cantidad apreciable de elementos y en concordancia con la revisión realizada, el diseño y desarrollo de software educativo. La misma se presenta en virtud de que, en opinión de los autores, las metodologías consultadas se subdividen en las diferentes áreas de estudio, produciendo una serie de procedimientos que pueden ser repetitivos, motivando que el desarrollo del software se convierta en un camino largo y complejo por un lado y por otro genera altos costos en función del tiempo y el esfuerzo en la producción del mismo.

Es por esto que la metodología propuesta, pretende lograr la sinergia entre las teorías de aprendizaje, el análisis, diseño, desarrollo e implementación de siste-

mas propias del área de la ingeniería y las etapas del desarrollo de un software para educación, mediante una concreción gráfica donde se puede apreciar en su totalidad, los diferentes elementos que conforman la metodología, logrando a la vez, visualizar cada una de las partes así como también la integración de las mismas, de manera sistémica. De esta forma los especialistas que van a desarrollar el software pueden operacionalizar cada una de las etapas y al mismo tiempo, verificar toda la secuencia de forma integrada.

El modelo propuesto (Tabla 1), plantea 5 etapas, en las cuales las dos primeras comprenden el análisis del sistema, iniciando en la etapa 1 el estudio preliminar en el cual se determina la necesidad existente en el área de conocimiento, así como la población a la cual se dirige el software.

En la segunda etapa, se analiza el diseño instruccional correspondiente al área de estudio, se consideran los objetivos instruccionales propuestos por el Ministerio de Educación y Deporte, en el caso de Venezuela, con sus respectivas evaluaciones, para tomarlo en cuenta en el desarrollo del software educativo.

Las etapas 3 y 4 son respectivamente el diseño y el desarrollo. Durante su realización lo mas relevante es la determinación de la o las teorías de aprendizaje que van a ser consideradas, pues son ésta las que orientan la conducta en el proceso de aprendizaje del usuario. Para determinarlas, es necesario, realizar una revisión exhaustiva de las mismas y relacionarlas con las etapas anteriores. Por tanto, luego de conocerlas todas, se podrán determinar los niveles de interacción que tendrá el usuario con el software.

Una vez establecidas las teorías a considerar, el primer paso de la etapa 3, lo constituye el diseño de los guiones. Ellos guiarán la forma como se presentarán los contenidos y la distribución de éstos en las pantallas, los niveles de interacción (conceptos, imágenes, sonidos), y elementos dinámicos. De esa forma, se obtendrá un primer esbozo de como se irá construyendo el software.

Una vez obtenido el guión, se procede a trabajar en la etapa 4, en la cual se procede a elaborar el diseño del mapa de navegación u algoritmo, éste consiste en graficar las relaciones entre los diferentes contenidos expresados en pantallas y los elementos de las mismas. Esta será la guía que utilizarán los especialistas en el momento de la programación utilizando programas de aplicación tales como los lenguajes Macromedia Authorware 6 y Microsoft Photo Editor, entre otros. Una vez realizada la programación se concreta el material educativo computarizado.

Una vez finalizadas las etapas anteriores, se tiene como producto el software, el cual será sometido a un proceso de validación, antes de exponerlo a los usuarios finales (Etapa 5). La validación consiste en la evaluación de expertos y pruebas pilotos, mediante los cuales se revisa lo siguiente: los contenidos, el área instruccional, y los software de aplicación. Los contenidos tienen que ver con el área de estudio hacia el cual esta orientado el software, el área instruccional se refiere a como se distribuyen por objetivos los diferentes contenidos programáticos en el plan de acción didáctica; y la plataforma computacional tiene que ver con los programas de aplicación y con el hardware requerido.

Tabla 1
Software Educativo

Análisis		Teoría de Aprendizaje		Implementación
		Diseño	Desarrollo	
			Etapa 4	Etapa 5
		Etapa 3	Fase de construcción	▲ Validación
	Etapa 2	Desarrollo de Guiones	Mapa de navegación (algoritmo)	Contenido
Etapa 1	Diseño Instruccional	Diseño de pantallas	Programas de Aplicación	Área instruccional
Estudio Preliminar	Objetivos Instruccionales	Niveles de interacción entre pantallas	Programación	Programa de Computación
Identificar la población	Evaluación establecida como logro de objetivos	Elementos dinámicos y didácticos a considerar		Hardware requerido
Determinar área de Estudio		Relatos de ejecución de pantallas		▼ Reingeniería

Fuente: Quintero, Luque, González y Portillo 2004.

A partir de esta etapa, el software puede o no entrar en un proceso de rein-geriería, dependiendo de los resultados de la etapa de validación. Cuando el software ya esta en proceso de implementación y se coloca en manos del usuario final, sigue el proceso de validación y las recomendaciones y sugerencias de los usuarios son recogidas de manera de realizar los ajustes necesarios para la constante actualización del material.

Esta metodología se ha puesto en practica dentro del proyecto de investigación “Diseño y producción de materiales educativos para la enseñanza de la matemática y las ciencias experimentales, dentro del cual se han generado algunos productos como el software “Aprendiendo Física con Experimentos” y “Conociendo algunos físicos”, los cuales, se han utilizado como material de apoyo para la enseñanza en la asignatura Física de diferentes menciones de la Escuela de Educación de la Universidad del Zulia, tales como: Fundamentos de Física de Básica Integral, Física para Química, Física para Biología y en algunas asignaturas propias de la Licenciatura en Educación Matemática y Física. Fue utilizado también en talleres dictados a profesores de Educación Media y en el curso de noveno grado Unidad Educativa Santa Angela.

La razón de la utilidad del material en un publico tan diverso obedece a la necesidad sentida que se presenta en los profesores y maestros en la enseñanza de física y ciencias naturales en cuanto a la falta de implementos en los laboratorios para cumplir con los objetivos educacionales, a la poca accesibilidad de material no impreso para conocer biografías de científicos y al desconocimiento por parte de los docentes en asociar los fenómenos físicos al entorno natural y cotidiano, lo cual fue verificado haciendo uso de la primera etapa de la metodología planteada, además de las evidencias de autores tales como Castellano y D Alessandro (2003) en su investigación.

En la Fig. 1 se muestran las pantallas principales del material educativo Enseñando física con experimentos y la Fig. 2 muestra parte del mapa de navegación del software.

En una primera evaluación realizada al software Enseñando física con experimentos, se pudo constatar, mediante la aplicación de un instrumento a docentes de básica en ejercicio, la aceptación de este material. Ello se evidencia desde el momento de la aplicación de la prueba piloto donde los docentes, solicitan un ejemplar del mismo para utilizarlo en su practica cotidiana. Con relación a los resultados específicos de la aplicación del instrumento, se apreció que al 81,2% le pareció fácil la utilización del mismo, el 84% se sintió cómodo con la utilización del mismo, con relación a la facilidad para conseguir el material para realizar las experiencias que plantea el software un 69,3% manifestó que era de fácil y un 22,7% medianamente, con lo cual se pude aseverar que los materiales son de fácil adquisición.

Figura 1

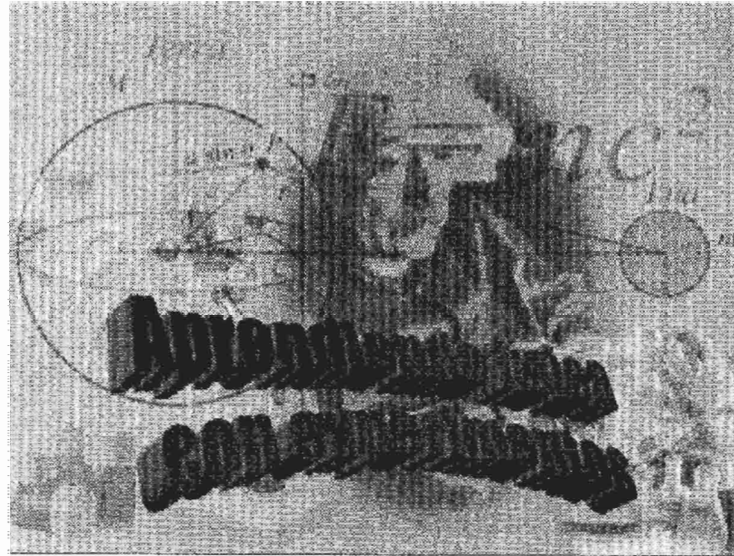
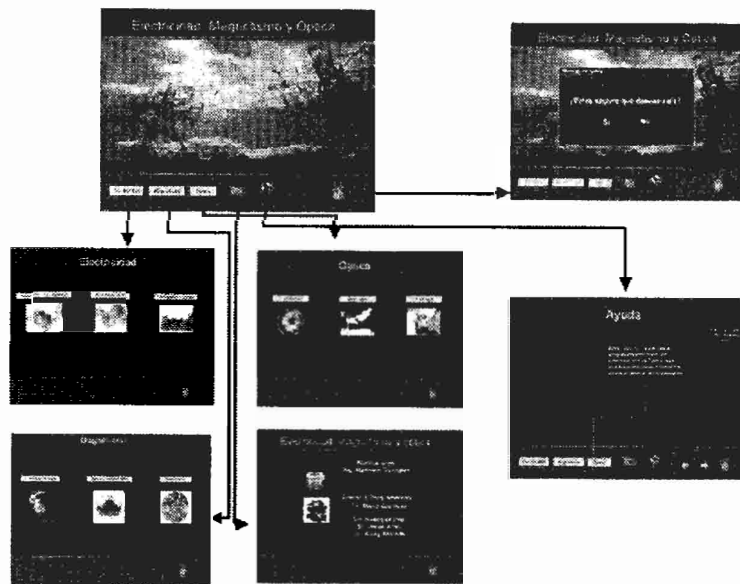


Figura 2



Conclusiones

- Luego de una revisión de las metodías para el desarrollo de software, se evidencia que existe similitud entre todas las metodologías y que todas las fases son muy importantes.
- En las fases de diseño y desarrollo es donde se concentra el mayor grado de importancia porque se interrelacionan las teorías de aprendizaje a utilizar con las etapas de análisis y diseño preliminares y se logra concretar el guión el cual va a ser orientador para que el programador pueda lograr un material de calidad adaptado al área y al usuario a quien va dirigido.
- Para el desarrollo del software educativo es necesario conocer ante todo la necesidad en el área del conocimiento y los usuarios naturales a quien va a ir dirigido el software.
- Para el diseño de software con una determinada intencionalidad educativa, se necesita tener una concepción clara acerca de cómo se producen los procesos de enseñanza/aprendizaje, es decir a partir de las teorías de aprendizaje se debe fundamentar el desarrollo de software educativo y cómo lo condicionan.
- Todo material educativo es necesario someterlo a una validación para verificar la necesidad de aplicar la reingeniería para lograr un material de óptima calidad.

Bibliografía

- Amaya, J.(1999). El software educativo. Encontrado en <http://www.angelfire.com/az2/educacionvirtual/menuprincipal.html>). 1999. fecha de consulta 13 de junio 2003. hora 19:19.
- Castellano y D´ Alessandro (2003). Proyectos de Investigación: una metodología para el aprendizaje significativo de la Física en Educación Media. **Revista de Pedagogía**. Vol. XXIV, No.69. 101-136.
- Cataldi, Z. Lage, F. García, R (2003). Metodología Extendida para la creación de software educativo desde una visión integradora.[enred]. Disponible: <http://www.itba.edu.ar/capis/webcapis/RGMTBA/articulosrgm/R-extremadura-2.pdf>.
- Carrión, P.; García, A.; Perez, Y. (2003). La Ingeniería de Software Aplicada al desarrollo de software educativo. .[enred]. Disponible:<http://132.248.45.5/enlinea/ponencia/mesa2/patiCM.doc>.
- CIED (1997). **Nuevas Tecnologías**, Material mimeografiado.
- Contreras, M (2003). El Ingeniero en Sistemas en la elaboración de software educativo.[en red]. Disponible en: <http://usbbog.edu.co/ingenium/uno/formaciondisciplinar.com>
- Galvis, A. (1994). **Ingeniería de Software Educativo**. Ediciones Uniandes.

Hugo Quintero, Lisbeth Portillo, Rafael Luque y Marisela González
Telos Vol. 7, No. 3 (2005) 383 - 396

Laboratorio de Investigación y Desarrollo de Informática Educativa (1997).
Universidad de Los Andes.

Peley, H. (1997). **Software Educativo en el área numérica sobre polinomios.**
URBE.

Quintero, H. (1999). Diseño de un software educativo para el desarrollo de los
procesos cognoscitivos del niño con retardo mental moderado: caso 10 a
12 años. URBE.

Sánchez, J. (2000). **Nuevas Tecnologías de la Información y Comunicación
para la Construcción del Aprender.** Universidad de Chile. Santiago.