

## **METODOLOGÍA EXTENDIDA PARA LA CREACIÓN DE SOFTWARE EDUCATIVO DESDE UNA VISIÓN INTEGRADORA**

**ZULMA CATALDI<sup>1</sup>, FERNANDO LAGE<sup>1</sup>, RAÚL PESSACQ<sup>2</sup> y  
RAMÓN GARCÍA-MARTÍNEZ<sup>3</sup>**

1.- Lab. de Informática Educativa. Facultad de Ingeniería  
Universidad de Buenos Aires  
Paseo Colón 850 (1063) Capital Federal (Argentina)

2.- Facultad de Ingeniería. Universidad de La Plata  
Calles 1 y 47(1900) La Plata. Buenos Aires (Argentina)

3.- Programa de Magister en Ingeniería de Software  
Instituto Tecnológico de Buenos Aires  
Av. Madero 399 (1106) -Capital Federal (Argentina)

**Resumen:** En este artículo se resume el trabajo de investigación que llevamos a cabo durante los últimos años, en el área de las metodologías para el diseño, desarrollo y evaluación de software educativo. La metodología que se describe, es aplicable al proceso de desarrollo de software educativo, ya que contempla en las distintas etapas metodológicas los aspectos de naturaleza pedagógico-didácticas que no son tenidos en cuenta en las metodologías convencionales para el desarrollo de software. Debido a la diversidad y a la multiplicidad de las actividades que se requieren para elaborar el producto software, la metodología da soporte a un desarrollo tecnológico interdisciplinario, que tiene como pilares a la ciencia informática y a las teorías del aprendizaje.

**Palabras clave:** software educativo, nuevas tecnologías en la educación, tecnología educativa, metodología didáctica, teorías del aprendizaje.

**Resumo:** Este artigo se resume em um trabalho de investigação que desenvolvemos durante os últimos anos, na área de metodologia de projetos, desenvolvimento e avaliação de programas (software) educativos. A metodologia que se descreve, e aplicável ao processo de desenvolvimento de software educativo, já que contemplam nas distintas etapas metodológicas os aspectos da natureza pedagógico-didática que não são tidas em conta nas metodologias convencionais para o desenvolvimento de software. Devido à diversidade e da multiplicidade de atividades que se requer para elaborar o produto software, a metodologia da suporte para um desenvolvimento tecnológico interdisciplinar, que tem como pilares a ciência da informática e as teorias da aprendizagem.

**Palavras chaves:** software educativo novo tecnologias na educação, tecnologia educativa, metodologia didática, teorias de aprendizagem.

---

## INTRODUCCIÓN

Este trabajo surge como respuesta a los problemas con que se encuentran los docentes, especialmente los no-informáticos cuando tienen que integrar los equipos de desarrollo de software educativo al decidir construir sus propios programas educativos. A través de entrevistas a los docentes de los diferentes niveles del sistema educativo se detectó la existencia de muy poca información publicada en un lenguaje de fácil acceso para el docente no especializado y mucha menos que considere la necesidad de integrar los aspectos pedagógicos para las buenas prácticas educativas a los desarrollos de software efectuados desde la programación.

Como resultado de las investigaciones realizadas en el LIE (Laboratorio de Informática Educativa), respecto del diseño y la evaluación de software educativo, se puede decir que los marcos conceptuales tenidos en cuenta para llevar a cabo el desarrollo esencialmente están centrados en dos pilares que son las teorías del aprendizaje y las metodo-

logías propias de la ingeniería de software arribando a lo que denominamos ingeniería del software educativo.

Como punto de partida, se ha realizado una prospección de las teorías de la educación contemplando el condicionamiento operante de Skinner (1958-63), la instrumentalización cognitiva de Bruner (1988-1991), la perspectiva psicogenética de Piaget (1989), el aprendizaje significativo de Ausubel (1973), los mapas conceptuales de Novak (1984), el marco sociocultural de Vigotzkii (1989), la teoría uno de Perkins (1995), la teoría de las inteligencias múltiples de Gardner (1987, 1993,1995), sin dejar de lado los aportes de Coll (1994), Sancho (1994), Jonhson-Laird (1998), Pozo Municio (1998), Cabero (1992-2000), a fin de considerar de acuerdo a las necesidades tanto el conductismo, el constructivismo como las teorías cognitivas actuales y sus variantes. Estas teorías permitirán elaborar el software de acuerdo a la necesidad en cada caso, ya sea un programa para el entrenamiento de personal en primeros auxilios, de ejercitación, de refuerzo o para la construcción de significados a partir del descubrimiento. Es decir de acuerdo al requerimiento el docente podrá utilizar la que crea más conveniente.

En cuanto a la Ingeniería de Software, la investigación documental se orientó hacia a la identificación de metodologías de diseño, que contienen los métodos, las herramientas y los procedimientos específicos para la construcción de software (Boehm, 1981,1988, Piattini, 1996; Pressman, 1993; Juristo 1995), y las metodologías que contemplan las métricas de calidad del producto obtenido (Fenton, 1991). En una primera instancia no se han incursionado en aquellas orientadas a objetos, dejándolas para una etapa posterior.

Las dos perspectivas descritas anteriormente se articulan *en la matriz de actividades del modelo del ciclo de vida elegido*. El uso de este instrumento, pareció razonable, ya que los docentes durante las entrevistas efectuadas hicieron evidente la necesidad de herramientas fácilmente Interpretables. Sin perder de vista los dos grandes marcos conceptuales parece prudente hacer algunas observaciones acerca de la evolución del software educativo en cual se identifican dos grandes etapas: la primera es el uso de la computadora como un espacio constructivista de aprendizaje con base piagetiana (salvando las distancias era el

equivalente a poner a un niño a jugar en el mundo de bloques) y la segunda, en la última década, con base a la creación de ambientes facilitadores del proceso de construcción de software la computadora pasó a ser el instrumento para soportar ambientes de desarrollos de software que asistieran a los aprendizajes dejando de lado esa visión constructivista de la primera etapa, y convirtiéndose en un instrumento complementario de las actividades áulicas.

Esta última etapa se ha consolidado en los últimos tiempos a través de la creación de ambientes diferenciadores (Cabero, 2000) específicos de desarrollo basados en lenguajes visuales y herramientas de autor.

## DESARROLLO

A partir de los dos marcos conceptuales citados anteriormente, se pueden enunciar el problema detectado del modo siguiente: *“Las metodologías propias de la ingeniería del software no cautelan aspectos pedagógicos-didácticos del producto software educativo a desarrollar”*.

Esta falencia tiene su correspondencia en la definición de actividades de cada proceso en las respectivas matrices de actividades y se debe destacar que muy pocas metodologías cautelan los aspectos comunicacionales con el usuario que son un componente elemental y básico que debe ser satisfecho en el momento de desarrollar un software educativo.

Debido a ello, es que la solución propuesta para esta problemática consistió en la formulación de la extensión de una metodología para que estos aspectos sean contemplados. Esta extensión se logró redefiniendo la matriz de actividades correspondiente al ciclo de vida del software elegido, siendo esta el documento en el que quedan definidos todos los procesos y las actividades a efectuar en cada etapa del desarrollo del software.

Para articular la solución que se propone se exploraron los ciclos de vida más frecuentemente utilizados en el marco de las metodologías ingeniería de software tales como el modelo en cascada, el modelo incremental o de refinamientos sucesivos, el prototipado evolutivo, el

modelo en espiral de Boehm (Boehm,1988) y los modelos orientados al objeto. (Piattini, 1996).

Analizados en detalle los distintos modelos de ciclo de vida, se eligió como ciclo a extender para cautelar los aspectos pedagógicos-didácticos, el de prototipado evolutivo, como una primera aproximación a la solución.

La elección del modelo se debe básicamente a las siguientes razones:

- Cuando se trata de un software a ser desarrollado por encargo, es deseable obtener un primer esbozo de lo que será el programa tan pronto como fuera posible a fin de satisfacer la curiosidad del usuario, y para saber realmente qué es lo que éste quiere e incorporar sus sugerencias de cambio, si las hubiera, lo antes posible, es decir en etapas tempranas de la construcción.
- Por otra parte, es necesario saber lo antes posible si los desarrolladores han interpretado correctamente las especificaciones y las necesidades del usuario.
- En muchos casos los usuarios no tienen una idea acabada de lo que desean, por lo tanto los desarrolladores deben tomar decisiones y suponer que es lo que el usuario quiere. Por este motivo, ello la emisión de los prototipos brinda la posibilidad de efectuar refinamientos de los requerimientos en forma sucesiva a fin de acercarse al producto deseado.

Es decir, la decisión se fundamenta en la ventaja de la realización de los cambios en etapas tempranas y la posibilidad de emisión varios prototipos evaluables durante el desarrollo, obteniéndose de este modo paralelamente una metodología integral también para el proceso de evaluación del programa. (Cataldi, Lage, et al., 2000c y da).

En la Tabla 1 se enumeran las fases del ciclo de vida para prototipos evolutivos básica y en publicaciones previas de Cataldi Z., Lage F., et al. (2000a) se pueden encontrar la definición detallada de cada una de ellas.

<ol style="list-style-type: none"> <li>1. <i>Factibilidad (FAC),</i></li> <li>2. <i>Definición de requisitos del sistema (RES),</i></li> <li>3. <i>Especificación de los requisitos del prototipo (REP),</i></li> <li>4. <i>Diseño del prototipo (DPR),</i></li> <li>5. <i>Diseño detallado el prototipo (DDP),</i></li> <li>6. <i>Desarrollo del prototipo (codificación) (DEP),</i></li> <li>7. <i>Implementación y prueba del prototipo (IPP),</i></li> <li>8. <i>Refinamiento iterativo de las especificaciones del prototipo (aumentando el objetivo y/o el alcance). Luego, se puede volver a la etapa 2 o continuar si se logró el objetivo y alcance deseados. (RIT),</i></li> <li>9. <i>Diseño del sistema final (DSF),</i></li> <li>10. <i>Implementación del sistema final (ISF),</i></li> <li>11. <i>Operación y mantenimiento (OPM),</i></li> <li>12. <i>Retiro (si corresponde) (RET).</i></li> </ol>
---

Tabla 1: Las fases del ciclo de vida de prototipos evolutivos

Luego de la definición de las etapas mostradas en la Tabla 1, se procede a la construcción del documento denominado *matriz de actividades*, donde se definen las actividades a desarrollar en cada fase, y los procesos asociados a cada una de ellas.

Para este caso seleccionado, se identificó que deben ser extendidas las diez primeras fases. Por lo tanto hubo necesidad de incorporar *procesos nuevos que cautelaran las necesidades pedagógicas-didácticas* (están listados en la Tabla 2) *y hubo que definir nuevas actividades que cautelasen cuestiones pedagógicas-didácticas en los procesos existentes*, en diez de las fases del ciclo de vida en los procesos que se describen en la Tabla 3 de acuerdo a Cataldi, Lage, et al. (2000a).

<b>1. PROCESO DE IDENTIFICACIÓN DE LA NECESIDAD EDUCATIVA</b>	<i>Identificar la necesidad del programa educativo Seleccionar la teoría educativa a utilizar.</i>
<b>2. PROCESO DE ANÁLISIS DE LOS REQUISITOS EDUCATIVOS.</b>	<i>Definir los objetivos educativos Definir las características del grupo destinatario Definir contenidos y el recorte de contenidos Definir estrategias didácticas Definir las actividades mentales a desarrollar Definir nivel de integración curricular Definir tipo de uso del programa y nivel de interactividad</i>

	<p><i>Definir efectos motivantes</i>  <i>Definir posibles caminos pedagógicos</i>  <i>Definir tiempo y modo de uso del programa</i>  <i>Definir hardware asociado</i></p>
<b>3. PROCESO DE EVALUACIÓN DE LOS PROTOTIPOS DE SOFTWARE</b>	<p><i>Confeccionar el instrumento de evaluación</i>  <i>Evaluar prototipos del programa</i>  <i>Elaborar los resultados</i>  <i>Identificar cambios y ajustes a realizar</i>  <i>Llevar a cabo modificaciones pertinentes</i>  <i>Archivar resultados</i></p>
<b>4. PROCESO DE EVALUACIÓN INTERNA Y EXTERNA DEL SOFTWARE</b>	<p><i>Confeccionar el instrumento de evaluación</i>  <i>Evaluar interna y externamente el programa</i>  <i>Elaborar los resultados</i>  <i>Identificar cambios y ajustes a realizar</i>  <i>Llevar a cabo modificaciones pertinentes</i>  <i>Archivar resultados</i></p>
<b>5. PROCESO DE EVALUACIÓN CONTEXTUALIZADA</b>	<p><i>Diseñar la evaluación: definir grupos: de control y experimental, docente, tiempo, modo.</i>  <i>Aplicar la prueba</i>  <i>Identificar posibles problemas</i>  <i>Realizar las modificaciones y ajustes de la versión</i></p>
<b>6. PROCESO DE DOCUMENTACIÓN DIDÁCTICA</b>	<p><i>Planificar la documentación didáctica</i>  <i>Elaborar guía didáctica</i>  <i>Adjuntar la información didáctica pertinente</i>  <i>Producir la documentación y adjuntarla al programa.</i></p>

Tabla 2: Los procesos nuevos que se incorporaron y sus actividades.

<b>1. PROCESO DE SELECCIÓN DEL MODELO DE CICLO DE VIDA</b>	<p><i>Seleccionar un modelo de ciclo de vida acorde con la teoría educativa elegida</i></p>
<b>2. PROCESO DE INICIACIÓN, PLANIFICACIÓN Y ESTIMACIÓN DEL PROYECTO</b>	<p><i>Establecer la matriz de actividades considerando la teoría educativa elegida</i></p>
<b>3. PROCESO DE EXPLORACIÓN DE CONCEPTOS</b>	<p><i>Identificar las necesidades educativas</i>  <i>Formular posibles soluciones potenciales</i>  <i>Formular soluciones potenciales compatibles.</i></p>

<p><b>4. PROCESO DE ASIGNACIÓN DEL SISTEMA</b></p>	<p><i>Definir las funcionalidades del programa Desarrollar la arquitectura del programa en base a la teoría educativa elegida.</i></p>
<p><b>5. PROCESO DE ANÁLISIS DE LOS REQUISITOS DE SOFTWARE</b></p>	<p><i>Definir el tipo de programa a desarrollar Definir el tipo de interactividad Integrar requisitos educativos y de software</i></p>
<p><b>6. PROCESO DE DISEÑO</b></p>	<p><i>Definir la organización de los menús Definir tipo de iconos a usar Seleccionar efectos a usar (sonido, vídeo, etc.) Seleccionar textos a usar Asegurar facilidad de lectura Realizar diseño de las pantallas Realizar diseño de los menú Realizar storyboards Definir los criterios de navegación Definir las actividades (búsqueda, ejercitación, etc.) Definir tipo de módulos (problemas, evaluación, etc.) Definir tipos de ayudas didácticas (errores, mensajes)</i></p>
<p><b>7. PROCESO DE DOCUMENTACIÓN TÉCNICA</b></p>	<p><i>Incluir los resultados de las evaluaciones</i></p>

Tabla 3: Las nuevas actividades incorporadas a los procesos existentes.

Finalmente, en la Tabla 4 se listan todos los procesos a tener en cuenta, se enumeran los documentos a emitir como así también las técnicas, los métodos y las herramientas a utilizar en cada uno de ellos. Desde el punto de vista metodológico, la definición de cada uno de los pasos a seguir determinando las herramientas a usar, significa disponer de una manera clara y precisa de una "hoja de ruta" para continuar con el proyecto.

<i>Procesos</i>		<b>Documento de salida</b>	<b>Métodos/Técnicas/ Herramientas a emplear<sup>1</sup></b>
<i>Proceso de identificación de la necesidad educativa</i>		Definición del marco educativo y comunicacional.	Encuesta, entrevista
<i>Proceso de selección del modelo de ciclo de vida</i>		Ciclo de vida adoptado	
<i>Proceso de iniciación, planificación y estimación del proyecto</i>		Plan de gestión del proyecto	Diagrama de Gantt o CPM. Modelos empíricos de estimación
<i>Proceso de seguimiento y control del proyecto (programa),</i>		Análisis de riesgos y plan de contingencias. Registro histórico del proyecto	Modelizado. Prototipado. Revisiones. Auditorías. Análisis CPM.
<i>Proceso de gestión de calidad del software</i>		Plan de garantía de calidad. Recomendaciones de mejora de calidad.	Técnicas de planificación. Métricas de calidad del software
<i>Proceso de exploración de conceptos</i>		Informe de necesidades. Posibles soluciones factibles	Análisis Costo Beneficio. DFD. Prototipado
<i>Proceso de asignación del programa (sistema).</i>		Especificación de requisitos funcionales de hardware y software. Especificación de interfaces del sistema o programa. Descripción funcional. Arquitectura.	DFD Módulos
<i>Proceso de análisis de requisitos educativos</i>		Especificación de los objetivos y estructuración de conceptos. Selección de contenidos y pertinencia.	Enfoques cognitivistas. Enfoques constructivistas. Estrategias cognitivas.
<i>Proceso de análisis de requisitos del software</i>		Especificación de requisitos del software, de interfaces de usuario y otros software. Interface de hardware y con el sistema físico.	Análisis estructurado. DFD. Diagramas E/R. Técnicas de Prototipación.
<i>Proceso de diseño</i>	<i>de los contenidos</i>	Identificación de los procesos mentales a estimular. Definición de las actividades a realizar por los alumnos. Jerarquización de los conceptos.	Uso de estrategias cognitivas. Teoría de Ausubel y Novak. <sup>2</sup> Uso de mapas conceptuales.

<sup>1</sup> CPM significa Método del Camino Crítico, DFD es diagrama de flujo de datos, E/R son diagramas entidad- relación.

<sup>2</sup> Se ha señalado esta teoría como una guía simplemente, a modo de referencia, puede utilizarse otra, de acuerdo a las necesidades y/o tipo de propuesta educativa. Ver referencias: Ausubel (1983) y Novak (1988).

	<i>del software</i>	Descripción del diseño del software y de la arquitectura. Descripción del flujo de información, bases, interfaces y algoritmos.	Programación estructurada. Programación Orientada a objetos. Técnicas de prototipado.
<i>Proceso de implementación e integración de módulos</i>		Datos para las pruebas. Documentación del sistema o programa y del usuario. Plan de integración.	Lenguajes de Programación
<i>Proceso de instalación</i>		Plan de instalación. Informe de Instalación.	Lenguajes de Programación
<i>Proceso de operación y soporte</i>		Histórico de pedidos de soporte	Análisis estadístico.
<i>Proceso de mantenimiento</i>		Recomendaciones de mantenimiento	Reaplicar el ciclo de vida
<i>Proceso de retiro</i>		Plan de retiro	
<i>Proceso de verificación y validación</i>		Plan de verificación y validación Plan de pruebas. Especificación y resumen de la prueba. Software probado.	Pruebas de caja negra y pruebas de caja blanca
<i>Proceso de evaluación de los prototipos del software</i>		Diseño del instrumento de evaluación. Resumen de la prueba. Selección de la muestra.	Cuestionario estructurado, semi y abierto.
<i>Proceso de evaluación interna y externa del software</i>		Diseño del instrumento de evaluación. Resumen de la prueba. Selección de la muestra.	Cuestionario estructurado, semi y abierto
<i>Proceso de evaluación contextualizada</i>		Diseño de la experiencia. Definición de los grupos de control y experimental.	Técnicas de análisis pre-post. Test de Raven. Prueba de Wilcoxon.
<i>Proceso de configuración</i>		Plan de gestión de la configuración	Base de datos Diagramas Gantt
<i>Proceso de documentación técnica</i>		Plan de documentación técnica.	
<i>Proceso de documentación didáctica</i>		Plan de confección de la documentación didáctica.	
<i>Proceso de formación y capacitación del personal</i>		Plan de formación y capacitación	

Tabla 4. Listado de todos los procesos a tener en cuenta.

Las actividades que se debieron adicionar a la matriz dan idea de una aproximación constructivista-cognitivista, ya que en este caso para desarrollar el software experimental, se usó una teoría que posibilitara al sujeto activo tanto la construcción de sus propios aprendizajes al modo de Bruner (1988) y le permitiera la incorporación de nuevos conocimientos a las estructuras ya existentes de un modo permanente, jerárquico y relacional, es decir buscando la incorporación de conceptos nuevos de modo significativo y duradero. (Ausubel, 1983-1997).

Se podría también haber considerado al entorno, adicionando otras actividades, desde la visión de Vigotskii (1878) donde la construcción de significados es un proceso de internacionalización que implica la mediación por asociación de ideas cuyos significados fueron tomados desde el medio, en forma opuesta a lo que sostiene Piaget (1994) quien habla de un sujeto que construye significados en forma autónoma. (Pozo, 1994).

### ALGUNOS PUNTOS CRÍTICOS

Uno de los primeros puntos críticos que se encuentran al desarrollar un software educativo está relacionado con la necesidad del uso de un lenguaje común entre los miembros del equipo de desarrollo. Esto se debe a la heterogeneidad de sus disciplinas que incluye profesionales de diversas ramas de las ciencias de la computación y de las ciencias de la educación. La descripción de estos profesionales, agrupados en cuatro tipos principales se puede observar en la Tabla 5. Se busca una comunicación fluida y sin ambigüedades.

<i>Profesionales del área en la que se quiere desarrollar el software</i>	Profesores y especialistas en pedagogía para determinar los contenidos a incluir y expertos en el área de desarrollo
<i>Profesionales desarrolladores de software</i>	Analistas y programadores. Para el análisis del proyecto y la codificación.
<i>Coordinador del proyecto</i>	Como en todo proyecto soportado por una ingeniería de base, recaerá en el ingeniero de software.

<p><i>Personal técnico de apoyo (diseño gráfico y sonido)</i></p>	<p>De acuerdo a las dimensiones del desarrollo habrá operadores, diseñadores gráficos, especialistas en sonido, vídeo.</p>
---	--

Tabla 5: Perfiles de los integrantes del equipo de desarrollo

El segundo punto crítico, aparece al querer definir el significado de *interacción* con el software y el establecimiento de diferentes niveles de *interacción* cuando sea requerido. Según Fainholc (1998): "*La interacción se produce por la copresencia donde los sucesos tiene lugar y por la circularidad complementaria donde las percepciones y cogniciones se modifican formando parte de la presencia y conducta del otro*", y que en este caso son los programas didácticos.

En cambio la "*interactividad pedagógica*" que permiten los materiales didácticos, tiene en cuenta la medio, pudiéndose definir *interactividad*, que proviene del "*inter*" que significa "*entre nosotros*" y "*actividad pedagógica*" que es "*intervenir o interponer acciones didácticas para la elaboración de conceptos o el desarrollo de competencias, los que permitan comprender y transferir a la acción la esencia de los objetos implicados a fin de actuar apropiadamente*". (Fainholc, 1998)

Los software educativos permiten la *conversación didáctica* y la recreación de mensajes a lo largo del tiempo, integrando mensajes de tipos diferentes tales como escritos, verbales, visuales, auditivos. Constituyen un tipo de estructura de la comunicación didáctica, según Piaget dado que "*una estructura es un sistema de transformaciones que en tanto sistema se enriquece por el juego de las transformaciones*", una noción de totalidad, de transformación y de autorregulación o autoconstrucción constante. (Fainholc, 1998)

La interacción se da, por lo tanto entre los sujetos y la interactividad entre el sujeto y el medio, los diferentes niveles conversacionales permiten brindar a cada usuario un ritmo de trabajo personalizado.

El tercer punto crítico está centrado en el concepto de la calidad y en la polisemia de este término, ya que los programas educativos tienen características muy específicas ya que deben tener en cuenta, entre

otras cosas, las necesidades del grupo destinatario y los objetivos educativos. La calidad de los software educativos está relacionada con su grado de adaptación a un contexto en particular donde convergen una serie de variables, tales como las características curriculares, las edades del grupo destinatario, el estilo de aprendizaje y de enseñanza que se requieren para el análisis correcto. Un software educativo debería permitir evaluar cuatro ejes básicos: aspectos educativos pedagógicos y didácticos, aspectos técnico-económicos, aspectos comunicacionales y organizativos-estructurales del programa.

## LA DOCUMENTACIÓN

La documentación del programa tanto interna como externa, a lo largo del ciclo de vida del mismo es uno de los aspectos claves a considerar. Esto se debe a que la interna, que incluye los comentarios en los programas facilita los cambios y las modificaciones en etapas posteriores ya sea por el mismo grupo trabajo u otro.

La externa es aquella documentación que se refiere al material creado desde las etapas iniciales de análisis y que incluye todos los diagramas que muestran el flujo de procesos, el diseño de los módulos, etc. Estos permiten saber cómo está estructurado el programa y facilita los cambios y las actualizaciones.

Se debe señalar que los software deben incluir un manual o guía didáctica para asistir al usuario docente que debería contener los problemas más frecuentemente encontrados y las soluciones a los mismos. Por supuesto, que en este manual se deberá incluir toda la información pertinente al uso del programa. Deberá proveer además de ejemplos de uso, objetivos, contenidos, direccionamiento, actividades propuestas, teorías del aprendizaje consideradas y el tratamiento de los errores de los estudiantes durante el proceso de aprendizaje.

Sería aconsejable, incluir los resultados de las evaluaciones llevadas a cabo y los problemas señalados por los docentes, los aspectos positivos y negativos que éstos han encontrado con una estadística detallada de los mismos.

## LA EVALUACIÓN DEL SOFTWARE EDUCATIVO

Una vez establecida la metodología se preverá la emisión de los prototipos siguiendo la matriz de actividades que se puede encontrar en forma detallada en Cataldi Z., Lage F., et al. (2000b). En nuestro caso hemos emitido dos prototipos, los que fueron sometidos a evaluaciones con un grupo de alumnos del un posgrado en Tecnología, cuyas características eran similares a las del grupo destinatario del producto final. En cada caso se analizaron las sugerencias y se incorporaron las correcciones pertinentes. Luego, para terminar, se hizo una evaluación del producto final.

### a) La evaluación de los prototipos

Se tomó como caso para desarrollar el de un programa solicitado por los docentes y alumnos de una carrera de postgrado universitaria no Informática, para la asignatura "*Computación Aplicada*" : un software donde pudieran apreciar las partes de la computadora y en especial el funcionamiento interno de la misma, ya que se consideraba que una clase expositiva no era suficiente para las expectativas de los alumnos.

Se consideró oportuno realizar un programa en Delphi 3, y para la evaluación del mismo desarrollar varios prototipos con incrementos en las funcionalidades. Para ello, se le solicitó a un programador realizar el programa de acuerdo a la metodología propuesta en el capítulo anterior, con la ayuda de un especialista en contenidos del área temática. De este modo, se partió de la realización de un mapa de conceptos de este tema, como los desarrollados por Novak (1988), siguiendo un orden conceptual jerárquico del tipo árbol.

Las preguntas de la grilla de evaluación, básicamente consideraban aspectos de la interface de comunicación y de los contenidos desarrollados, debiendo ser valoradas con una escala de calificaciones entre 1 y a 5 (siendo 5: excelente 4: muy bueno 3: bueno 2: regular 1: malo o 5: muy adecuado 4: bastante 3: poco 2: muy poco 1: nada, de acuerdo al tipo de pregunta), pudiéndose obtener un valor promedio de las calificaciones para cada ítem.

Este valor permite obtener una puntuación de los aspectos tenidos en cuenta, para poder reformular o modificar aquellos que hayan tenido una puntuación menor que 2.5.

En todos los casos de evaluación se dispuso de un espacio abierto para las sugerencias al cambio o reflexión acerca del programa o de la situación de interacción.

Se evaluaron los dos prototipos y la versión final del software, siendo esta última evaluada también en forma interna, externa y contextualizada.

### a.1) Evaluación del Prototipo I

Para la evaluación del mismo, se tomó un grupo de 20 alumnos, recojiéndose los resultados cuantitativos en la tabla que se observa en el Apéndice I.

Para el primer prototipo, se consideró pertinente presentar un pre-diseño de las pantallas, el menú desplegable y el árbol de contenidos. Las imágenes y los vídeos y el sonido no estaban cargados aún. Se pensó en un diseño de interface del tipo Windows estándar, pero el programa en sí mismo mostraba muy poco. Las preguntas realizadas se pueden ver en la Tabla 6.

1. ¿Considera adecuado el diseño general de la pantalla?
2. ¿Considera adecuado el uso de las
Ventanas
Botones
Colores
Tipos de letras?
3. ¿Considera que el programa es interactivo?
4. ¿Considera la interface como amigable?
5. ¿Le da buena información acerca del recorrido?
6. ¿Considera criteriosa la secuenciación de las pantallas?
7. ¿Es de fácil manejo?
8. ¿Considera que el uso de los íconos es correcto?
9. ¿Le resulta útil el uso de teclas rápidas?
10. ¿Ha despertado interés en usted?
▪ Sugerencias de cambio Si- No

Tabla 6: Esquema para la evaluación del primer prototipo

Los resultados obtenidos, se pueden resumir cualitativamente ya que se puede decir que el diseño de la pantalla pareció adecuado, como las ventanas y los botones, pero no así con los colores utilizados y los tipos de letras. La interface pareció fácil de navegar y la secuenciación de las pantallas en general fue considerada como muy buena y de fácil manejo.

No hubo problemas en cuanto a la interactividad y despertó el interés y curiosidad en saber como sería el segundo prototipo del programa, ya con más funcionalidades incorporadas. Otra cuestión a señalar, fue que muchos desconocían la existencia de las teclas rápidas, lo que realmente no les interesaba.

Hubo una pregunta no ponderada y abierta, donde los alumnos que lo creían conveniente debían realizar sugerencias de cambio antes de pasar a una etapa posterior del desarrollo.

Las sugerencias se centraron básicamente en los siguientes puntos:

- Usar un tamaño de letra más grande de modo que fuera bien legible en una notebook.
- Cambiar los colores para que hubiera más contraste.
- Cambiar el puntero del mouse cuando se activaba un objeto de la pantalla.

## **a.2) Evaluación del Prototipo II**

De acuerdo a las preguntas ponderadas y las sugerencias realizadas incorporadas las mejoras pertinentes, se pasó al siguiente prototipo incrementando las funcionalidades. Se cargó el glosario con la terminología usada, las imágenes, algunos vídeos y la información acerca de cada una de las partes de la computadora, pudiéndose tener ahora una idea mucho más cercana a lo que sería el programa finalizado.

Para la nueva evaluación, las preguntas acerca de la interface comunicación fueron pocas y se precisaban los en aspectos relacionados a los contenidos y su pertinencia, se hizo mucho hincapié en la presentación de los mismos, la estructuración y la adecuación a las necesidades del grupo.

<ol style="list-style-type: none"> <li>1. Considera adecuada la selección de los contenidos?</li> <li>2. ¿Consideraría adecuado el uso del programa terminado en otros niveles?</li> <li>3. ¿Los cambios realizados fueron pertinentes?</li> <li>4. ¿Quisiera que el programa fuera un tutorial?</li> <li>5. ¿Le facilita la comprensión acerca del tema?</li> <li>6. ¿Quisiera sonido en los vídeos?</li> </ol> <p>Sugerencias de cambio Si- No</p>
--

Tabla 6: Esquema para la evaluación del segundo prototipo

En la tabla 6 se pueden observar el esquema de las preguntas efectuadas. En es Apéndice II, se adjuntan los resultados obtenidos con las ponderaciones y las sugerencias de los alumnos. Cabe destacar que a la mayor parte de los alumnos no les interesó en demasía considerar la realización de un programa del tipo tutorial, por lo que se aprecia que no le interesaba al grupo reemplazar totalmente al docente en sus explicaciones, sino usar el software como material de apoyo al docente y orientado a la ejercitación.

Respecto a las sugerencias de cambio, quizás la idea más relevante que expresaron fue que el programa finalizado les permitirá *“ver cosas que no hubieran imaginado”*.

### a.3) Evaluación de la versión final

En este caso se confeccionó una planilla con preguntas pertinentes a diferentes criterios, tomando como base la *utilidad*, los aspectos pedagógicos y didácticos y aspectos técnicos.

Las preguntas efectuadas observan en la Tabla 7, y se las debe ponderar como en los casos anteriores.

UTILIDAD	Facilidad de uso Grado de adaptación para otros niveles de usuarios.
PEDAGOGICOS y DIDACTICOS	Nivel de actualización Claridad de los contenidos

	Interface de navegación Nivel de motivación Adecuación para interpretar el tema Adecuación para aprender el tema
TECNICOS	Nivel de documentación y ayuda Adecuación de los recursos que se necesitan

Tabla 7: Esquema para la evaluación del Producto Final

Por último, se solicitaron algunas sugerencias a los usuarios, mediante un ítem abierto, ya sean para el uso del programa o para realizar algún cambio que se considere pertinente.

A partir de los resultados obtenidos, que se detallan en el Apéndice III, se puede observar que hubo aceptación y acuerdo respecto de los cambios producidos en las etapas anteriores.

#### **b) La evaluación interna**

El grupo que trabajó en el desarrollo del programa, estuvo de acuerdo con los cambios propuestos por los alumnos. Además consideró la pertinencia de las sugerencias.

Es importante destacar que un programa de esta naturaleza debe ser actualizado permanentemente, lo que implica un gran tiempo insu- mido en actualizar los contenidos.

Además se consideró la propuesta de los alumnos, de usarlo parale- lamente a las explicaciones del docente o de usarlo como apoyo a las clases de práctica y entrenamiento.

#### **c) La evaluación externa**

En este caso se presentó el programa a docentes de una carrera de Ciencias no Informáticas, quienes lo consideraron como una herra- mienta interesante a la hora de tener que profundizar los conocimien- tos acerca del tema. Se les proporcionó una planilla similar a las ante- riores, con preguntas cerradas y abiertas y se los proveyó del producto terminado.

Los resultados cuantitativos, se pueden ver en el Apéndice IV.

Cualitativamente consideraron la propuesta interesante y remarcaron que a veces el grado de dificultad que tienen los usuarios no informáticos para entender cómo funciona la máquina, es muy grande y que el programa le facilitaría la comprensión.

Otra de las consideraciones realizadas es que desde la escuela primaria se debería alfabetizar en informática, y que habría que pensarlo en lo sucesivo, ya que cada vez se hace más imprescindible el uso de la Computadora en casi todos los aspectos de la vida diaria.

**d) Qué medir al evaluar software educativo**

Considerando que el producto software educativo, de acuerdo a las necesidades de aplicación y a los objetivos educativos seguidos, requiere de características especiales, y además que éstas sean apropiadas en cuanto a calidad y pertinencia, se elaboró la tabla 8, partiendo del criterio de utilidad (en el sentido de amigabilidad) y analizando algunos subcriterios tal como se describen en la Tabla 8. A éstos se los puede calificar de acuerdo a tres niveles propuestos: muy bueno, bueno o malo, donde cada nivel tiene una puntuación. Al final de la evaluación el puntaje obtenido, estará entre los que se observan en la tabla 9, donde obviamente, se ve que un programa con una puntuación entre 21 y 30 puntos estará dentro de un nivel de calidad que definimos de aceptable.

El programa ha sido evaluado de acuerdo a la Tabla 8, obteniéndose un promedio de 22.1 para los 20 docentes evaluadores tal como se presenta en el Apéndice V.

UTILIDAD (falicilidad de uso & amigabilidad)	Sub criterios	Apreciación <i>Muy buen/a,</i> <i>Buen/a o Ma-</i> <i>lo/a</i>	Puntuación <i>3. Muy buen/a,</i> <i>2. Buen/a o</i> <i>1. Mal/a</i>	Puntuación obtenida
UTILIDAD EXTERNA (basada en [9])	Velocidad de aprendi- zaje ( <i>learnability</i> )			
	Facilidad de uso ( <i>operability</i> )			
	Nivel de adicción			

UTILIDAD INTERNA (basada en [9])	Nivel de legibilidad (Lecturability)			
	Grado de entendimiento			
	Estructura de los manuales			
	Uso de menús, gráficos e imágenes			
	Mensajes de error y de información			
	Ayudas On-line			
	Definición de la adaptación de la interface			
	Puntuación obtenida			

Tabla 8: Criterios y sub criterios

Puntuación	Evaluación de la Propuesta	CALIDAD
1-10	Mala	INACEPTABLE
11-20	Regular	DUDOSA
21-30	Buena	ACEPTABLE

Tabla 9. Esquema de puntuación

### e) Evaluación contextualizada

Finalmente se efectuó la evaluación contextualizada con un grupo de estudio similar al de los usuarios potenciales, teniendo en cuenta las variables involucradas en la situación áulica. (Cataldi, et al, 2000 c y d).

Este tipo de evaluación proporciona los resultados más representativos ya que dan cuenta de las reacciones de los potenciales usuarios ante el programa y dan cuenta de la eficacia del producto. (Fainholc, 1998).

Para ello, se tuvieron en cuenta las variables involucradas en el proceso de enseñanza y de aprendizaje tales como la tipología y el estilo docente, tipo de alumnos destinatarios, el tiempo y modo de uso del software, el currículo, entre otras.

Esta experiencia se efectuó a fin de validar la metodología, al menos parcialmente, ya que se deben considerar las limitaciones al trabajar

con una *metodología*, con *grupos humanos reducidos* y *durante un período de tiempo limitado*.

Lo que se intenta fue demostrar que: *“El software educativo desarrollado con una metodología que contempla os aspectos pedagógicos en el modelo del ciclo de vida permiten un mejor aprendizaje de los conceptos que un software que ha sido desarrollado con una metodología que no los contempla”*.

Primeramente, se formularon y se describieron las etapas preparatorias y, posteriormente describe la experiencias realizadas a fin de establecer las diferencias en cuanto a logro de aprendizajes significativos entre un software desarrollado con una metodología extendida para cautelar los aspectos pedagógicos, partiendo de una paradigma clásico de la ingeniería de software (Cataldi el al., 2000) y uno de idéntica funcionalidad desarrollado con una metodología estándar.

Para la experiencia, se formaron dos grupos equilibrados mediante la definición de pares homólogos: uno de control, llamado A y otro experimental ó B.

La idea de la creación de pares homólogos surgió a raíz de la necesidad de llevar a cabo una experiencia con dos grupos equilibrados cuya respuesta ante los aprendizajes de los conceptos fuera similar. Para definir grupos equilibrados, se partió de la aplicación del test de matrices progresivas de Raven (1979) a los sujetos.

Luego de una evaluación diagnóstica inicial, ambos grupos, en conjunto recibieron la misma instrucción acerca de los aspectos teóricos, mediante clases expositiva, siendo el tema desarrollado: el funcionamiento interno de una computadora personal. Luego, al grupo de control "A" se le mostró aspectos inherentes a la lógica de funcionamiento mediante un software desarrollado con una metodología que no contemplaba los aspectos pedagógicos y al grupo experimental "B", mediante un software desarrollado con la metodología extendida.

Para corroborar la hipótesis, se llevaron a cabo las etapas siguientes:

1. Se tomó un curso de Computación de una carrera no informática de la Facultad de Ingeniería.

2. Se aplicó el Test de Raven de Matrices Progresivas (1979), formándose pares homólogos en dos grupos: A ó de control y B ó experimental. (Se podría haber utilizado bien una evaluación diagnóstica a fin de determinar un estado inicial de base).
3. Se tomó un contenido y se explicó el tema teórico a ambos grupos. *El grupo A se ejercitó con un software desarrollado con una metodología que no contemplaba los aspectos pedagógicos y el grupo B se ejercitó con el software desarrollado con la metodología extendida propuesta, tomando como base la teoría del aprendizaje significativo y los mapas conceptuales.* Luego, los dos grupos fueron evaluados con el mismo conjunto de preguntas.

Cabe destacar y es observable que, con independencia de las variables ocultas, se entiende que la única variable se controlaba en el experimento es la metodología, ya que los contenidos son los mismos, la ejercitación se hace con software, la variable por lo tanto es la *metodología de diseño*. (Cataldi, Lage , et al. (2000d).

Una vez realizadas las experiencias, se verificó el rendimiento de los alumnos mediante la aplicación de la misma evaluación para los dos grupos. Esta prueba fue diseñada específicamente por un profesional de ciencias de la educación, para saber si los alumnos habían logrado la incorporación de los conceptos nuevos en su estructura cognitiva y podían transferir (nos referimos a transferencia cercana debido al poco tiempo disponible para la realización de las experiencias) desde otras áreas del saber y hacia otras áreas, si podían establecer algunas de las *actividades de comprensión* o "*procesos de pensamiento*" previstas en la programación didáctica tales como: explicar relaciones causa efecto, formular conclusiones válidas, describir limitaciones de los datos, confrontar conocimientos nuevos con previos, clasificar y seleccionar información, producir, organizar y expresar ideas, elaborar mapas conceptuales (teniendo en cuenta la reconciliación integradora y la diferenciación progresiva), integrar el aprendizaje en diferentes áreas, inferir correctamente, evaluar el grado de adecuación de las ideas, presentar argumentos pertinentes frente a fenómenos, defender un punto de vista y fundamentar criterios, resolver problemas elaborando estrategias metacognitivas, entre otras.

Con esta evaluación se intentó tener una medida acerca de la comprensión del tema y sus relaciones, lo que implica el compromiso reflexivo del alumno con el contenido de enseñanza y la habilidad para articular significativamente el material comunicado por acciones de guía (Cedipro, 1998).

Finalmente, como una medida indirecta de ello, se aplicó a las notas obtenidas por ambos grupos un test estadístico de comparación para muestras pequeñas, obteniéndose las conclusiones que se enuncian.

## RESULTADOS EXPERIMENTALES

A partir de estas observaciones se puede reformular la hipótesis del modo siguiente: *“Siendo los valores de las variables independientes los mismos para ambos grupos, salvo la variable programa utilizado para la ejercitación; el grupo de alumnos que trabaje con el programa desarrollado con la metodología propuesta debe tener un mejor rendimiento que el grupo de alumnos que utilice el programa desarrollado con la metodología convencional”*.

Habiendo realizado la evaluación a ambos grupos y aplicando el test de Wilcoxon (Ledesma, 1980; Universidad de Málaga, 1999), para muestras pequeñas, a los resultados obtenidos por los pares homólogos, se observa una diferencia significativa a favor del grupo experimental ó B, por lo que este resultado verifica la tesis.

## CONCLUSIONES

Los aportes de este trabajo consisten en la identificación de deficiencias en las metodologías de diseños de software educativo, en cuanto a los aspectos pedagógicos que deben ser cubiertas y como solución a este problema se seleccionó una metodología de ingeniería de software, ampliándose la definición de sus fases para dar solución al problema identificado.

Esto condujo a la elaboración de un software a fin de probar tanto la metodología como las evaluaciones que se describen a fin de “probarlo” experimentalmente, observándose que un software educativo desarrollado con la metodología propuesta extendida da “mejores” resultados que uno desarrollado con una metodología convencional de ingeniería de software.

Quedaría por considerar la reacción de los alumnos más allá del efecto novedad que cita Cabero (2000), ya que quizás la incorporación de la tecnología produce un efecto estimulante y, luego corroborar la investigación experimental a fin de desechar posibles distorsiones de los resultados o sesgos.

Finalmente, de acuerdo a los resultados obtenidos, se han entregado copias del trabajo de investigación a los docentes a quienes se entrevistó inicialmente a fin de saber si la metodología responde a sus expectativas durante un período de dos años, luego del cual deberán elevar un informe con sus resultados.

### **FUTURAS LÍNEAS DE INVESTIGACIÓN**

Se prevé explorar la ampliación de otras metodologías de Ingeniería de Software con base en otras teorías educativas y orientación a objetos a fin de comparar las distintas metodologías extendidas, y diseñar estrategias de infoalfabetización para docentes que deseen aplicar y/o diseñar software educativo utilizando estas metodologías.

Se piensa también en la creación de ambientes integrales de trabajo-estudio “diferenciadores” (Cabero, 2000) que permitan la estimulación de los diferentes sistemas simbólicos de los usuarios incorporando nuevas actividades adicionales en el ciclo de vida.

APÉNDICES

Apéndice I: Planilla de evaluación de la interface de comunicación - Prototipo I

Número de orden	Calificación de 1 a 5																				Prom
	(5: excelente 4: muy bueno 3: bueno 2: regular 1: malo)																				
	(5: muy adecuado 4: bastante 3: poco 2: muy poco 1: nada)																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
1. Considera adecuado el diseño general de la pantalla?	4	4	3	3	3	3	3	3	3	3	4	4	3	4	4	3	3	4	4	3	3.4
2. Considera adecuado el uso de las	Ventanas	5	4	4	4	4	4	4	4	4	4	4	4	4	4	3	4	3	4	4	3.95
	Botones	4	4	3	4	4	4	3	4					3	3	4	3	4	4	4	2.33
	Colores	4	1	2	2	2	2	2	2	2	3	3	2	2	3	3	2	2	2	2	2.25
	Tipos de letras?	3	4	2	2	4	4	3	4	3	2	3	3	3	3	3	3	2	2	2	2
3. Considera que el programa es interactivo?	5	4	4	4	2	3	2	3	4	3	4	4	4	3	3	3	4	4	4	4	3.55
4. Considera la interface como amigable?	4	4	4	3	4	4	4	4	4	4	4	4	4	4	4	3	4	3	3	3	3.6
5. Le da buena información acerca del recorrido?	5	5	4	5	4	4	4	5	4	4	4	4	4	4	4	4	4	5	5	5	4.15
6. Considera criteriosa la secuenciación de las pantallas?	4	4	4	3	4	4	4	4	5	4	5	5	5	4	4	4	3	3	3	4	4.0
7. Es de fácil manejo?	4	5	5	4	5	5	5	4	5	4	5	5	4	5	4	5	5	5	5	5	4.7
8. Considera que el uso de los íconos es correcto?	4	5	4	5	5	5	5	4	4	3	4	5	4	4	4	4	4	4	4	4	4.25
9. Le resulta útil el uso de teclas rápidas?	3	3	3	5	-	-	-	4	-	-	-	-	-	-	-	-	3	-	4	4	3.62
10. Ha despertado interés en usted?	4	4	3	5	3	4	3	4	4	4	4	4	4	4	5	5	4	4	3	4	3.95
Sugerencias de cambio S/N	N	S	S	S	S	S	S	N	-	-	S	S		S	S	S	S	S	S	S	

REVISTA LATINOAMERICANA DE TECNOLOGÍA EDUCATIVA  
Volumen 2. Número 1

<i>Número de orden</i>	<b>Sugerencias de cambio</b>
1	Cambiar los colores en pantalla para que resulte más atractivo
2	Para evaluar mejor tendría que estar más completo, sugiero mayor colorido y que el indicador sea distinto. Es decir no flecha sino mano.
3	Cuando se posiciona en algo que se expande que aparezca otro apuntador como la mano, y que a la vez haya cambio de relieve. Ponerle sonido al momento de activar algo. Más vistoso, más atractivo.
4	Cambio de colores - más contraste
5	Interconectar los elementos por medio de dibujos que representen cables en la pantalla de presentación. Tipo diagrama de flujo
6	
10	
15	Cambiar la flecha de indicación por la manito
16	Tamaño de letra más grande para usar en la notebook.
16	Cambiar los colores de pantalla para que resulte más contraste.
18	Hace falta un icono de retorno en la pantalla. Tamaño y tipo de letra
18	Sería bueno agregar algún tapiz de fondo por que parece que los elementos en la pantalla están flotando?
19	Las letras de los botones son muy débiles, les falta fuerza.
19	Colocar el icono de "volver" fijo en la pantalla para retornar a ventanas anteriores.
20	Mejor color de pantalla, letra más gruesa, crear subventanas u opciones de algunos temas

*Apéndice II: Evaluación de los contenidos y su pertinencia - Prototipo II*

<i>Número de orden</i>	<b>Calificación de 1 a 5</b>																				<b>Prom</b>
	(5: excelente 4: muy bueno 3: bueno 2: regular 1: malo)																				
	(5: muy adecuado 4: bastante 3: poco 2: muy poco 1: nada)																				
<i>Número de orden</i>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	<b>Prom</b>
1. ¿Considera adecuada la selección de los contenidos?	4	5	3	5	4	5	4	3	5	3	4	4	4	5	4	5	3	4	4	4	4.1

2. ¿Consideraría adecuado el uso del programa terminado en otros niveles?	4			5	5	5	5	5	4	4	4	4	4	5	5	4	4	4	4	4	5	5	4.42
3. ¿Los cambios desde realizados fueron pertinentes?	4	4	4	4	4	4	4	4	5	5	5	5	5	4	4	4	3	3	4	4	4	4	4.15
4. ¿Quisiera que el programa fuera un tutorial?	2	3	4	3	3	3	2	2	2	3	3	3	3	3	3	3	3	3	3	3	4	3	2.9
5. ¿Le facilita la comprensión acerca del tema?	4	4	4	4	5	5	4	4	5	5	4	4	4	4	4	4	4	4	4	5	5	5	4.35
6. ¿Quisiera sonido en los vídeos?	3	3	3	4	4	3	3	3	3	4	4	4	3	3	3	3	3	3	3	3	3	3	3.75
Sugerencias de cambio Si- No	-	S	-	S	S	-	S	-	S	-	S	-	S	-	-	-	-	-	-	-	-	-	

NÚMERO DE ORDEN	Sugerencias de cambio
1	Poner las direcciones web y la bibliografía en las pantallas explicativas.
2	Ahora puedo evaluar mejor la capacidad del programa y me lo imagino terminado
3	Sugiero que no se le ponga audio, ya que me gusta tener al docente que me explique lo que pasa en el video y así puedo hacerle preguntas. Sólo lo usaría para los eventos.
6	Considero que los cambios estuvieron bien y que respeten en los programas el formato de Windows
10	Los colores siguen siendo muy pálidos
16	El problema es que hay que actualizarlo constantemente.
19	Me parece bueno el programa porque me permite ver cosas que no me hubiera imaginado y que desconocía

Apéndice III: Evaluación del producto Final

		Calificación de 1 a 5																				Prom	
		(5: excelente 4: muy bueno/a 3: bueno/a 2: regular 1: malo/a) (5: muy adecuado 4: bastante 3: poco 2: muy poco 1: nada)																					
Número de orden		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20		
Aspecto																							
Utilidad	1.Facilidad de Uso	5	5	5	5	4	5	4	5	5	5	5	4	4	4	4	4	4	4	4	4	4.75	
	2.Grado de adaptación a otros niveles de usuarios.	4	5	4	5	4	5	5	5	5	5	4	5	5	4	4	4	5	5	5	5	4.65	
Pedagógicos y didácticos	3.Claridad de contenidos	4	4	4	3	3	3	4	4	3	4	3	3	3	3	3	4	4	4	4	4	3.55	
	4.Nivel de actualización	5	5	5	5	5	5	4	4	5	5	5	4	4	4	4	4	4	4	4	4	4.45	
	5.Interface de navegación	4	4	4	4	3	3	3	3	4	4	3	3	3	3	3	3	3	3	3	3	3.3	
	6.Nivel de Motivación	4	4	4	3	3	3	3	4	4	4	4	4	4	4	4	4	4	3	3	3	3.8	
	7.¿Es adecuado para la comprensión del tema?	3	5	5	5	4	3	3	3	4	4	5	5	5	3	4	4	3	3	3	3	4	3.85
	8.¿Es adecuado para el aprendizaje del tema?	3	5	5	5	3	3	3	4	4	4	4	3	5	5	5	4	4	4	3	3	3	3.9
Técnicos	9.Documentación y ayudas	4	4	4	4	5	5	5	4	4	3	5	5	5	5	4	4	4	3	3	3	4.05	
	10.¿Son adecuados los recursos que necesita?	5	5	5	5	5	3	3	4	5	5	5	5	5	5	5	5	3	3	3	3	4	4.1
Sugerencias S/N		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		

Número de orden	Sugerencias de Cambio
1	Los colores más fuertes.
5	Las letras más grandes

10	Habría que considerar usarlo en otros cursos
12	No me interesan las teclas rápidas.
14	Habría que poner más ventanas.
15	Me vino muy bien por que yo no sabía nada
16	Quizás sería bueno, proyectarlo mientras el docente explica

Apéndice IV: Evaluación Externa del Producto Final

Número de orden	Calificación de 1 a 5																				Prom
	(5: excelente 4: muy bueno 3: bueno 2: regular 1: malo)																				
	(5: muy adecuado 4: bastante 3: poco 2: muy poco 1: nada)																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
1. ¿Considera adecuado el diseño general de la pantalla?	4	4	4	4	4	3	4	4	5	5	5	5	5	5	5	5	5	5	5	5	4.55
2. ¿Considera adecuado el uso de	Ventanas	3	3	4	4	4	4	4	5	5	5	5	5	4	5	4	5	4	5	4	4.3
	Botones	4	4	4	4	4	4	5	5	5	5	5	5	4	4	4	4	4	4	4	4.3
	Colores	3	3	3	3	4	4	4	4	3	3	3	3	4	4	4	4	4	4	4	3.6
	Tipos de letras?	4	4	4	4	4	3	3	3	4	3	3	4	4	4	4	4	4	4	4	3.75
3. ¿Considera que el programa es interactivo?	4	4	4	4	4	4	4	4	5	5	4	4	4	4	4	4	4	4	4	4	4.1
4. ¿Considera la interfaz como amigable?	4	4	4	4	4	4	5	5	5	4	4	5	5	4	4	4	4	4	4	4	4.25
5. ¿Le da buena información acerca del recorrido?	5	5	5	5	4	4	4	4	5	5	5	5	5	5	5	5	5	5	5	5	4.8
6. ¿Considera criteriosa la secuenciación de las pantallas?	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
7. ¿Es de fácil manejo?	5	5	5	5	4	4	5	5	5	4	4	4	4	4	4	5	5	5	5	5	4.25
8. ¿Considera que el uso de los íconos es correcto?	5	5	5	5	5	4	4	4	4	4	4	5	5	5	5	5	5	5	5	5	4.7
9. ¿Le resulta útil el uso de teclas rápidas?	3	3	3	3	2	-	-	2	2	2	-	-	-	2	2	-	2	2	2	-	2.3

10. ¿Considera adecuada la selección de los contenidos?	4	4	4	4	4	4	4	5	5	5	4	4	4	4	4	5	4	5	4	5	4	4.05	
11. ¿Consideraría adecuado el uso del programa terminado en otros niveles?	4	4	4	4	4	4	4	4	4	4	4	4	5	5	4	5	4	5	5	4	4	4	
12. ¿Quisiera que el programa fuera un tutorial?	4	4	4	-	-	-	3	3	3	-	-	-	-	-	3	3	3	-	-	-	-	3.33	
13. ¿Le facilita la comprensión acerca del tema?	4	4	4	4	3	3	3	3	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4.15
14. ¿Quisiera sonido en los vídeos?	5	2	3	4	4	3	3	2	2	2	2	3	3	-	-	3	-	-	3	3	-	2.37	
15. ¿Ha despertado interés en usted?	5	5	5	5	5	4	4	4	4	3	3	3	-	5	5	5	-	5	-	5	-	4.41	
Sugerencias de cambio Si-No	S	-	S	-	-	-	S	-	-	-	S	-	-	S	-	-	S	-	-	S	-		

*Apéndice V: Aplicación de Criterios y Subcriterios de Calidad*

Número de	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Pr.
Puntaje Obtenido	20	21	23	20	23	20	24	25	20	21	23	22	22	22	23	21	17	24	26	25	22.1

**REFERENCIAS BIBLIOGRÁFICAS**

Ausubel, D.; Novak, J. y Hanesian, H.(1978). *Psicología educativa. Un punto de vista cognitivo*. Trillas. Ediciones 1978, 1997.

Boehm, B. (1981). *Software Engineering Economics*. Englewood Clifs, Nueva Jersey.

Boehm, B. (1988). A spiral model of software development and enhancement. *Computer* 1988 IEEE , 61-72.

Bruner, J. (1988). *Desarrollo cognitivo y educación*. Madrid: Morata.

- Bruner, J. (1991). *Actos de significado. Más allá de la revolución cognitiva*. Madrid: Alianza.
- Cabero, J. (1992). Diseño de Software Informático. *Bordón*, 44 (4), 383-391.
- Cabero, J. (2000). *Tecnología Educativa*. Madrid: Editorial Síntesis.
- Cataldi Z., Lage F., et al. (2000a). *Methodology of design and development of educational software from a pedagogical perspective*. Anales de ICECE 2000 International Conference on Engineering and Computer Education. 27-30 de agosto. San Pablo. Brasil.
- Cataldi Z., Lage F., et al.(2000b). *Criterios de calidad de Ingeniería de Software aplicados al software educativo*. Proceedings del VI Congreso Internacional de Ingeniería Informática ICIE 2000. 26-28 de abril.
- Cataldi Z., Lage F., et al. (2000c). *Evaluation of educational software from an integral perspective*. Cacic 2000, VI Congreso Argentino de Ciencias de la Computación, 2-7 de Octubre, Ushuaia. IE002.
- Cataldi Z., Lage F., et al. (2000d). *Evaluación contextualizada de software educativo*. Cacic 2000, VI Congreso Argentino de Ciencias de la Computación, 2-7 de octubre, Ushuaia. IE004.
- Coll, C. (1994). *Psicología y Curriculum*. Barcelona: Paidós.
- Cruz Feliú, J. (1986). *Teorías del Aprendizaje y Tecnología de la Enseñanza*, Trillas.
- Fenton N. (1991). *Software Metrics. A rigorous and practical approach*. Boston: PWS Publishing Company.
- Gardner, H. (1987-8). *La nueva ciencia de la mente: Historia de la psicología cognitiva*. Barcelona: Paidós.
- Gardner, H. (1993). *Las Inteligencias Múltiples: La teoría en la práctica*. Barcelona: Paidós.
- Gardner, H. (1995). *La mente no escolarizada*. Barcelona: Paidós.
- Johnson-Laird, P. N. (1998). *El ordenador y la mente: introducción a la ciencia cognitiva*. Barcelona: Paidós.
- Juristo J., N. (1996). *Procesos de construcción del software y ciclos de vida*. Madrid: Universidad Politécnica de Madrid.
- Ledesma, D. A. (1980). *Estadística Médica*. Eudeba.

- Manual de la Universidad de Málaga. Bioestadística: Métodos y Aplicaciones. Facultad de Medicina. [www.ftp.medprev.-uma.es/libro-/node148.htm] consultado el 28/9/99 a las 10 hs.
- Marquès, P. (1998a). La evaluación de programas didácticos. *Comunicación y Pedagogía*, 149, 53-58.
- Marquès, P. (1998b): *Programas didácticos: diseño y evaluación*. Universidad Autónoma de Barcelona. Consultado en octubre de 1998. [www.doe.d5.ub.es/te]
- Perkins, D. (1995). *La Escuela Inteligente*. Barcelona: Gedisa
- Piaget, J. (1989). *La construcción de lo real en el niño*. Barcelona: Crítica-Grijalbo.
- Piattini, M. (1996). *Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión*. Rama. Madrid.
- Pozo Muncio, I: (1998). *Aprendices y Maestros*. Madrid: Alianza.
- Pressman, R. (1993). *Ingeniería de Software. Un enfoque práctico*. Madrid: McGraw Hill.
- Raven, J. C. (1979). *Test de Matrices Progresivas. Escala General*. Vol. 3b. Buenos Aires: Paidós.
- Raven, J. C. (1979). *Test de Matrices Progresivas. Manual para la Aplicación*. En J. Sánchez y O. Alonso (1997-8). *Evaluación distribuida de software educativo a través de Web*. [www.dcc.uchile.cl/~oalonso/educacion]
- Sancho, J. (1994). *Para una Tecnología Educativa*. Barcelona: Editorial Horsori.
- Schunk. D. H. (1997). *Teorías de la Educación*. Madrid: Prentice Hall.
- Skinner, B. F., (1958, 1963). *Teaching Machines, Science*, publicado en 1958; *Reflection on a decade of teaching Machines*, publicado en 1963, citados por Cruz Feliú, Jaime (1986) en *Teorías del Aprendizaje y Tecnología de la Enseñanza*, Trillas.
- Vigotzki, L. (1978). *Mind in Society. The development of higher psychological process*. Cambridge, MA.: Harvard University Press.