

|Cuatro Modelos para Evaluar el Software Educativo¹

Dr. Carlos Vargas Castillo
Universidad de Costa Rica
Escuela de Ciencias de la Computación e Informática
cvar@costarricense.cr

RESUMEN

Se investigan aquellas teorías que orientan la evaluación del software educativo. Para ello, se identifican los constructos teóricos importantes para formular marcos de referencia que fundamenten la evaluación del software educativo. Se analizan cuatro propuestas teórico-metodológicas representativas de los diversos enfoques utilizados para evaluar el software educativo.

Palabras claves: Informática Educativa, evaluación de software educativo.

(CACIC 2003: II Workshop de Tecnología Informática Aplicada en Educación).

1. Introducción

La informática educativa se introdujo masivamente en Costa Rica, desde la segunda mitad de los ochenta, vía la Fundación Omar Dengo (Fonseca,1991). Primeramente, se introdujo en la enseñanza primaria del sector público. El componente informático en que se basó, fue originalmente el LOGO, lenguaje desarrollado en el *Instituto Tecnológico de Masachussets*, e impulsado por Papert (1980; 1987), el cual propicia un enfoque educativo constructorista. Posteriormente, el ambiente evolucionó hacia Micromundos, herramienta de software que integra el LOGO con elementos multimediales y con la Internet.

En Costa Rica, diversas instituciones e individuos han contribuido con investigaciones relacionadas con la informática educativa, con producción de herramientas de software educativo, y con material didáctico en formato digital. Se destacan el Instituto para el Mejoramiento de la Educación Costarricense (IIMEC), y la Escuela de Ciencias de la Computación e Informática (ECCI), ambos de la Universidad de Costa Rica (UCR), Universidad Estatal a Distancia (UNED), la Universidad Nacional (UNA), el Instituto Tecnológico de Costa Rica (ITCR), el Programa de Educación Ambiental del INBio y la Fundación Omar Dengo.

En un estudio que realizamos para conocer las orientaciones y tendencias en desarrollo de software educativo en Costa Rica (Vargas, 2002), se abarcó la producción de software educativo

¹ Elaborado dentro del proyecto **Desarrollo de Software Educativo**, y el **Programa de Cooperación Académica para el uso de las tecnologías de información en el mejoramiento de la docencia universitaria**, ambos inscritos ante las Vicerrectorías de Investigación y de Docencia de la Universidad de Costa Rica.

desarrollada desde 1990, por las universidades estatales costarricenses. Se comprobó una tendencia creciente en el desarrollo de aplicaciones de software educativo, pero también se evidenció que alrededor de la mitad del software producido no sufrió un proceso de evaluación. Se percibe que se tiende a darle gran importancia al desarrollo y utilización del software educativo, pero con menor énfasis en la evaluación del mismo.

En este artículo se ofrece una visión amplia acerca de los diversos marcos teórico-metodológicos que sustentan la evaluación de software educativo. Nuestra premisa es que, no solamente es tarea requerida el evaluar el software educativo, pero además, que es importante que el equipo de desarrollo, conozca y establezca de modo explícito, desde cual marco conceptual se persigue evaluar el software, previo al diseño de los procesos e instrumentos para su realización práctica. Para ello, en este artículo identificamos los modelos paradigmáticos que pueden ser adoptados para la evaluación de software educativo.

2. Nociones preliminares

La evaluación de software educativo se centra en dos momentos: durante el proceso de diseño y desarrollo, con el fin de corregir y perfeccionar el programa; y, durante su utilización real por los usuarios, para juzgar su eficiencia y los resultados que con él se obtienen.

Al primer momento se le designa evaluación interna o formativa. Zaragoza y Cassado(1992) indican que una vez que se haya elaborado un primer prototipo del software educativo, o sea una versión preliminar, éste es sometido a evaluación: “Una primera evaluación interna, realizada por los miembros que han intervenido en la fase de prediseño y diseño; es decir, los maestros y los programadores, que lo examinan desde un punto de vista crítico, sometiéndolo a situaciones límite de manera parecida a como pueden hacerlo los alumnos.”(p.79). Similarmente, Insa y Morata (1998) definen la evaluación formativa: “La evaluación formativa de un sistema de Enseñanza Asistida por Ordenador se lleva a cabo antes de dar salida al producto final, es decir, es la que se hace con un prototipo y el control de calidad a que se somete la aplicación antes de ser presentada a los usuarios. Esta evaluación se centra en el proceso de desarrollo del programa, en especial en los aspectos técnicos, comprobando que todos los elementos del programa funcionen correctamente” (p.200).

Al segundo momento, se le designa como evaluación externa o sumativa, que consiste en “Una segunda evaluación Externa, realizada por personas que no han estado implicadas directamente en el proyecto. El perfil de estas personas, normalmente se identifica con el del grupo de alumnos de la clase” (Zaragoza y Cassado, 1992, p.79). Por su parte, Insa y Morata (1998) indican que “La evaluación sumativa se realiza una vez que se ha implementado el sistema con los usuarios a los que estaba destinado. Su cometido es comprobar los efectos de la aplicación y en que medida se han conseguido los objetivos y finalidades propuestos desde la perspectiva del aprendizaje” (p.201).

Una vez que el software ha pasado y superado la evaluación formativa y sumativa está en condiciones de comenzar a ser utilizado, aunque siempre queda abierto a revisión para poder reconducir algunos aspectos, si fuera necesario.

Desde los comienzos de la informática educativa surgió la necesidad de establecer criterios prácticos que ayudasen a los profesores a evaluar el software educativo. Muchas de las primeras publicaciones sobre el tema presentaban criterios que formular sobre las características del software. Las listas de criterios se formulan como preguntas, las cuales se suelen agrupar por temas o categorías, de acuerdo con las preferencias pedagógicas de los proponentes. Vivet (1996) distingue cinco categorías de evaluación, progresivamente más cercanos al interés del docente: la tecnología de ese software, el material instructivo que contiene, el software educativo integrador de las dos primeras categorías, la enseñanza, y el aprendizaje.

Esta forma sistemática utilizada para la valoración de software educativo se materializa en un instrumento al que suele llamársele lista de control (*check-list*). Aunque las listas de control ofrecen una técnica fácil de utilizar, no constituyen por sí mismas un marco teórico de referencia, debido a algunas limitaciones importantes que se le señalan, tales como que las listas de control no prevén diferentes estrategias docentes o que se le otorga igual ponderación a todos los ítems (Squirres y McDougall, 1997).

Para Heller (1991), aunque la revisión formal con listas de control es deseable, presenta ciertas limitaciones en cuanto a la selección de software más reciente e innovador: “Incluso las revisiones formales no constituyen una respuesta definitiva a la revisión del software. Sobre todo en el mundo actual, en el que los nuevos enfoques tecnológicos, como los hipermedia y la realidad artificial, pueden llevar al aula nuevas formas de software educativo, es muy posible que un programa sea excelente de un modo no contemplado en las orientaciones para la selección” (p.286).

En la actualidad existe una enorme cantidad de listas de control desarrolladas por individuos, escuelas, organizaciones y empresas comerciales (Santiago, 1998; Gutiérrez y Levis, 2000; Harasim y otros, 2000; Duart y Sangrá, 2000). La Internet brinda fácil acceso a muchas de esas listas.

Por las razones apuntadas, se requieren marcos teóricos más generales que las listas de control, que sirvan como fundamentos más comprensivos para la evaluación de software educativo. En la siguientes secciones se analizan cuatro marcos teóricos, representativos de distintas posiciones pedagógicas y epistemológicas imperantes.

3. Enfoque crítico

Un marco teórico para evaluar el software educativo, orientado por la teoría crítica, lo desarrolló Kemmis y sus colaboradores, tomando como punto de partida el trabajo de clasificación previamente realizado por Kemmis y otros (1977), durante una evaluación a gran escala del *National Development Programme in Computer Assisted Learning* británico. La clasificación se basa en la proposición de cuatro paradigmas de la educación denominados: instructivo, revelador, de conjeturas y emancipador.

Paradigma instructivo.

Busca el dominio del contenido. La instrucción se racionaliza mediante técnicas como el establecimiento de la sucesión de los aprendizajes, la presentación y el reforzamiento mediante la información sobre las consecuencias. El software relacionado con este paradigma pretende explícitamente enseñar determinado material dividiéndolo en partes más pequeñas y presentándolas a los estudiantes. El alumno tiene oportunidad de responder a preguntas recibiendo información sobre la precisión de sus respuestas. El software se encarga de presentar los contenidos, proporcionar la información sobre los resultados y asignar tareas. Ejemplos de este software son los programas de ejercicio y reconocimiento de problemas aritméticos y los sistemas inteligentes de aprendizaje dirigido.

Paradigma revelador.

Resalta el aprendizaje por descubrimiento y el desarrollo de la intuición respecto al campo de estudio. El software se utiliza para que proporcione ambientes para la exploración y el descubrimiento. Ejemplos típicos de este software son los programas de simulación, en que se simula un ambiente real, que plantea situaciones que de otro modo no podrían estudiarse en el aula. Los estudiantes investigan el modelo introduciendo datos y observando los resultados que se producen, de modo que las ideas y conceptos importantes comienzan a surgir naturalmente.

Paradigma de conjeturas.

Está relacionado con la articulación y la manipulación de ideas y con la comprobación de hipótesis. Destaca el desarrollo de la comprensión mediante la construcción activa del conocimiento. El software aporta ambientes en los que se articulan y exploran ideas mediante la creación de modelos, programas, planes y estructuras conceptuales. Ejemplo típico de este software son los micromundos informáticos que dejan que los estudiantes expresen sus ideas y construyan soluciones para los problemas modificando el estado de un objeto a través de la programación.

Paradigma emancipador.

Tiene relación con los programas que explotan la capacidad del computador para procesar grandes masas de datos y realizar muchas operaciones a gran velocidad, de manera que los estudiantes no pierdan tiempo en tareas tediosas, necesarias pero no relevantes para el aprendizaje. Por ejemplo los cálculos aritméticos para analizar datos de experimentos en ciencias. Este paradigma no está relacionado con ningún modelo educativo, pero apoya a los paradigmas anteriores. Los programas emancipadores evitan el trabajo no auténtico, aquel que no aporta nada a la tarea educativa pero que hace falta realizar.

El marco de referencia propuesto evalúa las relaciones entre los cuatro paradigmas mencionados y una tipología de las interacciones del estudiante en el aprendizaje asistido por computador. Ésta tipología se compone de cinco clases de interacción: reconocimiento, recuerdo, comprensión reconstructiva o intuitiva global y comprensión constructiva. La Figura No. 1 representa el alcance de las interacciones entre cada paradigma y los diversos tipos de interacción, indicando que existen correlaciones entre cada paradigma y los distintos tipos de procesos de aprendizaje. El enfoque ofrece un marco de referencia global que ayuda al pensamiento y al análisis crítico sobre el software educativo.

PARADIGMA	Reconocimiento	Recuerdo	Comprensión reconstructiva	Intuitiva global	Comprensión constructiva
Instructivo			← ● →		
Revelador				← ● →	
De conjeturas					← ● →
Emancipador					

Figura No.1: Relaciones entre paradigmas y tipos de interacción (Squires y McDougall, 1997)

4. Enfoque cualitativo basado en interacciones

Squires y McDougall (1994) proponen un paradigma global, basado en la idea de interacción que se da entre los tres grupos de participantes en el diseño, uso y evaluación del software educativo, a saber: los estudiantes, los profesores y los diseñadores. La propuesta parte de que el uso del software educativo esta relacionado esencialmente con tres cuestiones: ¿cómo puede mejorarse el aprendizaje mediante el uso del software?; ¿cómo el software educativo puede ser utilizado por los profesores para mejorar y ampliar su enseñanza?; y ¿cómo interactúan docentes y estudiantes en las aulas donde se utiliza el software?

El paradigma propuesto considera las interacciones entre las perspectivas de los participantes tomados de dos en dos: interacción entre las perspectivas del diseñador y del estudiante; interacción entre las perspectivas del diseñador y del profesor; interacción entre las perspectivas del profesor y del estudiante. La Figura No.2 muestra la estructura básica del marco de referencia.

La interacción de las perspectivas del profesor y del estudiante tiene sentido bidireccional dado que se establece entre dos personas, manifestándose en interacciones bidireccionales físicas y sociales muy directas entre los dos participantes, en la medida en que participan juntos en las actividades que se desarrollan en el aula, de manera que la conducta de uno influye y es influida por el otro.

Las interacciones entre las perspectivas del diseñador y del profesor y entre las del diseñador y del estudiante no son directas. En ambos casos, la interacción se establece entre la perspectiva de un actor activo (estudiante o profesor) y la de un actor pasivo, el diseñador. Una vez desarrollado el software, la participación directa del diseñador desaparece. Sin embargo, la perspectiva del diseñador se expresa en el programa, que incorpora los supuestos de partida y las intenciones del diseñador. Todo aquello que el diseñador hubiera querido aportar está implícito en el software.

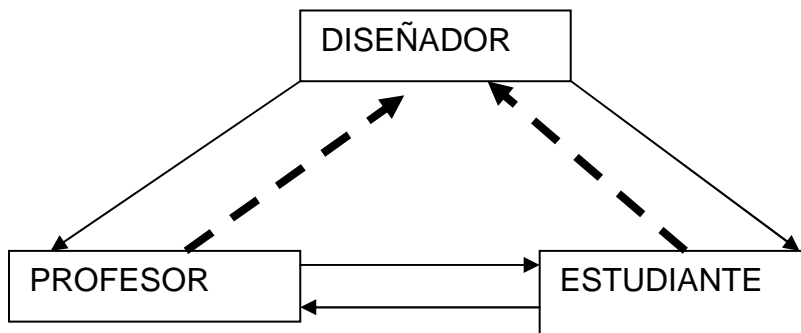


Figura No. 2: Paradigma de las interacciones entre las perspectivas (Squirres y McDougall, 1997).

La relación entre el diseñador y el estudiante se refiere al modelo mental (Schneiderman, B., 1998) que éste tenga sobre como se relacionan los alumnos con el software y como lo utilizan, generando cuestiones que apuntan al desarrollo cognitivo y a la interacción entre un humano y el computador. La relación entre el diseñador y el profesor se refiere al currículo y a las pedagogías asociadas con él, planteando cuestiones sobre el desarrollo curricular y los enfoques de enseñanza.

La relación entre el estudiante o el profesor y el diseñador, representada por líneas de guiones, pareciera ser redundante, si el software solo se utilizara tal como lo pretendía el diseñador. Sin embargo, con frecuencia los profesores y estudiantes utilizan el software de maneras no previstas por los diseñadores, lo que a veces conduce a usos innovadores del software.

El paradigma propuesto proporciona un modo de pensar sobre la selección del software educativo y sienta las bases para la evaluación del mismo, así como un medio general para reflexionar sobre la utilización del software educativo en las escuelas. En el contexto, la selección del software educativo debe comprenderse como la valoración de éste, efectuada por los profesores previendo su utilización con grupos de alumnos en las aulas o con estudiantes individuales, mientras que la evaluación se entiende como la valoración del mismo basada en la observación concreta de la utilización del mismo.

5. Enfoque basado en la ingeniería del software

El diseño y evaluación del software educativo regido por planteamientos propios de la ingeniería del software pone gran énfasis en la calidad del producto, la eficiencia y la organización del conocimiento (Galvis, 1995). Se busca obtener la calidad mediante la adopción de estándares internacionales².

En el desarrollo de software educativo bajo el modelo sistémico-ingenieril es más frecuente la evaluación formativa, o sea, se evalúa durante el proceso de diseño y desarrollo, con el fin de corregir y perfeccionar el programa. Los diseñadores de software mantienen como mecanismo

² ISO/IEC 9126 (1991). Information technology – Software Product Evaluation – Quality characteristics and guidelines for their use.

ISO/IEC 9001 (1991). Quality Systems –Model for quality assurance in design/development, production, installation an servicing.

necesario la evaluación del proceso de diseño y producción (Sánchez, 1995). Cuando los diseñadores de software son ingenieros de sistemas, lo que intentan evaluar son los aspectos computacionales, la relación hombre-máquina y a la facilidad o funcionalidad de uso del programa.

Existe gran variedad de modelos sistémicos, caracterizados por la descomposición del desarrollo del software educativo en tareas y procesos. Los más populares son el modelo de desarrollo lineal, también conocido como modelo en cascada, y el modelo de desarrollo por prototipos. El modelo lineal (Gros, 1997), considera la elaboración del software educativo como un proceso de cinco fases independientes: análisis, diseño, desarrollo, evaluación e implementación. La evaluación constituye una etapa más de todo el proceso. Mientras que el modelo de desarrollo rápido de prototipos “se caracteriza por un factor de revisión continua y actualización del producto” (Gros, 1997, p.23). Este modelo está basado en cinco fases: formulación de objetivos, diseño del programa, soluciones, prototipos, revisión de la solución y revisión de objetivos. Dentro del modelo de desarrollo rápido de prototipos, la tarea de desarrollo de software educativo se concibe como un proceso de resolución de problemas.

Sobrino (2000) propone un modelo sistémico para la evaluación de software educativo, esquematizado en la Figura No.3, que distingue tres fases: valoración del software, revisión del software y selección del software. Este modelo resulta más adecuado para evaluar software educativo que los modelos sistémicos, debido a que estos últimos son muy generales, utilizados para cualquier tipo de aplicación.

La primera fase, de valoración, requiere de procedimientos de observación y del establecimiento de una lista de criterios apropiados, escogidos por el evaluador. La observación es entendida en un sentido amplio, englobando posibles metodologías de encuesta, medida o, estrictamente de observación.

El proceso de observación es muy flexible, ya que considera cualquier instrumento que sea apropiado. Por ejemplo: informes de los resultados del aprendizaje, informes de las características de los alumnos, listas de control, diarios de clase, escalas de valoración del profesor, del alumno, o del experto (Marqués, 1991). También permite que la evaluación forme parte de un proceso de investigación por parte del profesor, para la mejora de su práctica, pudiendo tratarse de una investigación evaluativa, o una de corte cuasiexperimental, o de un modelo reflexivo como la investigación-acción (Rodríguez y otros, 1996; Colás y Buendía, 1994).

En cuanto a los criterios, o constructos teóricos, los propuestos por Marqués (1998) abarcan tres categorías. Los criterios técnicos, que hacen referencia a las cuestiones relacionadas con la calidad de los elementos audiovisuales, a la interacción y a la documentación. Los criterios didácticos, que son concernientes con aspectos instructivos del producto: sus objetivos, contenidos, actividades, evaluación y la metodología subyacente. Por último, los criterios de integración curricular, que evalúan los aprendizajes propiamente.

La segunda fase, la revisión del software educativo, se realiza una vez que se ha comparado lo observado contra los criterios. El profesor emite juicio acerca de la calidad del software

educativo, que se plasma en un informe con la reflexión crítica del docente acerca de las posibilidades del producto, en el ámbito de la adecuación técnica, instructiva y didáctica.

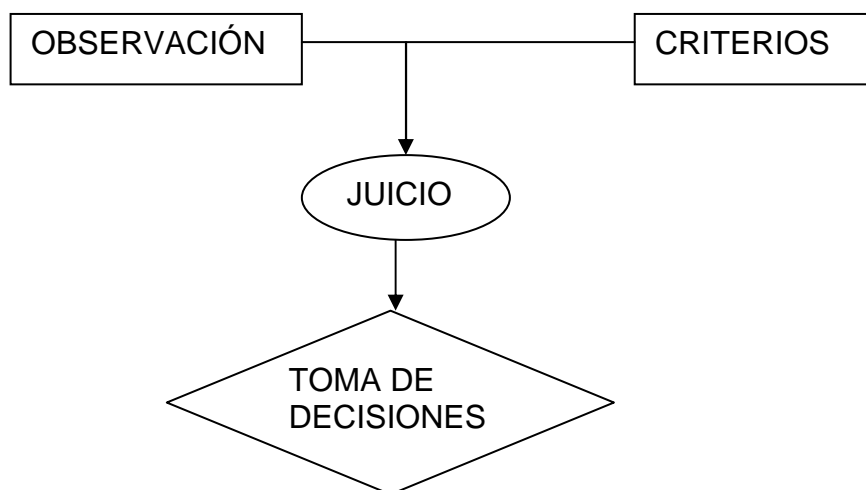


Figura No. 3: Fases para la evaluación de software educativo (Sobrino 2000).

En la última fase, el profesor selecciona el software educativo de mayor calidad, cuando se estén comparando varias alternativas. En el caso de evaluaciones por parte del alumno, el proceso se detendrá en la fase valorativa, y los datos provenientes de ella le facilitarán al profesor la redacción del informe. Cuando el evaluador sea un experto, la toma de decisiones consecuente al informe puede traducirse en la mejora de la construcción del propio software, en una versión posterior.

6. Enfoque cuantitativo

La propuesta para evaluar software educativo de Galvis (1994), analizada por López (1996), se fundamenta en un enfoque de evaluación sistémico-cuantitativo, que busca desarrollar un producto de software de alta calidad educativa y computacional, llevado a cabo mediante dos tipos de pruebas: evaluación de expertos y diseño de pruebas cuasiexperimentales. Ambas pruebas son utilizadas para la evaluación formativa y para la sumativa. También se incluye una prueba uno a uno y se recomienda una prueba diagnóstica inicial.

Con la evaluación se busca obtener información que permita, por un lado indagar acerca de la coherencia entre el diseño teórico y el desarrollo de Material Educativo Computarizado (MEC), y por otro lado, estimar el impacto que éste produce en el proceso de aprendizaje de una muestra representativa de aprendices. Se presupone que el desarrollo del MEC lo realizan un grupo

interdisciplinario, que incluye, para efectos de evaluación, a expertos en contenido, metodología e informática.

El enfoque sistémico-cuantitativo de evaluación busca la objetividad, para ello utiliza un conjunto de criterios preestablecidos, relevantes y consistentes, en contraste con las evaluaciones subjetivas que usan puntos de vista particulares, por lo tanto no siempre establecidos a priori (Rodríguez y otros, 1996).

Los criterios preestablecidos valoran tres componentes: calidad educativa, factores computacionales y la viabilidad de su uso. Para efectos de operacionalización, cada uno de estos componentes es caracterizado por una serie de variables, que constituyen el conjunto de atributos que lo afectan críticamente. Para cada variable se establecen indicadores que permiten observar el estado de ésta, así como niveles mínimos, aceptados como el estado adecuado de la variable (Colás y Buendía, 1994).

La prueba uno a uno (Galvis, 1994) consiste en una prueba operacional del MEC en que se busca determinar errores con relación a la conducta de entrada o con el análisis estructural de la instrucción. También busca detectar fallas con la comunicación textual, gráfica o sonora, así como la presentación misma del material. El evaluador trabaja con un alumno a la vez, quién interactúa con el material o partes coherentes y completas del mismo. De los comentarios y actuación del aprendiz al utilizar el MEC se determinan los problemas que el mismo pueda presentar.

Es necesario que los estudiantes que usan el material sean representativos de la población a la que se dirige el MEC. Al estudiante se le prepara para usar el MEC y se le deja interactuar libremente. Un observador acompaña al aprendiz durante su interacción, anota las dificultades que tiene, procura hallar las razones de éstas, observa la reacción del aprendiz a cada pantallazo, a cada situación. Luego se hacen los ajustes pertinentes y se ensaya la corrección con otro aprendiz. Cuando el MEC parezca estar bien, “habrá necesidad de pedir a algunos estudiantes que sigan secuencias de excepción, que traten de comportarse como alumnos *malos*, que traten de *romper* o de hacerle trampa al material, de encontrarle fallas” (Galvis, 1994, p.226). Se trata de asegurar que, además de que el software esté correcto desde la perspectiva teórica, también al ser usado por la población objeto no haya barreras de comunicación o de tratamiento didáctico que interfieran en el aprendizaje.

La prueba piloto y de campo con estudiantes empieza con un diseño para tratar de establecer la efectividad cognitiva y actitudinal que proporciona el MEC. Para ello se recomienda el diseño cuasiexperimental, en el que se manipulan algunas variables independientes con el propósito de determinar la influencia de estas sobre las variables dependientes. La aplicación de la concepción experimental demanda una gran rigurosidad en el control de las condiciones de la experiencia, para que sea posible lograr garantizar su validez interna y su validez externa, para así poder generalizar los resultados obtenidos (Colás y Buendía, 1994).

Se propone el establecimiento de un grupo de control, o sea un grupo que no recibe el tratamiento, para el que la variable independiente no será manipulada. La comparación del grupo tratado y del grupo no tratado permitirá apreciar el efecto del tratamiento sobre el nivel del logro de los estudiantes (la variable dependiente). Una prueba previa al tratamiento (*pre-test*) con una

prueba posterior al tratamiento (*post-test*) ayudará a establecer la efectividad que encierra el uso del MEC. El análisis de covarianza sirve de recurso para estimar la magnitud del impacto producido por el tratamiento (Hernández y otros, 1995).

7. Conclusiones

No existe una única teoría que fundamente los procesos de evaluación del software educativo. Por el contrario, hemos mostrado que se han elaborado una gama muy diversa de enfoques o marcos de referencia teórico-metodológicos. Sin embargo, consideramos que esto no debe percibirse como un obstáculo o algo indeseable, sino más bien, como una consecuencia natural de la rica diversidad de pensamientos y prácticas educativas que han propiciado toda una serie de pedagogías y opciones epistemológicas, las cuales necesariamente deben reflejarse en los marcos conceptuales para evaluar el software educativo.

De modo que la utilización del software en el ámbito educativo aún se encuentra en evolución. Los marcos teóricos de referencia y las terminologías necesarias para examinar y valorar el software educativo todavía no están suficientemente bien desarrolladas para establecer elementos unívocos de análisis. Se evidencia que la tarea de evaluar software educativo es bastante compleja y difícil.

Este trabajo logra distinguir aquellos aspectos más relevantes y característicos de cada enfoque, facilitando así su comparación y el relacionarlos con la epistemología y la pedagogía subyacente en ellos. Al destacar las características de los diversos paradigmas utilizados para evaluar el software educativo, necesariamente se concluye que algunos paradigmas son epistemológica y operacionalmente divergentes, mientras que otros tienen mayor convergencia en sus planteamientos. Así, el enfoque cuantitativo viene a ser una extensión o complemento del enfoque basado en la Ingeniería del software. Es decir se trata de enfoques teóricos convergentes. Mientras que el enfoque basado en la teoría crítica, es antagónico al enfoque cuantitativo, son enfoques altamente divergentes. Por otra parte, el enfoque basado en las interacciones es altamente convergente con el basado en la teoría crítica, viniendo a complementarlo.

Finalmente, consideramos importante señalar que algunos marcos teóricos tienden a darle mucho énfasis a los aspectos técnicos del diseño y evaluación del software educativo, pero hay que discernir que un diseño técnico sólido no garantiza por sí mismo el valor educativo de un software. No se trata tanto de medir el software educativo como producto, sino, sobre todo, los procesos de enseñanza-aprendizaje que facilita o potencia.

8. Bibliografía

- Colás, M. y Buendía L.(1994). Investigación educativa. Ediciones Alfar. Sevilla, España.
- Duart, J. y Sangrá, A. (2000). Aprender en la virtualidad. Gedisa. Barcelona, España.
- Fonseca, C. (1991). Computadoras en la escuela pública costarricense: la puesta en marcha de una decisión. Primera edición. Fundación Omar Dengo. San José, Costa Rica.

- Galvis, A.(1995). Fundamentos de tecnología educativa. Editorial de la Universidad Estatal a Distancia. San José, Costa Rica.
- _____ (1994). Ingeniería de software educativo. Ediciones UNIANDES, Universidad de los Andes, Santafé de Bogotá, Colombia.
- Gros, B. (1997). Diseños y Programas Educativos. Editorial Ariel S.A. Barcelona, España.
- Gutiérrez, M. y Levis, D. (2000). ¿Hacia la herramienta educativa universal?. CICCUS. Ríobamba, Argentina.
- Harasim, L., y otros. (2000). Redes de aprendizaje. Gedisa. Barcelona, España.
- Heller, R.(1991). Evaluating Software: A Review of the options”, Computers and Education, Vol. 17, No.4, pp. 285-291.
- Hernández, R. y otros. (1995). Metodología de la investigación. McGraw-Hill. México.
- Insa, D. y Morata, R. (1998). Multimedia e Internet. Editorial Paraninfo. Madrid, España.
- Kemmis, S. (1976). The Educational Potential of Computer Assisted learning: Qualitative Evidence about Student Learning. University of East Anglia. Gran Bretaña.
- Kemmis, S. y otros (1977). How Do Students Learn?, Working Papers on CAL, Doc.No.5, Centre for Applied Research in Education, University of East Anglia. Gran Bretaña.
- López, E. (1996). Guía de estudio para el curso evaluación de hardware y software educativo. Oficina de publicaciones de la UNED. Universidad Estatal a Distancia. San José, Costa Rica.
- Marqués, P. (1991). Ficha de evaluación y clasificación de software educativo. Novática, Vol. XVII, No. 90. pp. 29-32.
- _____. (1998). Programas didácticos: diseño y evaluación. [Disponible en <http://www.sauce.pntic.mec.es/~global.html>]. Última fecha: junio del 2003.
- Papert, S. (1980). Mindstorms: Children, Computers, and Powerfull Ideas. New York: Basic Books, Inc.
- Papert, S. (1987). Desafío a la mente. Editorial Galapos, Buenos Aires.
- Rodriguez, G. y otros. (1996). Metodología de la investigación cualitativa. Ediciones Aljibe. Malaga, España.
- Sánchez, J. (1995). Informática Educativa. Editorial Universitaria. Santiago, Chile.
- Santiago, R. (1998) Evaluación de software educativo multimedia: Proyecto PEMGU. [Disponible en <http://www.irabia.org/pmgu/documento.html>]
- Schneiderman, B. (1998). Designing the user interface: Strategies for effective human-computer interaction. Addison-Wesley Editors. USA.
- Sobrino, A. (2000). Evaluación del software educativo. En Reparaz, Ch., Sobrino, A. y Santiago, R. Integración curricular de las nuevas tecnologías. Ariel Practicum. Barcelona, España.
- Squirres, D. y McDougall, A. (1997). Cómo elegir y utilizar software educativo. Ediciones Morata. Madrid, España.
- Vargas, C. (2003). Software educativo: aporte de las universidades estatales costarricenses durante el periodo 1900-2001. Revista de Ingeniería. Universidad de Costa Rica. Vol. 12, No.2, pp.100-111.
- Vivet, M. (1996). Evaluating Educational Technologies. Proceedings de CALISCE. San Sebastian. España. 9-31 de Julio de 1996.
- Zaragoza, J. y Cassado, A. (1992). Aspectos técnicos y pedagógicos del ordenador en la escuela. Editorial Bruno. Madrid, España.