

VALIDACIÓN DE SOFTWARE EDUCATIVO

Lic. Juan José Fallas Monge¹ - Lic. Jeffrey Chavarría Molina²
jfallas@itcr.ac.cr jchavarría@itcr.ac.cr

Resumen

Un software educativo es cualquier programa o entorno computacional creado con la finalidad de contribuir o apoyar el proceso de enseñanza-aprendizaje. En el diseño de un software educativo debe contemplarse, como una de sus etapas, un proceso de validación, el cual permita determinar su pertinencia y calidad. En este artículo se detalla sobre los criterios que comúnmente se toman en cuenta para la validación de un software educativo. Además, se abordarán algunas interrogantes que giran alrededor de este tópico: ¿qué se entiende por software educativo? ¿En qué consiste la validación de un software? ¿Qué instrumentos se emplean para dicho proceso? ¿Por qué es importante validar un software educativo?

Palabras claves: Software educativo, Validación de software.

Abstract

Educational software is any computer program or environment created to contribute to or support a teaching-learning process. A validation process is an essential part of the design of educational software to establish the quality and the pertinence of the software. In this article the criteria commonly used to validate educational software are shown. Some of the following questions will guide the discussion about the topic: What is educational software? What is needed to validate educational software? What instruments are used in these processes of validation? Why is it important to validate educational software?

Keywords: Educational software, Software validation.

1. Concepto de software educativo

Con fundamento en González (2002), puede decirse que un software educativo es un programa cuya finalidad específica se orienta a su uso como medio didáctico o de facilitación de los procesos de enseñanza-aprendizaje.

De manera concordante, Marquès (1996) indica que las expresiones software educativo, programas educativos y programas didácticos, se pueden utilizar como sinónimos para designar genéricamente los programas para ordenador creados con la finalidad específica de ser utilizados como medio didáctico, es decir, para facilitar los procesos de enseñanza y de aprendizaje.

Adicionalmente, es preciso aclarar, como lo hace Marquès (1996), que la definición anterior:

¹ Profesor de la Escuela de Matemática del Instituto Tecnológico de Costa Rica.

² Profesor de la Escuela de Matemática del Instituto Tecnológico de Costa Rica.

se encuentra más basada en un criterio de , nalidad que de funcionalidad, por lo tanto se excluyen del software educativo todos los programas de uso general en el mundo empresarial que también se utilizan en los centros educativos con funciones didácticas o instrumentales, como por ejemplo: procesadores de textos, gestores de bases de datos, hojas de cálculo, editores grá, cos. Estos programas, aunque puedan desarrollar una función didáctica, no han estado elaborados especí, camente con esta , nalidad.

En general, existe gran variedad de de, niciones para el término **software educativo**. Sin embargo, la mayoría de estas de, niciones presentan aspectos comunes que deben caracterizar a un software para ser considerado educativo: **nalidad didáctica**, **atencionalidad pedagógica**, **apoyo curricular**, **material pedagógico**, **medio didáctico**. En el caso particular de este artículo, entenderemos como software educativo todo programa o entorno computacional creado con la , nalidad de contribuir o apoyar el proceso de enseñanza-aprendizaje de algún contenido especí, co de una determinada asignatura, con las limitantes que anotó Marquès (1996) y que indicamos anteriormente.

2. ¿En qué consiste la validación de un software educativo?

A pesar de que varias de las de, niciones que se pueden encontrar para el término **software educativo** hacen referencia implícita o explícitamente a la **atencionalidad pedagógica**, creemos que la **atención**, aunque es importante, no es suficiente. El diseño de un software debe contemplar, como una de sus etapas, un proceso de validación que permita determinar su pertinencia y calidad.

Cataldi (2000), de, ne la evaluación (o validación) de programas educativos como:

un proceso que consiste en la determinación del grado de adecuación de dichos programas al contexto educativo. Cuando el programa llega al docente, es de suponer que ha sido analizado y evaluado tanto en sus aspectos pedagógicos y didácticos, como en los técnicos . . . (p. 37).

Según esta autora, para realizar la validación de un software educativo se debe concretar una evaluación interna y una evaluación externa del software. La evaluación interna estará a cargo del equipo que desarrolla el software y se relaciona con los aspectos técnicos de este. Es decir, consiste en constatar que el programa funcione bien; es decir, hace lo que se supone que debe hacer y lo hace bien. Como lo indica Marquès (1996), consiste en un análisis estructural de los elementos que componen el programa.

Por otra parte, la evaluación externa permite obtener las sugerencias de los alumnos potenciales, quienes serán en de, nitiva los usuarios del software, y de los docentes, que lo utilizarán como material didáctico. En esta etapa se encuentran a menudo errores imprevistos que no han sido detectados anteriormente (Cataldi, 2000).

3. ¿Qué instrumentos se utilizan en la validación de un software educativo?

En procesos de evaluación externa es común que se utilicen instrumentos de validación diseñados para tal , n. Según Díaz-Antón, Pérez, Grimán y Mendoza (2002), un instrumento de validación es un modelo que permite medir la calidad de un

software educativo. Normalmente, los instrumentos de validación de esta índole corresponden a listas de cotejo en las cuales se verifica la presencia o no de ciertos elementos dentro del software (Hernández, 2007).

Los instrumentos habitualmente empleados para la evaluación de software educativo son fundamentalmente de tipo técnico (requerimientos del software y público meta), de tipo pedagógico (objetivos, contenidos y actividades que se abordan) o de aspectos funcionales (venta y respecto a otros medios, cómo, para qué y cuándo utilizarlos) (Marquès, 1998).

Según Cataldi (2000), en general, los instrumentos más usados para la evaluación de programas educativos son los cuestionarios de valoración, donde las respuestas son valoradas entre 0 y 5, por ejemplo, de manera que el resultado es el grado de conformidad del usuario con las afirmaciones propuestas.

	Malo	Regular	Bueno	Muy bueno	Excelente
¿Considera adecuado:					
el diseño de la pantalla?	<input type="checkbox"/>				
el diseño de las ventanas?	<input type="checkbox"/>				
el diseño de los botones?	<input type="checkbox"/>				
el uso de los colores?	<input type="checkbox"/>				
el uso de los tipos de letra?	<input type="checkbox"/>				

Figura 1: Extracto de un cuestionario de valoración.

4. ¿Para qué se valida un software educativo?

Existen varias razones que justifican la importancia de la validación de un software educativo. Estas razones dependen del interés del individuo o empresa que desarrolla el proceso de validación. En general, un proceso de esta índole se desarrolla para determinar, principalmente:

- La pertinencia del software como herramienta pedagógica para la enseñanza. De nada sirve un software educativo con un excelente diseño técnico, si no sirve para lo que fue creado: para enseñar.
- La pertinencia de los contenidos teóricos que se exponen en el programa. Si el software presenta errores conceptuales, en lugar de favorecer el proceso de enseñanza-aprendizaje del educando, lo va a obstaculizar.
- Las mejoras técnicas y funcionales que deben hacerse al programa. Un software debe estar depurado para que esté libre de errores y preparado para responder ante el ingreso incorrecto de datos por parte del usuario.

De este modo, un único proceso de validación podría resultar insuficiente para concluir que un software educativo está debidamente validado. Las validaciones suelen ser procesos largos e inclusive costosos, pero se justifican por la finalidad específica para la que son diseñados estos programas. De cierta manera, un software juega el papel de docente o instructor, entonces un programa que contiene errores podría ser inadecuado para la enseñanza, como lo es un profesor que no conoce bien la asignatura que imparte.

5. Algunas consideraciones sobre la validación

En el caso de los programas especializados, los periodos de prueba pueden extenderse por varios meses, incluso años. Expertos y usuarios de muchas partes del mundo ponen a prueba estos programas (llamados versiones Beta) y envían los comentarios y sugerencias a los diseñadores del software. La idea de las empresas es que cuando el programa salga a la venta carezca completamente de errores.

Con el software gratuito el problema es más serio. El hecho de que los usuarios no tengan que cancelar una licencia, da cabida a que muchos de estos programas no estén validados. No obstante, debe señalarse que sí existe software gratuito de excelente calidad.

En el caso del software gratuito con fines educativos, los errores de programación pueden tener consecuencias negativas en el proceso de enseñanza-aprendizaje de los estudiantes. Consideremos, a manera de ejemplo, la siguiente situación: se desea utilizar un software para reforzar el proceso de enseñanza-aprendizaje del tema de operaciones con números enteros; y ocurre que el software seleccionado para tal fin no distingue la prioridad entre las operaciones adición, sustracción, multiplicación y división. Sin duda alguna, este error puede generar confusiones en los estudiantes.

Por lo tanto, es importante que todo software educativo cuente con al menos una etapa de validación, mediante la cual el software adquiera una mejor calidad técnica. Sin embargo, es recomendable que si el programador o programadores deciden no realizar esta etapa, incorporen en la documentación del programa alguna advertencia que indique que éste no ha sido validado.

No siempre se tiene garantía sobre la calidad del software educativo que se tiene a disposición, por lo que es importante que el docente analice cualquier programa antes de emplearlo en su clase, ya que él es el principal responsable de la calidad y pertinencia del software educativo que se utiliza en el aula. Un error de funcionalidad de un programa detectado a tiempo, permite que un software sea descartado oportunamente por parte del docente.

El educador debe, también, decidir si el programa responde o no a los objetivos educativos del tema o temas que va a abordar con los estudiantes y, además, debe revisar que éste no contenga errores. Si se trata de un programa especializado como Derive o el Geómetra (por mencionar algunos ejemplos) no tiene sentido hacer un análisis en búsqueda de errores, pues estos programas han sido debidamente validados por las empresas diseñadoras; sin embargo, sí es importante que el docente reexamine sobre la pertinencia o no de incorporarlos a su clase. Un software educativo puede tener un excelente diseño y ser de muy buena calidad, pero no adaptarse a las necesidades específicas del docente.

Finalmente, en el caso del software gratuito existen muchos de muy buena calidad y, eventualmente, podrían responder mejor a las necesidades del docente. Sin embargo, esto no sucede en todos los casos y, por lo tanto, el docente debe ser muy cauteloso cuando desea utilizar programas de este tipo.

6. ¿Qué se evalúa a un software educativo?

Después de estudiar el planteamiento de varios autores (Cataldi (2000), Marquès (1998), González (2002)), llegamos a la conclusión de que existen ciertos estándares

mínimos que deben tomarse en cuenta a la hora de evaluar un software educativo. Estos estándares apuntan principalmente a dos aspectos: los aspectos pedagógicos y didácticos del programa, y los aspectos técnicos (funcionalidad del programa).

Al respecto, Marquès (1998) señala, entre otras, las siguientes características que deben evaluarse en un software educativo:

▪ **Facilidad de uso e instalación**

Los programas educativos, dentro de lo posible, deben ser agradables y fáciles de usar. Un usuario con conocimientos básicos de computación debe ser capaz de utilizarlos sin necesidad de recurrir inmediatamente a la ayuda o tutorial.

La instalación del programa en la computadora debe ser ágil y no necesitar la intervención excesiva del usuario en el proceso de instalación. Por último, debe existir un desinstalador.

▪ **Versatilidad**

La versatilidad de un programa se refiere a la capacidad de adaptación a diversos contextos educativos. Por ejemplo, un software educativo se debe poder utilizar en distintos entornos como en el laboratorio o en el hogar, y empleando diferentes estrategias metodológicas como el trabajo individual o el trabajo grupal.

▪ **Calidad del entorno audiovisual**

La interfaz de un software debe ser atractiva y técnicamente bien diseñada. Algunos de los aspectos que en este sentido deben cuidarse son:

- Diseño general claro y atractivo de las pantallas, sin exceso de texto.
- Calidad técnica y estética en: títulos, menús, ventanas, íconos, botones, barras de estado, fondo, lenguaje, letra, color, etc.

▪ **Contenidos**

En el diseño de un software educativo debe cuidarse la calidad de los contenidos y la forma en que son presentados al usuario. La información debe ser veraz y significativa. No debe haber faltas de ortografía, ni mensajes que falten a la integridad del usuario³.

▪ **Adecuación a los usuarios y a su ritmo de trabajo**

Un buen programa educativo debe adaptarse a usuarios que presenten diferentes niveles de conocimiento, tanto sobre el programa como sobre la materia. Esto significa que pueda ser utilizado tanto por principiantes, como por usuarios que ya conocen sobre el tema. Además, debe considerar las características individuales, así como los progresos que vaya teniendo el usuario.

³ Comúnmente los programas, dependiendo de las acciones del usuario, despliegan mensajes informativos o indicadores de error. Estos mensajes deben ser claros y no utilizar frases o términos que puedan ofender al usuario. Por ejemplo, la frase: *%a se ha equivocado varias veces, sin embargo, intente de nuevo+*, es innecesaria y podría insinuar al usuario que es una persona que se equivoca mucho. En su lugar basta con *%Intente de nuevo+*.

▪ **Enfoque pedagógico actual**

Hasta donde sea posible, un software educativo debe evitar la memorización, por lo que debe respetar un enfoque educativo centrado en el estudiante, que permita que sea partícipe en la construcción de su propio conocimiento.

▪ **La documentación**

Un software educativo debe contar con información que detalle las posibilidades de uso del programa. Esta información debe presentarse de una forma clara, agradable, sencilla y técnicamente bien redactada.

Con un planteamiento muy similar, González (2002) argumenta que para catalogar un software como bueno desde el punto de vista educativo, éste debe responder a diversos aspectos técnicos, pedagógicos, metodológicos y funcionales. Entre estos cita los siguientes: de fácil instalación y uso, calidad del entorno visual, versatilidad y adecuada documentación.

También, González (2002) plantea que la navegación (movilidad dentro del programa) deberá ser sencilla para así favorecer su facilidad de uso. El usuario debe tener el control en todo momento de la pantalla en donde se encuentra y saber cómo desplazarse con facilidad. Además, el software educativo debe fomentar el autoaprendizaje, permitiendo al usuario decidir sobre las tareas que va a desarrollar, el tiempo, el nivel de profundidad y la secuencia en que las va desarrollando (González, 2002).

Por otra parte, en el año 2006, el Departamento Nacional de Software Educativo del MINED (Ministerio de Educación Pública de la República de Cuba), propuso un instrumento para la validación de software educativo. En este material se citan los siguientes criterios (allí se denominan *indicadores*):

- Correspondencia del software con los objetivos y contenidos de la asignatura que abarca, según el grado.
- Fiabilidad conceptual (grado de veracidad, actualidad y asequibilidad de los contenidos con respecto al nivel escolar al que va dirigido).
- Excelente ortografía, lenguaje acorde con el usuario, legibilidad (letra: tamaño y color), posibilidades que brinda para la atención de las características individuales, variedad de ejercicios, facilidad de navegación e interacción y efectividad de la ayuda.

Siguiendo en esta misma línea, la Revista Mexicana de Educación y Cultura: *La Tarea*, en el año 2000, plantea como parte de las pautas a seguir en la evaluación de un software educativo, las siguientes:

- El software debe estar bien documentado.
- Su contenido debe ser revisado de manera que se pueda evaluar la calidad de la información, así como la forma en que ésta es presentada.
- El software debe tener un valor educacional y debe ser evaluado en función del impacto positivo que pueda tener en el alcance de los objetivos curriculares.

- El programa debe estar libre de errores (análisis de los aspectos funcionales del programa).
- Debe estar preparado para responder al ser utilizado en condiciones extremas (ingreso incorrecto de datos por parte del usuario). En general, el programa debe funcionar bien, técnicamente hablando.

En síntesis, podemos concluir que las cualidades que debe cumplir un software educativo son:

- Facilidad de uso y facilidad de instalación.
- Buena funcionalidad.
- Debe tener una , nalidad pedagógica y su uso debe potenciar el proceso de enseñanza-aprendizaje de algún contenido.
- Sus contenidos deben ser de calidad y no contener errores de ninguna índole.
- Su interfaz (pantallas, mensajes, menús,õ) debe ser agradable al usuario.
- Debe estar bien documentado. Debe contener como mínimo un archivo de ayuda al que se pueda recurrir en cualquier momento.

No obstante, como lo plantean Cataldi, Lage, Pessacq y García (1999):

Las largas listas de criterios a desarrollar para evaluar los programas educativos, son datos relativos a la hora de hacer uso del recurso computacional. El rol docente condiciona el uso de los programas, siendo la creatividad y la originalidad de las propuestas las que permiten incrementar el valor de los medios y no el medio mismo (p. 13).

En efecto, a pesar de todas las características planteadas anteriormente que debe tener un software educativo, el rol del docente es crucial para lograr un aprovechamiento adecuado de este recurso. Las fortalezas y debilidades que tenga un software no solo dependen de sus características intrínsecas, sino también de la forma en que el docente haga uso de él.

7. Conclusiones

1. Un proceso de validación permite determinar la pertinencia y la calidad de un software educativo.
2. La validación debe ser visto como un proceso inherente y paralelo al diseño de un software educativo. No basta con un diseño bien intencionado, sino que se debe garantizar, dentro de lo posible, la calidad del producto que se está ofreciendo a la docencia y al estudiantado.
3. La validación de un software educativo busca determinar, principalmente, la pertinencia del software como herramienta pedagógica para la enseñanza, la pertinencia de los contenidos teóricos que se exponen en el programa y las mejoras técnicas y funcionales que deben hacerse al programa. Por ende, dependiendo de la extensión del software, podrían ser necesarias varias etapas de validación que incorporen los puntos citados anteriormente.

4. Los instrumentos más comunes que se emplean para validar programas educativos son las listas de cotejo o cuestionarios de valoración, los cuales se orientan para validar, mayoritariamente, los aspectos técnicos y funcionales del programa, así como los aspectos pedagógicos.
5. No siempre se tiene garantía sobre la calidad del software educativo que se tiene a disposición, principalmente en el caso del software gratuito, por lo que es importante que el docente siempre analice cualquier programa antes de emplearlo en su clase.

8. Referencias

Cataldi, Z. (2000) *Metodología de diseño, desarrollo y evaluación de software educativo*. Tesis de Maestría, Facultad de Informática, Universidad Nacional de la Plata, Argentina.

Cataldi, Z., Lage, F., Pessacq, R. y García, R. (1999) *Ingeniería de software educativo*. Recuperado el 8 de octubre de 2007, desde: www.uba.ar/laboratorios/lsi/c-icie99-ingenieriasoftwareeducativo.pdf

Chavarría, J. y Fallas, J. (2008) *Validación del software educativo PoliEstudio 1.0*. Tesis de Licenciatura, Escuela de Matemáticas, Universidad Nacional, Costa Rica.

Departamento Nacional de Software Educativo (2006) Validación pedagógica de un software educativo. Recuperado el 8 de octubre de 2007, desde: <http://www.insted.rimed.cu/>

Díaz-Antón, M., Pérez, M., Grimán, A. y Mendoza, L. (2002) *Instrumento de evaluación de software educativo bajo un enfoque sistémico*. Recuperado el 26 de abril del 2008, desde: www.academia-interactiva.com/evaluacion.pdf

González, B. (2002) *Criterios y métodos de evaluación de software educativo*. Recuperado el 29 de setiembre del 2007, desde: <http://byrong.iespana.es/public/evsoftwared.pdf>

Hernández, V. (2007) *Un modelo de evaluación de software educativo para la Enseñanza de la Matemática*. Recuperado el 27 de abril del 2008, desde: www.utn.edu.ar/aprobedutec07/docs/58.pdf

Marquès, P. (1996) *El software educativo*. Universidad Autónoma de Barcelona. Recuperado el 5 de setiembre del 2007, desde: [http://www.lmi.ub.es/te/any96/marques software/#index](http://www.lmi.ub.es/te/any96/marques%20software/#index)

Marquès, P. (1998) *Programas didácticos: diseño y evaluación*. Recuperado el 15 de setiembre de 2007, desde: <http://www.karisma.org.co>

Sin autor (2000) *Aprender a elegir correctamente*. Revista electrónica La Tarea: Nuevas tecnologías en Educación. No 12, febrero 2000. Disponible en <http://www.latarea.com.mx/>