

Instrumento de evaluación de software educativo bajo un enfoque sistémico

Díaz-Antón, G., Pérez, M., Grimán, A., Mendoza, L.

Abstract—

La incorporación de las Tecnologías de la Información y la Comunicación (TIC) a la educación ofrece distintas dimensiones al proceso instruccional. En particular, el uso del software educativo en el proceso de enseñanza-aprendizaje permite mejorar en el estudiante las destrezas cognitivas. Este tipo de software fomenta el análisis de problemas, facilita el trabajo en grupo, provee soporte en actividades docentes; en el sentido más amplio, mejora las habilidades del pensamiento y la resolución de problemas. Ahora bien, para lograr todo esto, el software debe ser de calidad.

Cuando se hace referencia a calidad de software educativo, se requiere un producto que satisfaga tanto las expectativas de los docentes como de los alumnos, a un menor costo, libre de errores y cumpliendo con ciertas especificaciones instruccionales y tecnológicas. Esta necesidad conlleva a generar un modelo para medir la calidad del software como producto y como servicio. El objetivo de esta investigación es presentar el desarrollo de un modelo de calidad del software educativo con sus respectivas métricas; siguiendo un enfoque sistémico. Para ello se partió del Modelo Sistémico de Calidad de Software (MOSCA) elaborado por LISI-USB, ampliándolo de acuerdo a los requerimientos particulares de calidad del software educativo, tomando en cuenta no sólo los aspectos técnicos del producto, sino el diseño pedagógico y los materiales de soporte didáctico. Finalmente, a través de un caso de estudio real, se presenta la aplicación del modelo ampliado MOSCA, se analiza su desempeño, y se ofrece un conjunto de métricas que permiten evaluaciones posteriores. El resultado principal es un modelo en versión prototipo que permite medir la calidad para software educativo.

Index Terms— Evaluación de software educativo, calidad de software educativo, medición de calidad.

Este trabajo es parte del proyecto especial de postgrado de Gabriela Díaz-Antón, en Informática Educativa, de la Universidad Simón Bolívar (USB) y de la línea de investigación del grupo del Laboratorio de Información en Sistemas de Información (LISI), Departamento de Procesos y Sistemas de la USB, Caracas, Venezuela.

Pérez, María Angélica, PhD, es profesora y coinvestigadora del Laboratorio de Información en Sistemas de Información (LISI), Universidad Simón Bolívar (USB), Departamento de Procesos y Sistemas, Apartado postal 89000, Caracas 1080-A, Venezuela. (e-mail: mavalles@usb.ve).

Mendoza, Luis Eduardo, MSc, es profesor de sistemas de información en la Universidad Simón Bolívar (USB), Departamento de Procesos y Sistemas, Caracas, Venezuela. (e-mail: lmendoza@usb.ve).

Grimán, Ana Cecilia, MSc, es profesora de sistemas de información en la Universidad Simón Bolívar (USB), Departamento de Procesos y Sistemas, Caracas, Venezuela. (e-mail: agriman@usb.ve).

Díaz-Antón, Gabriela, es Especialista en Informática Educativa, directora y consultora académica en *Academia Educación Interactiva, C.A.*, Caracas, Venezuela, (autor a quien la correspondencia debe ser enviada: e-mail: gabriela_diaz@cantv.net).

I. INTRODUCCIÓN

Las ventajas del uso de software educativo como instrumento de ayuda instruccional en el proceso de enseñanza-aprendizaje, está ampliamente documentada (Beichner, 1994; Borrás y Lafayette, 1994; Cohen et al., 1994; Cohen, Tsai et al., 1995; Díaz-Antón, 2002; Jonassen y Reeves, 1996, Liu y Reed 1995; Mayer et al., 1997; Yildirim et al., 2001). Sin embargo, el uso adecuado de esta herramienta informática requiere de una cuidadosa selección para garantizar un producto de calidad.

El tema de evaluación del software educativo ha sido estudiado y documentado por diversos autores del ámbito educativo, proporcionando medidas de evaluación en el área educativa y técnica (Barrosot et al., 1998; Del Moral, 1998, Galvis, 2000; Gómez, 1997; González, 1999; Gros (Coord.) et al., 1997; Marquès, 1998; Martínez, 1993; MVU, 2002; Navarro, 1999; OTA, 1988; PEMGU, 1999; Reeves, 1998; Stephen, 1998). Destacan los métodos de evaluación de Galvis (2000) y de la Universidad Virtual de Michigan (2002), que utilizan métodos cuantitativos de evaluación. Sin embargo, en vista de que gran parte de las propuestas sobre software educativo, son de índole cualitativa o necesitan adaptarse a medidas estándares de evaluación de software según las normas ISO/IEC 9126 (1991), surge la necesidad de la disponibilidad de un instrumento de medidas estándares de calidad para la evaluación de software educativo, que sea de utilidad tanto para los desarrolladores de software educativo como para los interesados en adquirir software comercial (por ejemplo, educadores e instituciones educativas).

Se propone entonces un modelo de evaluación de software educativo bajo un enfoque sistémico de calidad, basado en *El Modelo Sistémico de Calidad de Software* (MOSCA) de Mendoza et al., (2001), elaborado por LISI-USB (Laboratorio de Información y Sistemas de Información, Universidad Simón Bolívar (USB), soportado por los conceptos de calidad total sistémica (Callaos y Callaos, 1993; Pérez et al., 1999).

Esta propuesta consiste en una serie de cuestionarios a través de los cuales se realiza la medición por docentes, especialistas de informática y alumnos. La propuesta de un modelo de evaluación de software educativo bajo un enfoque sistémico de calidad, ofrece una metodología de preselección y estudio de selección final para la adquisición del software educativo así como los estudios de campo para la validación del software o cuestionarios para el alumno en formatos estandarizados, dependiendo de si el software a evaluar se desea adquirir comercialmente como producto

final (institutos educativos, educadores, padres) o está en proceso de desarrollo (producción de software educativo).

Lo novedoso de este instrumento es que cuantifica las métricas de evaluación de calidad a partir de tres categorías, Funcionalidad, Usabilidad y Fiabilidad, con sus respectivas métricas, determinando si el software educativo es de calidad básica, intermedia o avanzada. Los resultados de la evaluación se traducen además en unas guías, en donde se consignan los juicios evaluativos, posibilidades de integración del software educativo con sentido pedagógico en un currículo o proyecto pedagógico real. Se toman en cuenta, tanto los aspectos del contenido (culturales, ideológicos y valorativos), como los aspectos informáticos o técnicos; así como también los documentos de soporte pedagógicos y técnicos.

Para la elaboración de la propuesta del modelo, se analiza en primer lugar *El Modelo Sistemico de Calidad de Software* (MOSCA). En segundo lugar, se selecciona un conjunto del total de las características, categorías y métricas de MOSCA, que se ajusten a la evaluación de software educativo, formando así la base de la propuesta. Por último se le añaden los parámetros específicos relacionados con la calidad educativa del software, dando lugar a la adición de un nuevo conjunto de medidas que involucran tanto un nuevo parámetro de medición (subcaracterísticas), como nuevas métricas que no están presentes en MOSCA. De esta manera se realiza una ampliación de MOSCA para el software educativo.

Además, se pauta la prueba del modelo de evaluación de software educativo con enfoque sistémico de calidad, a través de un estudio de campo. Finalmente, se enumerarán las conclusiones y recomendaciones que se han derivado de esta investigación en progreso.

II. CALIDAD DE SOFTWARE

Según Pressman (2002), la calidad del software es la concordancia con los requerimientos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente. La ausencia de defectos, la aptitud para el uso, la seguridad, la confiabilidad y la reunión de especificaciones son elementos que están involucrados en el concepto de calidad del software.

A la hora de definir la calidad del software se debe diferenciar entre la calidad del producto software y la calidad del proceso de desarrollo de éste -calidad de diseño y fabricación- (Callaos y Callaos, 1993; Pérez et al., 1999). No obstante, las metas que se establezcan para la calidad del producto van a determinar los objetivos del proceso de desarrollo, ya que la calidad del primero va a depender, entre otros aspectos, de éstos últimos. Según Callaos y Callaos (1993), la calidad de los Sistemas de Software no es algo que depende de una sola característica en particular, sino que obedece al compromiso de todas sus partes.

Tomando en cuenta la calidad del producto y la calidad del proceso, se desarrolló *El Modelo Sistemico de Calidad de software* (MOSCA), por LISI-USB (Mendoza et al., 2001), que integra el modelo de calidad del producto (Ortega, et al., 2000) y el modelo de calidad del proceso de

desarrollo (Pérez et al., 2001), y está soportado por los conceptos de calidad total sistémica (Callaos y Callaos, 1993; Pérez et al., 1999).

En cuanto al producto, este modelo plantea, sobre la base de las 6 características de calidad del estándar internacional ISO/IEC 9126 (1991), un conjunto de categorías, características y métricas asociadas que miden la calidad de un producto de software con un enfoque sistémico y hacen del modelo un instrumento de medición de gran valor ya que cubre todos los aspectos imprescindibles para medir directamente la calidad del producto de software. En cuanto al proceso, este modelo plantea sobre la base de las 5 características de calidad del estándar internacional ISO/IEC 15504 (1991), un conjunto de categorías, características y métricas asociadas que miden la calidad de un proceso de software con un enfoque sistémico.

MOSCA consta de cuatro niveles: dimensiones, categorías, características y métricas; con un total de 587 métricas. Además proporciona el algoritmo para evaluar la calidad sistémica. El algoritmo es un conjunto de pasos procedimentales que se realizan para ejecutar el modelo y estimar la calidad de software. El algoritmo contempla tres (3) fases:

1. Fase 1: Calidad del producto de software con un enfoque sistémico.
2. Fase 2: Calidad del proceso de desarrollo de software con un enfoque sistémico.
3. Fase 3: Integración de las mediciones de los submodelos de la calidad del producto y la calidad del proceso.

Cabe acotar que, según el método propuesto para aplicar el modelo de calidad del producto de software (Mendoza, et al., 2001), no todas las características y métricas tienen que ser usadas, sino que el modelo debe ser adaptado dependiendo de las características que se desean del producto, en virtud del dominio del software.

III. CALIDAD DE SOFTWARE EDUCATIVO

En primera instancia se define el término de software educativo. Según Galvis (2000), en el campo educativo suele denominarse software educativo a aquellos programas que permiten cumplir y apoyar funciones educativas. En esta categoría entran tanto los que dan soporte al proceso de enseñanza y aprendizaje (un sistema para enseñar matemáticas, ortografía, contenidos o ciertas habilidades cognitivas), como los que apoyan la administración de procesos educacionales o de investigación (ej., un sistema que permita manejar un banco de preguntas). El significado que se maneja en este trabajo está relacionado principalmente con la primera definición, es decir, con los materiales educativos computarizados que apoyan el proceso de enseñanza-aprendizaje, a las que en Inglés se denomina *courseware* (i.e., software educativo para cursos).

Cuando se hace referencia a calidad de software educativo, se requiere de un producto que satisfaga tanto las expectativas de los docentes como de los usuarios, a un menor costo, libre de defectos y cumpliendo con ciertas

especificaciones instruccionales y tecnológicas. Según Gros (2000), la calidad del software está determinada no sólo por los aspectos técnicos del producto sino por el diseño pedagógico y los materiales de soporte. Éste último aspecto es uno de los más problemáticos ya que existen pocos programas que ofrezcan un soporte didáctico. La evaluación de software educativo se ha centrado tradicionalmente en dos momentos:

1. Durante su utilización real por los usuarios, para juzgar su eficiencia y los resultados que con él se obtienen.
2. Durante el proceso de diseño y desarrollo, con el fin de corregir y perfeccionar el programa.

A continuación se presenta la propuesta del modelo de evaluación de software educativo, basado en el modelo MOSCA.

IV. PROPUESTA DEL MODELO DE EVALUACIÓN DE SOFTWARE EDUCATIVO

En primer lugar se describen las dos actividades que se realizarán para formular la propuesta de evaluación a partir de MOSCA y en segundo lugar, el algoritmo para la evaluación de la calidad del software educativo.

A. PROPUESTA PARA UNA PRESELECCIÓN DE SOFTWARE EDUCATIVO

Dado que la situación de las escuelas se sitúa principalmente en la adquisición de software educativo y no en la de desarrollo del mismo, se propone un procedimiento de preselección de software educativo para evaluar la gran oferta de material educativo computarizado (ver figura 1) (Galvis, 2000).

En esta propuesta de evaluación de software educativo se aplicarán, en primer lugar, los criterios de preselección sobre el material a evaluar, para tomar la decisión de una depuración inicial mediante un cuestionario de características mínimas que debe satisfacer el producto (21 preguntas); es decir, fundamentar la toma de decisiones de adquisición, considerando si el material educativo computarizado vale la pena ser evaluado con MOSCA, o si es descartado de una vez.

La razón de esta preselección se basa en:

1. La poca disponibilidad de tiempo de los docentes involucrados en la selección de software educativo.
2. En la numerosa disponibilidad de material educativo computarizado comercial que está disponible en el mercado y que puede ser usado como apoyo en el proceso de enseñanza-aprendizaje, y que además requeriría la inversión de horas de evaluación, de las cuales los docentes no disponen.

Una vez tomada la decisión de preseleccionar un software educativo, se procede a aplicar el algoritmo para la evaluación de calidad. En el caso de que se trate de desarrollo de software, se aplica el modelo que se describe a continuación para confirmar o evaluar la calidad de producción del software que está en proceso de desarrollo.

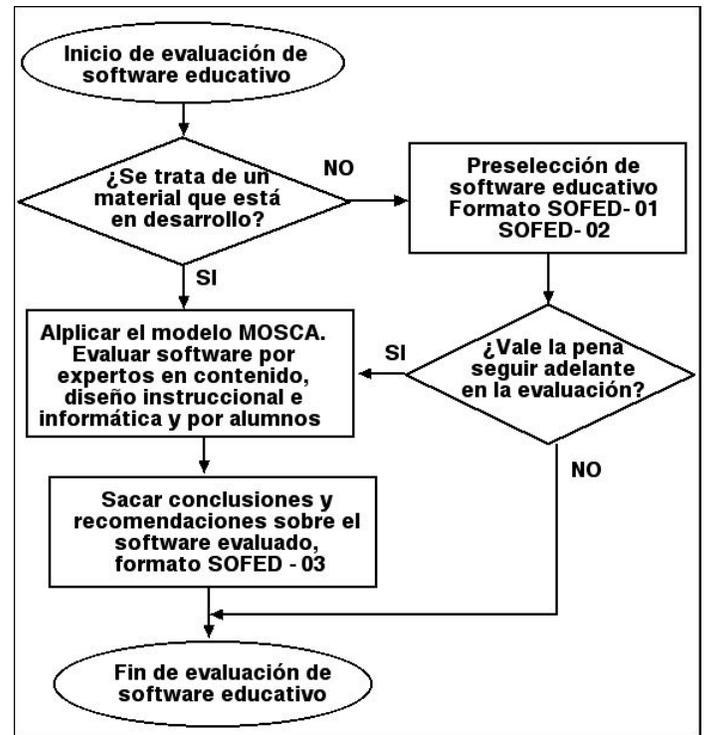


Figura 1. Método de preselección de software educativo (modificado de Galvis, 2000).

B. FORMULACIÓN DEL MODELO PROPUESTO BASADO EN MOSCA

Para el área de software educativo, se encontró que las características y métricas indicadas en MOSCA no se adaptan completamente a este tipo de software, debido a que las métricas están diseñadas genéricamente, y por lo tanto no consideraba los aspectos pedagógicos y metodológicos del proceso de enseñanza-aprendizaje que se debe tomar en cuenta al diseñar un instrumento de evaluación (Marquès, 1998).

Por tal motivo se procedió a realizar las siguientes 3 (tres) actividades para efectuar los cambios necesarios en MOSCA para adaptarlo y especificarlo en el área educativa. El resultado de estas actividades se puede apreciar en la Tabla 1.

1. Selección de las categorías. MOSCA consta de seis (6) categorías, de las cuales sólo se deben utilizar 3 (tres) para la evaluación de software educativo. Debido a que la categoría de Funcionalidad siempre debe estar presente, en esta actividad se seleccionan dos (2) categorías de las cinco (5) restantes del modelo del producto (Usabilidad, Fiabilidad, Eficiencia, Mantenibilidad y Portabilidad). Se seleccionaron Usabilidad y Fiabilidad (Grimán, Mendoza, Pérez y Rojas, 2001). La Usabilidad es seleccionada debido a que un software educativo motive al aprendizaje, es fundamental que el material educativo sea atractivo y de fácil manejo, debe generar actividades interactivas que motiven y mantengan la atención, actividades que deben ser variadas y que respondan a los

diversos estilos de aprendizaje. Se seleccionó Fiabilidad debido a que es importante que el producto funcione bajo las condiciones establecidas y mantenga un nivel específico de rendimiento.

Tabla 1.

Propuesta de categorías, características, subcaracterísticas y número de métricas, para el modelo propuesto basado en MOSCA. (modificado de Mendoza et al. 2001).

| CATEGORÍA | CARACTERÍSTICAS | SUBCARACTERÍSTICAS |
|------------------------|---|---|
| FUNCIONALIDAD (FUN) | FUN.1 Ajuste a los propósitos (118) | FUN.1.1 General (6) FUN.1.2 Objetivos de aprendizaje (10) FUN.1.3 Contenidos de aprendizaje (24) FUN.1.4 Actividades de aprendizaje (17) FUN.1.5 Ejemplos (5) FUN.1.6 Motivación (17) FUN.1.7 Retroalimentación (11) FUN.1.8 Ayudas (5) FUN.1.9 Evaluación y registro de datos (11) FUN.1.10 Metodología de enseñanza (12) |
| | FUN.2 Precisión (4) | |
| | FUN.3 Seguridad (4) | |
| USABILIDAD (USA) | USA.1 Facilidad de comprensión (91) | USA.1.1 General (13) USA.1.2 Interactividad (21) USA.1.3 Diseño de la interfaz (34) USA.1.4 Guías didácticas (23) |
| | USA.2 Capacidad de uso(11) | |
| | USA.3 Interfaz Gráfica (14) USA.4 Operabilidad (15) | |
| FIABILIDAD (FIA) | FIA.1 Madurez (11) FIA.2 Recuperación (4) FIA.3 Tolerancia a fallas (4) | |
| Total de métricas: 276 | | |

2. Selección y propuesta de las características y subcaracterísticas. Una vez seleccionadas las categorías que están relacionadas con la evaluación de software educativo (Funcionalidad, Usabilidad y Fiabilidad), se seleccionan las características asociadas a estas categorías en MOSCA, que estén relacionadas con el área educativa. Se decidió seleccionar ciertas características asociadas a la efectividad del producto y no a la eficiencia del producto, debido a que, al adquirir un software comercial, no se tiene acceso a los documentos que permiten aplicar las métricas correspondientes a esta dimensión, por lo que no es posible evaluarla. Para algunas características, tales como 'Ajuste a los propósitos' y 'Facilidad de comprensión del software', se agregó un conjunto de sub-características (14 en total) que añadieron el componente educativo a MOSCA.

3. Selección y propuesta de las métricas. Es necesaria una selección de métricas adicionales relacionadas con Funcionalidad, Usabilidad y Fiabilidad, que permitan adaptar MOSCA en el área de software educativo. En esta propuesta se realizó una investigación sobre instrumentos de evaluación de software educativo elaborados por universidades, organizaciones e instituciones gubernamentales, y expertos en la materia (Barroso et al., 1998; Del Moral, 1998; Galvis, 2000; Gómez, 1997; González, 1999; Gros (Coord.), Bernardote al., 1997; Marqués, 1998; Martínez, 1993; MVU, 2002; Navarro, 1999; OTA, 1988; PEMGU, 1999; Reeves, 1998; Stephen, 1998). De esta investigación se realizó la propuesta de un conjunto de métricas que se incorporaron en la estructura de MOSCA (276 en total).

C. DESCRIPCIÓN DEL MODELO PROPUESTO

La propuesta del modelo de evaluación de calidad de software educativo consiste en un conjunto de categorías, características, subcaracterísticas y métricas (ver Figura 2).

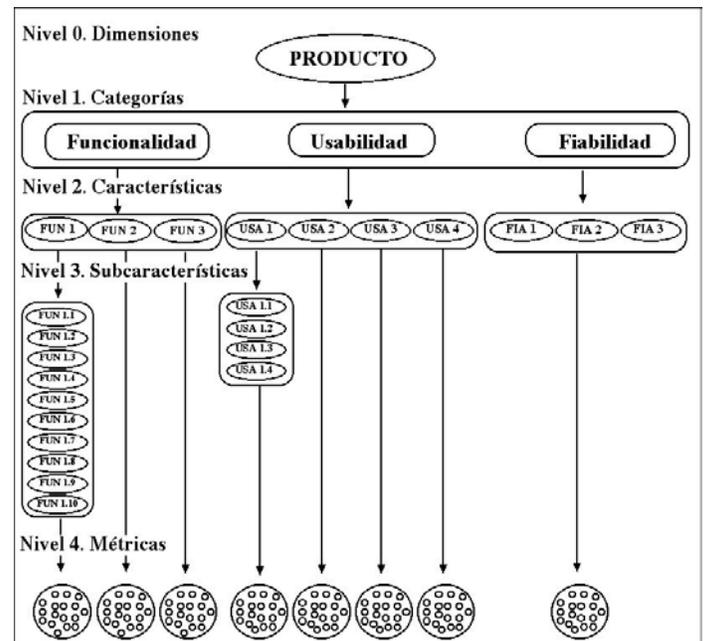


Figura 2. Propuesta del modelo de evaluación de software educativo (modificado de Mendoza et al., 2001)

La estructura del modelo consta de cuatro niveles que se explican brevemente a continuación:

- Nivel 0: Dimensiones.** Producto.
- Nivel 1: Categorías.** Se contemplan tres categorías:
 - Funcionalidad (FUN):** Es la capacidad del producto del software para proveer funciones que cumplan con necesidades específicas o implícitas, cuando el software es utilizado bajo ciertas condiciones.
 - Usabilidad (USA):** Esta categoría se refiere a la capacidad del producto de software para ser atractivo, entendido, aprendido y utilizado por el usuario bajo condiciones específicas.
 - Fiabilidad (FIA):** La fiabilidad es la capacidad del producto de software para mantener un nivel especificado de rendimiento cuando es utilizado bajo condiciones específicas.
- Nivel 2: Características.** Cada categoría tiene asociado un conjunto de características (10 en total).
- Nivel 3: Subcaracterísticas.** Para algunas de las características se asocian un conjunto de subcaracterísticas.
- Nivel 4: Métricas.** Para cada característica se propone una serie de métricas utilizadas para medir la calidad sistémica. Dada la cantidad de métricas asociadas a cada una de las características que conforman la propuesta, (276 en total) éstas no serán presentadas en el presente artículo.

En resumen, la propuesta del modelo de evaluación de software educativo consta de un total de 3 categorías, 10 características, 14 subcaracterísticas y 276 métricas.

Una vez formulado el modelo, se presenta una propuesta para la evaluación de software educativo.

D. DESCRIPCIÓN DEL ALGORITMO PARA LA EVALUACIÓN DE LA CALIDAD DE SOFTWARE EDUCATIVO

El algoritmo de evaluación que ha sido propuesto, contempla dos (2) fases:

Fase 1. Preselección (sólo en caso de adquisición por vía comercial)

1.1. Completar el formato SOFED- 01. Este formato consiste en un ficha general con información técnica y educativa que puede ser completada una sola vez por cada software a evaluar, con la información obtenida de la documentación que viene con el paquete educativo.

1.2. Completar el formato SOFED- 02, el cual ha sido diseñado para fundamentar la toma de decisiones sobre si un software educativo vale la pena ser evaluado por expertos o si se descarta. Esta evaluación se justifica en el caso de que el volumen de software a evaluar implique una cantidad de tiempo que los docentes no puedan manejar. Es necesario que en esta evaluación participen personas que estén relacionadas con la enseñanza del tema para el cual está preparado el material. Para conocer las características de un software, el profesor normalmente deberá leer el manual e interactuar con él con el propósito de determinar sus objetivos, los contenidos, el planteamiento didáctico, el tipo de actividades que presenta y la calidad técnica; es decir, deberá realizar una pre-evaluación del software. Para facilitar esta evaluación objetiva de las características de un software, se propone una ficha de catalogación y evaluación que permitirá recoger los rasgos principales y algunas valoraciones generales sobre sus aspectos técnicos, pedagógicos y funcionales. Se aplica un cuestionario que consta de 21 preguntas relacionadas con las métricas que posteriormente se proponen.

Una vez preseleccionados los software educativos que requieran de una evaluación más precisa, se procede a aplicar el algoritmo MOSCA, modificado y ampliado, para medir la calidad del producto de software con un enfoque sistémico.

Fase 2. Estimar la calidad del producto de software educativo con enfoque sistémico. A través de la ejecución de esta fase, se evalúa la calidad de producto de software; para tal fin, se siguen 3 actividades, las cuales son descritas a continuación:

2.1. Estimar la calidad de la Funcionalidad del producto. Según MOSCA, se establece que siempre y en todos los casos se debe medir primero la categoría Funcionalidad del producto, donde las características que se proponen para esta categoría deben ser

cumplidas para proceder a la próxima actividad. De lo contrario, la evaluación finaliza. Para la categoría funcionalidad se propone que al menos se cumpla la característica “Ajuste a los propósitos”, más una de las dos características restantes, es decir, “Precisión” o “Seguridad” (ver tabla 2).

Tabla 2.

Características mínimas que deben ser satisfechas para cada categoría (modificado de MOSCA, Mendoza et al., 2001).

| Categorías del producto | Características mínimas que deben ser satisfechas |
|-------------------------|--|
| Funcionalidad | 1. Ajuste a los propósitos 2. Precisión o Seguridad |
| Usabilidad | 3 (de 4) |
| Fiabilidad | 2 (de 3) |

2.2. Estimación de calidad para cada categoría. Para las dos (2) categorías seleccionadas previamente (Usabilidad y Fiabilidad), se debe:

2.2.1. Aplicar las métricas propuestas en el submodelo del producto para las categorías seleccionadas aparte de Funcionalidad; es decir, para Usabilidad y Fiabilidad.

2.2.2. Normalizar los resultados de las métricas a una escala del 1 al 5. La normalización de los resultados es llevada a cabo de acuerdo a la Tabla 3.

Tabla 3.

Normalización de las métricas para la evaluación del producto (Adaptado de PEMGU, 1999).

| No aplica o No sabe | 1 | 2 | 3 | 4 | 5 |
|---------------------|------------------|---------------------|-----------------------|--------------------|------------------------|
| | Nunca | Pocas veces | Algunas veces | Casi siempre | Siempre |
| | Muy mal | Mal | Normal | Bueno | Muy bueno |
| | Nada | Alguno | Normal | Casi | Mucho |
| | No tiene | Básico | Mediano | Alto | Muy alto |
| No aplica (NA) | Ninguno | Muy pocos | Pocos | Casi todos | Todos |
| | Muy baja | Baja | Media | Alta | Muy alta |
| No sabe (NS) | No | Poco | Medianamente | Casi todo | Completamente |
| | Inaceptable | Debajo del promedio | promedio | Buena | Excelente |
| | No está definido | Poco definido | Medianamente definido | Casi todo definido | Completamente definido |
| | No | | Promedio | | Si |

2.2.3. Verificar que el 75% de las métricas se encuentran dentro de los valores óptimos (mayor o igual a 4) para cada una de sus subcaracterísticas y características. Si no se cumple el 75% de las métricas asociadas, entonces esta subcaracterística o característica tendrá calidad nula. En el caso de institutos educativos la población de evaluadores serán docentes de aula y de informática y los estudiantes; y en el caso de empresas desarrolladoras de software educativo, la población de evaluadores serán los expertos en contenido, diseño instruccional, en informática y el coordinador o líder del proyecto. Las pautas para promediar las métricas se aplican tal como lo señala el algoritmo MOSCA en este apartado.

2.2.4. Evaluar la categoría. En MOSCA, una categoría es satisfecha si el número de características es altamente satisfecho; es decir, si satisface el 75% de las características asociadas a la categoría, tal como se aprecia en la tabla 2. Tratándose de software educativo, donde existen características que son imprescindibles que estén presentes, se proponen las características mínimas satisfechas que debe tener cada categoría del producto para que ésta pueda ser satisfecha. Luego se continúa con la fase 2.3.

2.3. Estimar la calidad del producto partiendo de las categorías evaluadas. En este punto se utiliza el algoritmo de MOSCA, con ciertas modificaciones (ver tabla 4). Nótese que para que un software educativo obtenga calidad intermedia, debe ser satisfechas las categorías de Fiabilidad y Usabilidad.

Una vez terminada la evaluación del producto, y sólo en caso de que éste obtenga al menos el nivel de calidad intermedio:

1. Se procederá a la toma de decisión de adquisición del software educativo en el caso de institutos o empresas de acuerdo a los recursos económicos.
2. Se procederá a evaluar la *calidad del proceso* a través del sub-modelo del mismo (fase 2 y 3 de MOSCA), en el caso de empresas o institutos desarrolladores de software educativo. El algoritmo de estas fases escapa al alcance de esta propuesta.

Tabla 4.

Nivel de calidad del producto con respecto a las categorías satisfechas para el producto software educativo (modificado del modelo MOSCA, Mendoza et al, 2001).

| Funcionalidad | Usabilidad | Fiabilidad | Nivel de calidad |
|---------------|---------------|---------------|------------------|
| Satisfecha | No satisfecha | No satisfecha | Básico |
| Satisfecha | No satisfecha | Satisfecha | Básico |
| Satisfecha | Satisfecha | No satisfecha | Intermedio |
| Satisfecha | Satisfecha | Satisfecha | Avanzado |

V. CASO DE ESTUDIO: APLICACIÓN DEL ALGORITMO DE EVALUACIÓN DE LA PROPUESTA

Para la prueba de este modelo de evaluación de software educativo con enfoque sistémico de calidad, se selecciona una variedad de 4 (cuatro) títulos, dos en español y dos en inglés, para distintas áreas de la educación y para diferentes edades y se aplica el instrumento de evaluación (ver tabla 5).

Se aplicaron dos tipos de cuestionarios, que son estándar para todos los software. El primero dirigido a los docentes y el segundo dirigido a los estudiantes. Ambos cuestionarios se desarrollaron a partir de las métricas que conforman el modelo propuesto. Cada software educativo fue evaluado por dos docentes y dos alumnos. A continuación se muestran los resultados.

Tabla 5.

Lista de software educativo que se consideró para la evaluación de la propuesta.

| | Software educativo | Materia | Edad | Fabricante |
|---|--|-----------------------|-------------------|---|
| 1 | Los aventureros. 3er. grado. | 3er.grado Curricular | 8 – 10 años | The Learning Company |
| 2 | Kids English | Inglés | 4 a 8 años | Syracurse Language |
| 3 | Trampolín Tercer ciclo | 3er. Grado Curricular | 8 a 10 años | Knowledge Adventure - Anaya Interactiva |
| 4 | Smart Start in English (Hable Inglés ¡ya!) | Idioma Inglés | 10 años a adultos | Syracurse Language – Language Larousse |

Funcionalidad: En la Figura 3 se muestran los porcentajes alcanzados por los cuatro (4) software educativos, en cuanto a los requerimientos de calidad asociados a *Funcionalidad*.

Como se puede observar en la Figura 3, sólo tres de los cuatro software educativos cumplen con las característica de ‘Ajuste a los propósitos’ y ‘Precisión’, por lo tanto se concluye que la categoría *Funcionalidad* es satisfecha para lo software educativos “Los aventureros. 3er. Grado”, “Trampolín tercer ciclo” y “Smart Start in English (Hable Inglés ¡ya!)”; y se procede a continuación con el algoritmo del modelo ampliado de MOSCA.. El software educativo “Kids English” no cumple con dos de las tres características de *Funcionalidad*, por lo que esta categoría no está satisfecha para este software educativo.

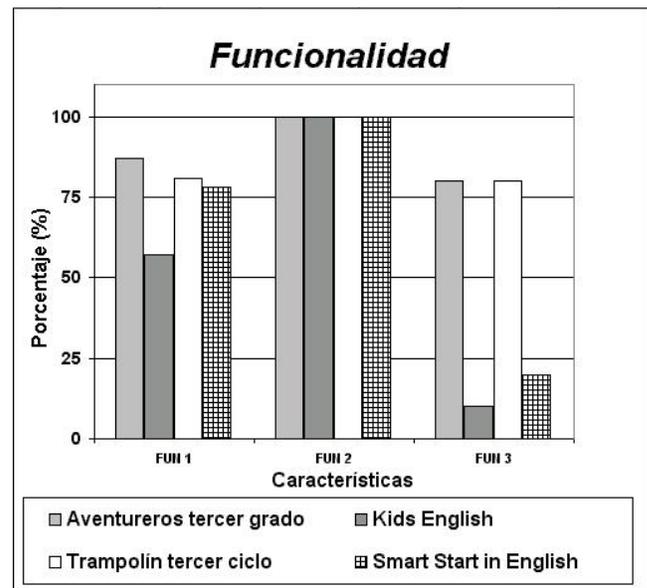


Figura 3. Porcentaje de satisfacción de los software educativos frente a las tres características de *Funcionalidad*.

Usabilidad: La Figura 4 muestra los porcentajes alcanzados por las características asociadas a la categoría *Usabilidad*.

Como se puede observar en la Figura 4, sólo dos de los cuatro software educativos cumplen con tres de las cuatro características asociadas a Usabilidad, por lo tanto se concluye que la categoría Usabilidad es satisfecha para “Los aventureros. 3er. Grado” y “Trampolín tercer ciclo”. Para el los software “Smart Start in English (Hable Inglés ¡ya!)” y “Kids English”, las características de ‘facilidad de

comprensión del uso software' y 'operabilidad' no alcanzaron el 75% de la satisfacción, por lo que se concluye que para estos software, la categoría Usabilidad no es satisfecha.

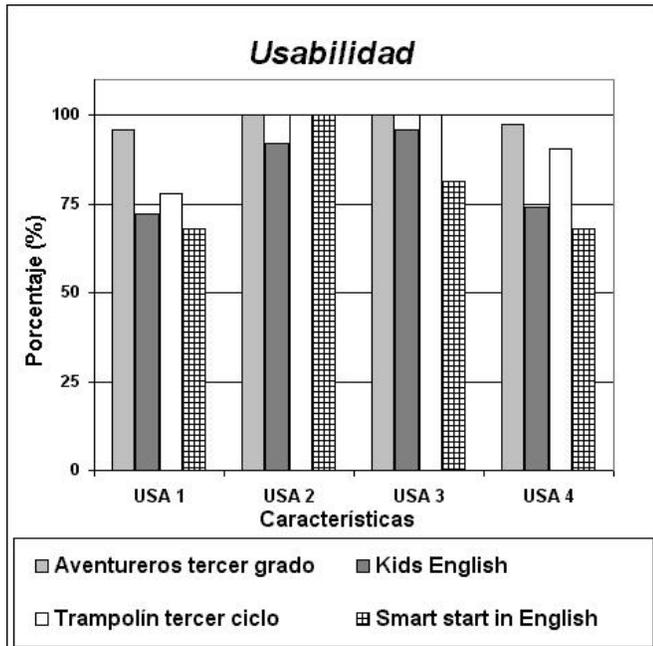


Figura 4. Porcentaje de satisfacción de los software educativos frente a las cuatro características de Usabilidad.

Fiabilidad: La Figura 5 muestra los porcentajes alcanzados por las características asociadas a la categoría Fiabilidad. Como se puede observar en la Figura 5, todos los software educativos cumplen al menos, dos de las tres características asociadas a Fiabilidad, por lo tanto se concluye que la categoría Fiabilidad es satisfecha para “Los aventureros. 3er. Grado”, “Kids English”, “Trampolín tercer ciclo” y “Smart Start in English (Hable Inglés ¡ya!)”.

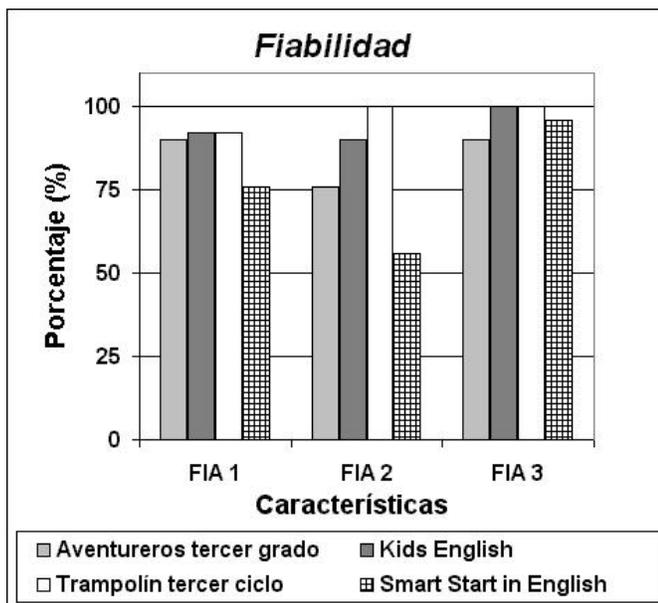


Figura 5. Porcentaje de satisfacción de los software educativos frente a las tres características de Fiabilidad.

VI. RESULTADOS

Analizando los resultados de las Figuras 3, 4 y 5, se tiene la tabla 6, la cual muestra que dos (2) de los cuatro (4) software evaluados presenta nivel de calidad avanzado; es decir, los software educativos “Los aventureros. 3er. Grado” y “Trampolín tercer ciclo” presentan nivel de *calidad Avanzada*. El software “Smart Start in English (Hable Inglés ¡ya!)” presenta nivel de *Calidad Básica*, debido a que la categoría Usabilidad no fue satisfecha. El software “Kids English” presenta nivel de calidad nula, debido a que la categoría Funcionalidad y Usabilidad, no fueron satisfechas.

Tabla 6.

Resultados de las evaluaciones de los cuatro software educativos según el modelo propuesto.

| Software educativo | Categoría | | | Nivel de calidad |
|--------------------------|---------------|---------------|------------|------------------|
| | Funcionalidad | Usabilidad | Fiabilidad | |
| Aventureros tercer grado | Satisfecha | Satisfecha | Satisfecha | Avanzada |
| Kids English | No satisfecha | No satisfecha | Satisfecha | Nula |
| Trampolín tercer ciclo | Satisfecha | Satisfecha | Satisfecha | Avanzada |
| Smart Start in English | Satisfecha | No satisfecha | Satisfecha | Básica |

VII. CONCLUSIONES

Se propone un modelo para la estimación de la calidad de software educativo bajo un enfoque de calidad total sistémica. Este modelo proporciona una valiosa herramienta de evaluación para el docente que tiene la necesidad de conocer el valor educativo de un software, su calidad y el uso posible de un software en el ambiente educativo, como criterios imprescindibles para su adquisición y uso por parte de las instituciones educativas. Es además, una excelente herramienta de evaluación para hacer ajustes durante el proceso de diseño o desarrollo de software educativo, de una evaluación de prueba, antes de la edición definitiva.

Además, se propone un algoritmo para la estimación de la calidad sistémica, haciendo uso del prototipo propuesto.

Tanto el modelo propuesto como el algoritmo de aplicación asociado a él, fueron probados a través de un estudio de campo que permitió medir el desempeño de este modelo objetivamente en cuatro software educativos utilizados a nivel de edad escolar y adolescentes.

AGRADECIMIENTOS

Los autores desean agradecer a los docentes y alumnos de la Unidad Educativa Institutos Educativos Asociados (IEA), Caracas, Venezuela, por su valiosa colaboración en la evaluación de los software educativos utilizados para este estudio.

REFERENCIAS

Barroso, J., Mendel, J.L., Valdeverde, J. (1998) "Evaluación de los medios informáticos: una escala de evaluación para el software educativo". En Cebrían, M. et al. "Creación de materiales para la innovación con nuevas tecnologías": *EDUTEC* 97, 355-358. Málaga:

- ICE Univ. Málaga. Disponible: http://www.ice.uma.es/edutec97/edu97_c3/2-3-08.htm.
- Beichner, R. (1994) Multimedia editing to promote science learning. *Journal of Educational Multimedia and Hypermedia*. 3 (1) 55 – 70.
- Borrás, I., Lafayette, R. (1994) "Effects of multimedia course subtitling on the speaking performance of college students in French", *The modern Language Journal*, 78, 61 – 75.
- Callaos, N. and Callaos, B. (1993) "Designing with Systemic Total Quality", en International Conference on Information Systems, N. Callaos and B. Sánchez (eds.), *International Institute of Informatics and Systemics*, Orlando, pp. 548-560.
- Cohen, S., Chechile, R., Smith, G., Tsai, F., Burns, G. (1994) "A method for evaluating the effectiveness of educational software" *Behavior Research Methods, Instruments & Computers*, (26) 236 - 241.
- Cohen, S., Tsai, F., Chechile, R. (1995) "A model for assessing student interaction with educational software". *Behavior Research Methods, Instruments & Computers*, 251-256.
- Del Moral, E. (1998). "El desarrollo de la creatividad y las nuevas herramientas tecnológicas". *Comunicación Educativa y Nuevas Tecnologías*, 51-66. Barcelona: Praxis.
- Díaz-Antón, G. (2002) "Uso de software educativo de calidad como herramientas de apoyo para el aprendizaje". Jornadas educativas: "La escuela como instrumento de cambio", IEA, Abril, Caracas. Disponible: <http://www.academia-interactiva/articulos.html>
- Galvis, A., (2000) "Evaluación de MECs por juicio de expertos", Capítulo 10 del libro: "Ingeniería de software educativo" 2da. reimpresión. Universidad de Los Andes. Colombia.
- Gómez, M.T. (1997) "Un ejemplo de evaluación de software educativo multimedia". En Cebrían, M. et al. "Creación de materiales para la innovación con nuevas tecnologías": *EDUTEC97*. Málaga: ICE Univ. Málaga, Disponible: http://www.ice.uma.es/edutec97/edu97_c3/2-3-03.htm
- González, M., (1999) "Evaluación de software educativo. Orientaciones para su uso" Proyecto Conexiones. Universidad de EAFIT. Disponible: <http://www.eafit.edu.co/articulos/evalSE.htm>
- Grimán, A, Mendoza, L., Pérez M., Rojas, T. (2001) "Modelo de calidad de software educativo: aplicación del estándar ISO/IEC 9126." Valencia: 2da. Edición EDUWEB – Congreso Venezolano de Educación e Informática. Julio. 2001, Venezuela.
- Gros, B. (2000) "Del software educativo a educar con software". Artículo publicado en el número monográfico de la revista digital *Quaderns Digital*. Disponible: <http://www.quadernsdigitals.net/articuloquaderns.asp?IdArticle=374>
- Gros, B. (Coord.), Bernardo, A., Lizano, M., Martínez, C., Panadés, M., Ruiz, I. (1997) "Diseños y programas educativos, pautas pedagógicas para la elaboración de software". Editorial Ariel, S.A. 149 pp.
- ISO/IEC 9126: 1991, JTC 1/SC 7. *Information technology - Software product evaluation - Quality characteristics and guidelines for their use*, JTC 1 Organization, Montréal, Québec, 1991.
- Jonassen, D., Reeves, T. (1996) "Learning with technology: using computers as cognitive tools". In Jonassen (Ed.) *Handbook of research for educational communications and technology* (pp 693 – 719). New York. Macmillan.
- JTC 1/SC 7. "Software Process Assessment", TR 15504, on-line, WG 10: Software Process Assessment. ISO/IEC Organization, 1991, <http://www.sqi.gu.edu.au/spice/>
- Kitchenham, B., Linkman, S., and Law, D. (1996) "DESMET: A method for Evaluating Software Engineering Methods and Tools", *Technical Report [ISSN 13 53-7776]* (TR96:09), Department of Computer Science, Keele University, pp.1-67.
- Liu, M., Reed, W. (1995) "The effect of hypermedia-assisted instruction on second language learning". *Journal of Educational computing Research*. 12, 159 –175.
- Marquès, P. (1998). "La evaluación de programas didácticos". *Comunicación y Pedagogía*, (149) 53-58, Barcelona. Disponible: <http://www.xtec.es/~pmarques/tecnoedu.htm>
- Martínez, R., Sauleda, N. (1993) "La evaluación de software educativo en el escenario de la evolución de los paradigmas educativos". *Enseñanza*, 161 – 174.
- Mayer, R., Shustack, M. Blanton, W. (1997) "What do children learn from using computers in an informal, collaborative setting?", *Educational Technology*, March – April, 39 (2) 27 – 31.
- Mendoza, L., Pérez, M., Rojas, T. (2001) "Modelo sistémico para estimar la calidad de los sistemas de software". (MOSCA).ASOVAC, Acta Científica Venezolana, (53:3) p 435. LISI, Universidad Simón Bolívar. Disponible: <http://www.lisi.usb.ve/publicaciones>
- MVU, Michigan Virtual University (2002) "Standard Quality on-line courses". Disponible: <http://ideos.mivu.org/standards>
- Navarro, E. (1998) "Evaluación de materiales multimedia". Proyecto SAMIAL. *Comunicación y Pedagogía*. 157, 36 – 39.
- Ortega M.; Pérez, M.; Rojas T. (2000) "A Model for Software Product Quality with a Systemic Focus", en The 4th World Multiconference on Systemics, Cybernetics and Informatics SCI 2000 and The 6th International Conference on Information Systems, Analysis and Synthesis ISAS 2000. Orlando, USA, Julio. Disponible: <http://www.lisi.usb.ve/publicaciones>
- OTA (1988) "Power on! New tools for teaching and learning". U.S. Congress, Office of Technology Assesment, SET-379, September, Government Printing Office, Washington D.C.
- PEMGU, (1999) "Pedagogical evaluation methods & guidelines for multimedia applications" Partners: Epral (Portugal), Colegio Irabia (Spain) and Holbaek Technical College (Denmark), DEL (Denmark), the University of Cologne (Germany) and Olivetti (Italy) . Disponible: <http://www.irabia.org/pemgu/>
- Pérez, M., Rojas T., Ortega, M., y Álvarez, C.. (1999) "Toward Systemic Quality: Case study", en 4Th Squad Meeting, Squad, Porlamar, Venezuela, July. Disponible: <http://www.lisi.usb.ve/publicaciones>
- Pérez, M.; Mendoza, L.; Rojas, T.; Grimán, A. (2001) "Systemic Quality Model for System Development Process: Case Study", en: Seventh Americas Conference on Information Systems AMCIS 2001. Boston, USA, Agosto 2001, Dsponible: <http://www.lisi.usb.ve/publicaciones>
- Poole, B. (1999) "Lo que las investigaciones dicen sobre el uso de la tecnología informática para la enseñanza y el aprendizaje". En Capítulo 1 del mismo autor: "Tecnología Educativa: educar para la sociocultura de la comunicación y del conocimiento". 2da. Edición. McGraw-Hill/ Interamericana de España, S.A.U. pp 1-21.
- Pressman, Roger. (2002) "Ingeniería de Software: Un enfoque Práctico". McGraw Hill.
- Reeves, T. (1996) "Learning with software: pedagogies and practices: Evaluating what really matters in computer-based education", University of Georgia.. Disponible: http://www.nib.unicamp.br/recursos/distance_education/evaluating-cbe.html
- Stephen, B. (1998) "Evaluating checklist. Evaluating training software". Lancaster University. Disponible: http://www.keele.ac.uk/depts/cs/Stephen_Bostock/docs/evaluationchecklist2.html
- Yildirim, Z., Yasar, M., Asku, M. (2001) "Comparison of Hypermedia Learning and traditional instruction on knowledge acquisition", *The Journal of Educational Research*. March- April, 94 (4), 207 – 214.