

# “Propuesta de un Método Generador de Metodologías para Desarrollo de Software Educativo”

Isaías Pérez Pérez<sup>1</sup>, Silvia Soledad Moreno Gutiérrez<sup>2</sup>  
isaiasp7@hotmail.com, silviam@uaeh.edu.mx

<sup>1</sup> Universidad Autónoma del Estado de Hidalgo, ICBI

<sup>2</sup> Universidad Autónoma del Estado de Hidalgo, Escuela Superior Tlahuelilpan

Entornos virtuales de aprendizaje  
La contribución de los espacios y recursos virtuales de aprendizaje

## RESUMEN

El presente trabajo, basado en el marco metodológico de la llamada Metodología Participativa (Tello Fernández, 2006), propone definir un método general, que sirva para definir propuestas de mejoramiento o creación de metodologías para desarrollo de aplicaciones de software educativo. Esto se justifica en base a que, haciendo el análisis de las principales ventajas y desventajas que presentan algunas de las actuales metodologías de este tipo (Panqueva, 2001; Pesce, 1996; Vaughan, 1994; Yokohama, 2008), éstas aun se encuentran limitadas en la especificación y utilización de técnicas, herramientas de software y otros elementos conceptuales (De Miguel Castaño y Piattini Velthuis, 1993), que son necesarios para definir las como metodologías generadoras de productos software de calidad, uno de los principales lineamientos que persigue la Ingeniería de Software (Pressman, 2006).

**Palabras clave:** Método, metodología, software educativo.

## INTRODUCCIÓN

El software educativo se define como programas para computadora, creados con la finalidad específica de ser utilizados como medio didáctico, es decir, para facilitar los procesos de enseñanza y de aprendizaje (Fernández Muñoz, 1998).

Se sabe que la construcción de un sistema de software implica la toma de decisiones sobre la arquitectura del mismo, como definir los componentes del sistema de software y sus interacciones. Estas decisiones son cruciales para el éxito o fracaso, por lo que se requiere seleccionar un "proceso de desarrollo con el fin de obtener la calidad y cumpla con los requerimientos establecidos". Hoy en día, esta problemática y sus respectivas soluciones, es el campo de estudio de la ingeniería de software (Alcántara Olguín, 2008).

El intento de la Ingeniería de Software es proporcionar un marco de trabajo para construir software con mayor calidad (Pressman, 2006). Debido a esto, diversos autores han utilizado la filosofía de la ingeniería de software para la elaboración de material multimedia interactivo, logrando de esta manera que el proceso de desarrollo y mantenimiento del software educativo sea una actividad que dependa de pautas establecidas, con modelos conceptuales y herramientas de trabajo, y no del arte de aquellos que tengan la experiencia exclusivamente.

Para lograr software educativo con las condiciones deseadas, se deben incorporar dentro de sus fases de análisis y diseño, aspectos didácticos y pedagógicos, es decir, el diseño instruccional, de manera que se faciliten y garanticen la satisfacción de las necesidades educativas del público al que va dirigido el software. Se deben involucrar también a los usuarios, para conseguir identificar necesidades o problemas específicos y se puedan establecer mecanismos de resolución adecuados, así como apoyar

cada una de las fases en sólidos principios educativos, comunicativos y computacionales (Alcántara Olgúin, 2008).

## **ESTADO DEL ARTE Y PROBLEMÁTICA PRESENTADA**

En las distintas áreas de la ingeniería de software se han realizado importantes esfuerzos para encontrar las metodologías más adecuadas; esto se debe al gran impacto que una metodología tiene en el desarrollo de un producto de software, ya sea en lo que se refiere a los costes y plazos de entrega del mismo, como a la calidad y mantenimiento del producto (De Miguel Castaño y Piattini Velthuis, 1993).

Por dicha razón, las diversas metodologías para el desarrollo de software educativo tienen la principal característica de ser abiertas y adaptables, lo cual permite que se lleven a cabo sólo aquellas actividades y modelos que sean necesarios o útiles para el proyecto a desarrollar (Alcántara Olgúin, 2008).

Algunos ejemplos de metodologías para desarrollo de software educativo existentes hoy en día, son la metodología para el desarrollo educativo, la metodología de análisis, diseño, desarrollo, evaluación y administración de ambientes virtuales educativos y la metodología para desarrollo de software educativo (Panqueva, 2001; Pesce, 1996; Vaughan, 1994; Yokohama, 2008). Al llevar a cabo un breve análisis de las ventajas y desventajas presentes en éstas, se puede observar que en general, abordan aspectos importantes del desarrollo de software y de tipo educativo, pedagógico y didáctico, además de que presentan un número adecuado de fases o etapas.

En cuanto a sus desventajas más evidentes, las metodologías de desarrollo para diverso tipo de software educativo, como libros electrónicos, tutoriales, simuladores, sistemas expertos, lenguajes de

programación, entre otros, presentan la limitante de que en cada una de las etapas o fases de la metodología planteada, es escaso encontrar una cantidad adecuada de técnicas y herramientas de software definidas que cubran la mayoría de los requerimientos específicos de diseño y construcción. Cada aplicación de software educativo presenta ciertos matices específicos. El no llegar a cubrir estas características adecuadamente, se ve reflejado en lograr un producto de software de baja calidad, lo cual es contrario a los lineamientos que busca la ingeniería de software (Pressman, 2006).

Es contraproducente que por una parte, existan investigaciones relativamente extensas en cuanto a la evaluación del software educativo como producto, como si se tratará de un proceso de medición de la calidad del mismo (Cova, Arrieta y Riveros, 2008) y (González Castañón, 2009), atendiendo dos rubros principales: su desarrollo técnico computacional y los aspectos de tipo educativo, pedagógico y didáctico que contempla; y por otra parte, que las metodologías propuestas para su desarrollo, carezcan de ciertas características necesarias para la adecuada definición general de una metodología, como es el caso de la escasa fundamentación teórica (rigurosidad), en algunos casos, y la falta de especificación de técnicas y herramientas de software necesarias para el desarrollo de software (adopción de estándares) (De Miguel Castaño y Piattini Velthuis, 1993). Parece ser que hoy en día, el software educativo es evaluado como un producto terminado, descuidando en parte, el proceso que permitió desarrollarlo. Como cualquier tipo de software, la calidad del software educativo, como lo marca la ingeniería de software, no se basa solamente en la evaluación del producto terminado, sino primordialmente en el control del proceso que permitió su desarrollo (Pressman, 2006).

Por su parte, el modelo del diseño instruccional (Yokohama; Torres Yañez; R. Hilera, 2008), que es una opción utilizada dentro de las etapas de análisis y diseño de las metodologías para desarrollo de software educativo, utilizado con el fin de coordinar el esfuerzo de desarrollo de las aplicaciones,

presenta las mismas dificultades que las anteriores metodologías mencionadas, en cuanto a su carencia de especificar las técnicas y herramientas de software necesarias, en la amplia diversidad de tipos de software educativo existentes.

### **PREGUNTAS DE INVESTIGACIÓN**

Se ha mencionado que las metodologías de desarrollo de software educativo existentes, se encuentran en un estado de desarrollo incompleto; por tanto, surgen las siguientes preguntas:

¿Cómo mejorar y enriquecer conceptual y metodológicamente las metodologías existentes?

Si fuese el caso, ¿es posible diseñar nuevas metodologías para desarrollo de software educativo?.

### **PROPUESTA DE SOLUCIÓN**

Para dar solución a las anteriores interrogantes, se propone llevar a cabo el desarrollo de un método general, denominado “Constructor de metodologías para desarrollo de software educativo”, que permitirá el mejoramiento o construcción de futuras propuestas de metodologías para desarrollo de diversos tipos de software educativo, tomando como modelo a seguir el método denominado “Constructor de metodologías para desarrollo de aplicaciones web”, propuesto en la tesis de Hernández Ortiz (2007), el cual se basa a su vez, en ciertos elementos conceptuales de otro marco metodológico, conocido como la metodología participativa.

### **METODOLOGÍA O TÉCNICA USADA**

Para diseñar el método que sirva para generar propuestas de metodologías para desarrollo de software educativo, se utilizará como guía las directrices de la llamada metodología participativa, que toma en

cuenta gran parte de las opiniones consultadas a los diversos usuarios implicados en un proyecto específico.

La metodología participativa promueve y procura la participación activa de todos los integrantes del grupo. En este sentido, la metodología participativa busca que los participantes unifiquen su experiencia con la de los otros, en lo que su aprendizaje se contextualiza en su realidad cotidiana y se ajusta a las particularidades de su proceso de desarrollo.

Los aspectos más relevantes de la metodología participativa, en una actividad de desarrollo de software, se pueden visualizar como el esquema de trabajo siguiente:

- 1) *Partir de lo que se conoce sobre el tema (aspectos relevantes del desarrollo de software educativo)*. Partir de la práctica, la experiencia, y los conocimientos que tienen sobre el tema, diseñadores y usuarios.
- 2) *Reflexionar y profundizar sobre el tema*. Tener presentes los aportes sistematizados, tanto de diagnóstico como de propuesta de los expertos y otros aportes teóricos sobre el tema, como el análisis de otras metodologías existentes.
- 3) *Determinar lo que se puede hacer*. Precisar el sentido del diseño, con el fin de definir que es lo que se puede hacer y que se necesita en una propuesta metodológica (Tello Fernández, 2006).

## DESARROLLO DE LA PROPUESTA

Para evitar utilizar un marco metodológico en el desarrollo de software educativo, que no sea lo suficientemente detallado, es necesario modificar o en su caso construir, una metodología que sirva para algunos tipos de software educativo similares y que especifique adecuadamente cada una de sus actividades, permitiendo que sean claramente entendidas y llevadas a cabo por el que las aplique. Pero surge la pregunta: ¿cómo lograr este objetivo?, ¿cómo mejorar o construir una propuesta de metodología con dichas características?. Por tal motivo, en el presente trabajo se presenta un método general, el cual, basándose en el esquema de trabajo de la metodología participativa ya mencionada, permita generar propuestas de metodologías para desarrollo de software educativo. A este método, se le ha denominado: “**Método Generador de Metodologías para Desarrollo de Software Educativo**”. Los pasos del citado método, son los siguientes:

***Paso I: Comentarios y recomendaciones de los desarrolladores.*** Analizar los comentarios y recomendaciones recabados de programadores, diseñadores gráficos, pedagogos, didáctas, psicólogos, entre otros, tomando esta información como punto de partida para el diseño de la propuesta metodológica.

***Paso II: Estudio de diversas metodologías conocidas.*** Se analizan las diversas metodologías para desarrollo de software educativo disponibles, en aspectos como sus características generales, proyectos, productos educativos y usuarios a los que van enfocados, así como las ventajas y desventajas que ofrecen cada una de ellas. Es recomendable en este paso, construir una tabla comparativa de metodologías, con los principales rubros a evaluar, con el fin de lograr un análisis más exhaustivo.

***Paso III: Generación de la propuesta de metodología.*** Este es el paso más laborioso del método. Aquí se

propone un esquema de todas las actividades que deben comprender las fases o etapas de la propuesta, así como el número específico de estas, tomando en consideración los siguientes aspectos:

a) Los componentes básicos que una metodología debe comprender, son:

- *Herramientas.* Es cualquier recurso particular a disposición de la metodología (Batini et al., 1981). Herramientas serán las tablas, diagramas, graficas, teorías etc.; que se aplican en las distintas fases del diseño.
- *Modelos.* Es un sistema formal y abstracto que permite describir sus componentes y los datos relacionados a ellos, de acuerdo con reglas predefinidas. (Ullman, 1990).
- *Lenguajes de programación.* Consisten en un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Están siempre basados en un determinado modelo.
- *Documentación.* Es un proceso de preparación de la información disponible, sobre el software que se está desarrollando.
- *Reglas.* Estas actuarán sobre los elementos de entrada de cada fase, para conseguir de manera semiprogramable las salidas de cada una de ellas.

b) Las características, que en general una metodología para desarrollo de software debe tener, son:

- *Claridad y Comprensibilidad.* Debe permitir ser explicada a distintos tipos de usuarios.

- *Capacidad de Soportar la Evolución de los Sistemas.* Deberá soportar la evolución del software sin problemas.
  - *Facilitar la Portabilidad.* Facilidad para que el producto de programación pueda ser transferido de un sistema informático a otro o de un entorno a otro.
  - *Versatilidad Respecto al Tipo de Aplicaciones.* No debe estar orientada a un solo tipo de aplicaciones concretas, sino que puede utilizarse en aplicaciones similares.
  - *Flexibilidad.* Pueda utilizarse tanto en proyectos grandes como pequeños.
  - *Rigurosidad.* Siempre que sea posible, tratar de apoyarla con sólidos fundamentos teóricos.
  - *Adoptar Estándares.* Se debe procurar aplicar todos aquellos estándares de diseño de software (uso de técnicas y herramientas de software).
  - *Automatización.* Para que una metodología resulte útil, debe poderse automatizar, así como su documentación (De Miguel Castaño y Piattini Velthuis, 1993).
- c) Se recomienda que para cada nivel o etapa de la metodología a proponer, estas correspondan con decisiones de diseño bien definidas, tratando de no plantear demasiados niveles, ya que acarrearía muchos conceptos implícitos y sería muy sensible a la interpretación individual de cada diseñador (De Miguel Castaño y Piattini Velthuis, 1993).
- d) Otros aspectos necesarios a considerar, en el proceso de diseño de la propuesta de la metodología, son:
- La naturaleza del proyecto

- Los resultados que desean obtener los que proponen el proyecto
- Las características de los diferentes interesados
- La experiencia participativa que los interesados hayan tenido (De Miguel Castaño y Piattini Velthuis, 1993).

e) Por último, es conveniente tener en cuenta las siguientes consideraciones:

- Explicitar claramente el propósito del proyecto o actividad propuestos.
- Realizar sugerencias en cuanto a la metodología y las técnicas "más adecuadas".
- Usar una variedad de técnicas en la propuesta de la metodología, las cuales deberán complementarse mutuamente.
- Los responsables de la aplicación de la metodología generada, deben tener experiencia y ser sensibles al medio cultural y social en que se aplicará.

*Paso IV: Enriquecimiento teórico.* Justificar en la medida de lo posible, cada una de las fases o etapas de la metodología, en base a los elementos teóricos de otras metodologías, con las cuales presente ciertas semejanzas. De igual forma, tratar de especificar las subetapas, procedimientos, técnicas y en su caso herramientas, que puedan o deban utilizarse (De Miguel Castaño y Piattini Velthuis, 1993).

## **CONCLUSIONES Y TRABAJOS FUTUROS**

Si es difícil encontrar metodologías específicas que eviten la generalidad excesiva o que se adapten a ciertos tipos de software, como es en el caso del desarrollo de aplicaciones de software educativo, es aún más escaso encontrar un método que aborde el problema de modificar o generar propuestas de

metodologías de desarrollo. Para ofrecer una alternativa a esta problemática, el presente trabajo plantea un método para modificar o generar esquemas de nuevas metodologías para desarrollo de software educativo. Se espera, que en un futuro, este trabajo aporte un aspecto metodológico más, al campo de la ingeniería de software actual.

Por su parte, el llevar a cabo la aplicación del método presentado, es decir, la generación de una propuesta de metodología no es una tarea fácil. Requiere de un análisis profundo de otras metodologías existentes; de las razones que las llevaron a su planteamiento, además de que fueron enriquecidas por la experiencia de las personas que las llevaron a cabo en diversos proyectos. Esto implica además, que es necesario enriquecer la propuesta de metodología con aspectos prácticos, llevar a cabo las pruebas necesarias para comprobar si la propuesta es funcional, para tener la posibilidad de mejorarla y corregir sus fases, entre los muchos aspectos que esto comprende (Hernández Ortiz, 2007). La presente propuesta proporciona una directriz metodológica para poder generar propuestas de metodologías nuevas o reestructurar metodologías existentes.

Finalmente, toda propuesta de metodología para desarrollo de software solo podrá ser depurada y ajustada aplicándola a proyectos tanto pequeños como grandes, en diversos proyectos de software educativo. Las metodologías en el ámbito de la Ingeniería de Software son principalmente, producto de su aplicación y prueba en proyectos reales, para determinar su efectividad y pertinencia, todo desde el punto de vista práctico. En definitiva, las metodologías son generadas por contextos teóricos, necesidades específicas y el sentido común del desarrollo, y son ajustadas y depuradas en la práctica.

## REFERENCIAS Y BIBLIOGRAFÍA

- Alcantara Olguin, M. Y. (2008). *“Nociones de Educación virtual y diseño de software instruccional”*. Monografía de licenciatura. Escuela Superior Tlahuelilpan. Universidad Autónoma del Estado de Hidalgo. México. pp. 49, 50 y 64.
- Batini, C. et al. (1981). *“Le basi di dati: principi, tecnologie e metodologie”*. Suplemento de Sistemi e Automazione No. 216. Italia.
- Cova, A.; Arrieta, X.; Riveros, V. (2008). *“Análisis y comparación de diversos modelos de evaluación de software educativo”*. Enl@ce: Revista Venezolana de Información, Tecnología y Conocimiento. Venezuela. (URL: [es.wikibooks.org/wiki/El\\_software\\_educativo\\_en\\_la\\_clase.\\_Intruso\\_o\\_...](http://es.wikibooks.org/wiki/El_software_educativo_en_la_clase._Intruso_o_...)). Fecha de consulta: marzo de 2009.
- De Miguel Castaño, A; Piattini Velthuis, M. G. (1993). *“Concepción y diseño de bases de datos. Del modelo E/R al modelo relacional”*. Editorial Addison-Wesley Iberoamericana. Primera edición. U.S.A. pp: 695, 709 a 712.
- Fernández Muñoz, R. (1998). *“Las nuevas tecnologías aplicadas a la educación”*. Editorial Paidós.
- González Castañón, M. A. (2009). *“Evaluación del software educativo: orientaciones para su uso pedagógico”*. (URL: <http://www.tecnoedu.net/lecturas/materiales/lectura27.pdf>). Universidad EAFIT. Colombia. Fecha de consulta: marzo de 2009.
- Hernández Ortiz, R. (2007). *“Propuesta de metodología para el desarrollo de páginas y sitios web”*. Tesis de licenciatura. Universidad Autónoma del Estado de Hidalgo. México. pp.: 64 a 72.
- Olivé, A. (1985). *“Modelización conceptual de sistemas de información”*. Apuntes de la VII Escuela de Verano de Informática.
- Panqueva, G. (2001). *“Ingeniería de software educativo”*. Editorial Mc Graw-Hill.
- Pesce, M. (1996). *“VRML para internet”*. Editorial Prentice Hall. México.

Pressman, R. E. (2006). *“Ingeniería de Software”*. Editorial Mc Graw-Hill. Sexta edición. México. pp: 641 a 649 y 690 a 695.

R. Hilera, J. (2008). *“Aplicación de la realidad virtual en la enseñanza a través de internet”*. (URL: <http://www.ucm.es/info/multidoc/multidoc/revista/num8/hilera-oton.html#resumen>).

Tello Fernández, H. (2006). *“Metodología participativa para la intervención de proyectos”*. Informe técnico: versión preliminar en edición. Biodamaz IIAP. Quito, Perú.

Torres Yañez, C. (2008). *“Aspectos del diseño instruccional”*. (URL: <http://www.mse.buap.mx/recursos/disenoinstruccional/disenoinstruccional.html>).

Ullman, J. D. (1990). *“Principles of Database and Knowledge Base Systems”*. Vol I. EEUU. Computer Science Press.

Vaughan, T. (1994). *“Todo el poder de la multimedia”*. Segunda edición. Editorial Mc Graw-Hill. México.

Yokohama, I. U. (2008). *“Breve historia de la educación”*. (URL: <http://www.monografias.com/trabajos18/historia-de-educacion/historia-de-educacion.shtml#modelohistoriadelaeducacion>).