

SINOPSIS DE METODOLOGIAS Y MODELOS DE SOFTWARE EDUCATIVO

Donaldo José Dueñas Bernal

donaldographicd@gmail.com

Adán Alberto Gómez Salgado

aagomez@correo.unicordoba.edu.co

Raúl Emiro Toscano Miranda

rtoscano@correo.unicordoba.edu.co

Manuel Fernando Caro Piñeres

manuelcaro@correo.unicordoba.edu.co

RESUMEN

El presente artículo detalla los distintos tipos de metodologías y modelos de software educativo utilizados en las organizaciones e instituciones educativas a nivel mundial, realizando un análisis comparativo detallado en donde se describe paso a paso cual es la necesidad educativa, las fases utilizadas para su elaboración, las características del sistema y el diseño educativo para la creación de software educativo, para fortalecer los procesos de enseñanza aprendizaje en los usuarios (estudiantes).

Key words: Educational Software, Methodology, Monitoring, Pedagogy, Documentation, education, computing, e-learning.

ABSTRACT

This article details the different types of methodologies and models of educational software used in educational organizations and institutions worldwide, performing a detailed analysis that describes step by step what is the educational need, the phases used for its preparation, the characteristics of the system and the educational design for the creation of educational software, to strengthen the teaching-learning processes in the users (students).

I. INTRODUCCIÓN

En la actualidad, a nivel mundial existen diversas metodologías y modelos de software educativo cuyo objetivo es apoyar los procesos de enseñanza-aprendizaje en los estudiantes. Estas metodologías intentan ver la manera de cómo poder aprovechar las habilidades y técnicas de los estudiantes y de cómo poder mejorar el proceso de enseñanza tradicional. Es por esto, que diversos autores plantean soluciones que han sido investigadas y analizadas muy a fondo sobre las necesidades que se presentan en las instituciones, para así poder dar una solución a estas falencias que se presentan. Por otro lado, se diseñaron prototipos, proyectos didácticos y lúdicos previamente estudiados y

analizados para poder ser implementados en las instituciones como herramienta de apoyo.

Así pues, el estudio de estas metodologías demuestran las estrategias y dinámicas para la realización de estas, es por esto que necesita conocer los aspectos que los caracterizan para la creación de un sistema de software, también se debe conocer cuáles son las etapas o fases que utilizan para la realización del prototipo o sistema basándose en las necesidades de los estudiantes o usuarios, desde esta perspectiva el estudio de estas metodologías nos permite indagar sobre los problemas que se encuentran inmersos en las comunidades estudiantes y organizacionales.

II. SINOPSIS DE METODOLOGIAS Y MODELO DE SOFTWARE EDUCATIVO

A continuación, se realizará un análisis de las distintas metodologías y modelos de software utilizadas en el campo de la educación que se han desarrollado desde 1978, hasta el año 2009.

1. MODESEC, es un Modelo de Diseño de Software Educativo Basado en Competencias, el cual nos muestra una visión de cómo desarrollar nuevas aplicaciones mediante la combinación de componentes pedagógicos, didácticos, multimediales y de ingeniería de software. El modelo está compuesto por cinco fases que detallan paso a paso los aspectos que se deben tener en cuenta para la creación de software educativo. La fase inicial constituye la descripción del diseño educativo, en la cual se analiza la necesidad educativa, se plantean los objetivos de aprendizaje y se representan los objetivos que se procuran desarrollar con la creación la aplicación, por otro lado, el diseño computacional y el diseño multimedial, están encargadas del análisis y diseño del software. En la fase de producción se enlazan todas las muestras recolectados, y por último la fase de aplicación, donde se hacen las pruebas de campo para evaluar el funcionamiento de la aplicación en la población que ha sido aplicada (Caro & Toscano, 2012).

2. En la metodología de Desarrollo De Software

Educativo: una propuesta metodológica se propone una metodología para el diseño de software en la educación, la cual reúne tanto los elementos educativos como los de análisis y diseño de sistemas, integrando las etapas de realización de software con las teorías de aprendizaje. Este tipo de metodología de software educativo se ha puesto en práctica en diversas áreas de la educación, las cuales se ha alcanzado una gran aceptación por parte de los usuarios, quienes solicitan el material para ponerlo en práctica en sus aulas, a un 81,2% le pareció fácil la utilización del mismo, el 84% se sintió cómodo con la utilización del software. Todo material educativo es necesario someterlo a una validación para verificar su funcionamiento y aceptación en los usuarios, para así poder lograr un material de óptima calidad. (Quintero H., Portillo L., Luque F., 2005).

3. Programación Extrema XP: Es un enfoque de la ingeniería de software formulado por Kent Beck. Es una de las llamadas Metodologías ágiles de desarrollo de software más exitosas de los tiempos recientes, nace como nueva disciplina de desarrollo de software. Está compuesta por un ciclo de vida que comprende seis fases para su desarrollo: la fase inicial es la de exploración, donde los clientes planean a grandes rasgos las historias de usuarios que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. La segunda fase se encarga de planificar la entrega del producto, aquí es donde se colocan de acuerdo los programadores y desarrolladores del software sobre qué actividades, contenidos y competencias se van a implementar en el producto, por otro lado, la fase de iteración detalla la posible entrega del sistema. El plan de entrega está compuesto por iteraciones de no más de tres semanas. Seguidamente la fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. En la fase de mantenimiento el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. Para culminar con la fase de muerte del proyecto, es cuando el cliente no tiene más historias para ser incluidas en el sistema. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo.

4. Dynamic Systems Development Method (DSDM) El Método de Desarrollo de Sistemas dinámico (DSDM) es una metodología de desarrollo de software

originalmente basado en la metodología de Desarrollo de Aplicación Rápida. Este tipo de metodología presenta una serie de fases que van en secuencia unas a otras, la primera fase es la de pre-proyecto donde se identifican los proyectos elegidos y se definen los presupuestos, la segunda fase es la del ciclo de vida que a su vez está dividida en subetapas, las cuales se genera el análisis, una vez concluido el análisis se desarrolla el sistema ya sea de forma iterativa o incremental, estas subfases están compuestas por una serie de actividades, por último encontramos la fase de post-proyecto que es la que se asegura de la eficiencia y eficacia del proyecto esto se lleva a cabo del mantenimiento, mejoras y ajuste del sistema DSDM El ciclo de vida no suele terminarse de una vez, sino que se vuelve a pasar por fases anteriores para un mayor rendimiento.

5. La técnica de Adaptive software Development fue desarrollada por Jim Highsmith y Sam Bayer. Esta metodología se adapta al cambio en lugar de luchar contra él. Se basa en la adaptación continua a circunstancias cambiantes. Es un modelo de implementación de patrones ágiles para desarrollo de software. Al igual que otras metodologías ágiles, su funcionamiento es cíclico y reconoce que en cada iteración se producirán cambios e incluso errores. En ella no hay un ciclo de planificación-diseño-construcción del software, sino un ciclo especular colaborar-aprender. Utilizando este tipo de metodología se pueden alcanzar excelentes resultados, pero debido a las características que maneja es más factible usarla para proyectos pequeños y medianos, para adquirir práctica y experiencia para así poder llegar a alcanzar productos de alta calidad.

6. Metodología Dinámica para el Desarrollo de Software Educativo En este nuevo paradigma educativo, el desarrollo de materiales computarizados es complejo, deben efectuarse decisiones en torno a los contenidos (selección, organización, adaptación a los usuarios), a las estrategias de enseñanza de dichos contenidos y a la forma de presentación más adecuadas con el objeto de facilitar el proceso de aprendizaje del usuario. Para esta metodología no se requiere culminación de una fase para pasar a la otra, lo que tiene como ventaja que es posible obtener rápidamente un prototipo que permita hacer validaciones parciales y correcciones de ser necesario.

7. Modelo THALES de Desarrollo de Software Educativo. El modelo de diseño y creación de Software Educativo seleccionado es el denominado Metodología THALES, propuesto por Madueño (2003), esta metodología representa un modelo híbrido no lineal, lo cual involucra que las fases de desarrollo se presentan

por medio de ciclos, dada la dificultad para separar las tareas a ejecutar, habiendo revisiones continuas del proceso y del producto, en concordancia con los resultados que se van obteniendo. Esta metodología comprende seis fases para su realización, las cuales son: la primera fase es la de planeación, es donde se plantea una idea inicial beneficiosa para el proceso de enseñanza aprendizaje, la segunda fase consta en diseñar el producto final y los componentes educativos que permiten lograr los objetivos educacionales, la tercera fase es la de construcción del software mediante la materialización del boceto creado en la fase previa, la fase cuatro es la prueba piloto que es donde se busca darle toques finales al prototipo a partir del uso de varios usuarios, utilizado como una evaluación preliminarmente al software, la quinta fase es la de evaluación donde se corroboran los objetivos propuestos y si han sido alcanzados por el software y por último la fase seis encargada del mejoramiento del software ya sea integrando actualizaciones, versiones nuevas y módulos a desarrollar.

8. Ingeniería de Software Educativo con Modelaje Orientado por Objetos: un Medio para Desarrollar Micromundos Interactivos Las metodologías convencionales de Ingeniería de Software Educativo (ISE) tienen mecanismos vigorosos para hacer un análisis de necesidades y diseño educativo completos, pero poco han evolucionado con la tecnología en lo relacionado con el diseño computacional. Esta metodología consta de 5 fases para su desarrollo, la primera fase se encarga de determinar el contexto en el cual se va a crear la aplicación y derivar de allí los requerimientos que deberá atender la solución interactiva, como complemento a otras soluciones basadas en uso de otros medios (personales, impresos, audio-visuales, experienciales), teniendo claro el rol de cada uno de los medios educativos seleccionados y la viabilidad de usarlos, la segunda fase es el diseño del Micromundo Interactivo donde se realiza a tres niveles diferentes: educativo, comunicacional y computacional, la fase tres es la de desarrollo, en esta fase se implementa la aplicación usando toda la información obtenida anteriormente, por último la fase cinco y seis permite ir depurando los componentes del modelo generado, haciendo validación con expertos de los prototipos durante la etapa de diseño y prueba uno a uno de los módulos desarrollados, a medida que estos están funcionales.

9. Metodología MERISE Es una metodología de Análisis, de concepción y de gestión completamente integrada al proyecto, contemplando con ello el desarrollo de los sistemas de información. nace como

proyecto para cubrir las necesidades tanto de la administración como también todo lo que involucra las empresas desde los años 90, debido a los cambios sucesivos del medio ambiente informático, esta metodología presenta cuatro fases para su elaboración, la primera fase es la hacer un estudio preliminar la cual comprende la planificación del sistema a desarrollar, la segunda fase está encargada de realizar un estudio detallado partiendo de la necesidad de los clientes o usuarios y para evaluar que conceptos tiene el cliente sobre el sistema, la fase tres es la de implementar, por la cual su objetivo es producir una solución eficiente para poner en marcha el producto, la fase cuatro es la puesta en marcha es la que se encarga de instalar el sistema de software desarrollado entorno al cliente y también se necesita realizar pruebas de adaptación necesarias.

10. FEATURE-DRIVEN DEVELOPMENT (FDD) (Desarrollo basado en funciones) es el número uno a nivel de las metodologías ágiles de desarrollo y forma parte de Alianza Ágil fue diseñado para trabajar con otras actividades de desarrollo de software y no requiere la utilización de ningún modelo de proceso específico. Esta metodología plantea cinco fases secuenciales para su realización, las cuales el diseño y la construcción del sistema se llevan a cabo. La primera fase es desarrollar un modelo global donde se cuente con una idea del contexto y los requerimientos del sistema, la segunda fase se enfatiza en la construcción de las características y requerimientos del sistema para que se obtenga una buena funcionalidad del sistema, la tercera fase se encarga de planificar unos listados de alto nivel, las cuales serán la prioridad y de esta dependerá el sistema, y por último la fase cuatro y cinco se encargan de diseñar y construir el sistema y que logre una funcionalidad esperada inspeccionando de forma detallada el producto y realizar prueba con los usuarios para determinar funcionalidad del sistema.

11. Modelo sistémico de calidad (MOSCA) del software este modelo se basa principalmente en el dominio de teorías para el diseño de ambientes educativos computarizados. también se enfoca en analizar qué problemas o situaciones problemáticas existen, sus causas y posibles soluciones, para entonces sí determinar cuáles de éstas últimas son aplicables y pueden generar los mejores resultados. Su primera fase contempla la calidad del producto de software enfocado hacia un modelo sistémico, la segunda fase se encarga de detallar los procesos de como se ha desarrollado el producto y de cómo ponerlo a prueba y la tercera fase busca generar una integración de las mediciones de los sub-modelos de la calidad del producto y alcanzar una calidad en el proceso. Este tipo de metodología puede

ser aplicada en las áreas de telecomunicaciones, desarrollo web, finanzas, tecnologías de la información y en el área de software administrativo.

12. Metodología del software educativo por Álvaro Galvis. esta metodología investiga el dominio de teorías para el diseño de ambientes educativos computarizados. también se enfoca en analizar qué problemas o situaciones problemáticas existen, sus causas y posibles soluciones, para entonces sí determinar cuáles de éstas últimas son aplicables y pueden generar los mejores resultados. contempla una serie de fases o etapas de un proceso sistemático, la primera fase es la de análisis esta etapa se encarga de determinar el contexto en el cual se va a crear la aplicación y atender una solución interactiva para mejorar los procesos de enseñanza, la segunda fase tiene como función resolver los interrogantes que se refieren al alcance, contenido y tratamiento que debe ser capaz de apoyar el Sistema Educativo, la fase tres es donde se implementa la aplicación usando la información obtenida anteriormente, por último las fases de prueba de campo y de implementación consiste en aplicar el sistema en la vida real y verificar los alcances del sistema

III. CONCLUSIONES

El proceso de desarrollo del software educativo ha sido abordado por diferentes autores que aportan metodologías y modelos con vistas a su estructuración y organización. Todas estas técnicas desarrolladas por ellos se generan con el fin de fortalecer los procesos de enseñanza y aprendizaje en los estudiantes, es por esto que se diseñan y se implementan diferentes estrategias didácticas, lúdicas y visuales con el fin de favorecer la efectividad y eficacia en los procesos de enseñanza, estas estrategias se utilizan como apoyo a los docentes en el aula. Para esto es necesario que se realicen pruebas de campo con los prototipos y sistemas creados para medir el impacto de estos.

Existe una innumerable cantidad de metodologías de desarrollo de software educativo propuestas, algunas quedan en el olvido y algunos otras en el proceso de adecuarse a los cambios de enfoques, a la innumerable cantidad de situaciones que deben enfrentar y a las diversas áreas de aplicación en que deben desenvolverse; conforme avanza su desarrollo, se han incorporado reglas, notaciones, prácticas y documentos que requieren mucha disposición y tiempo para poder seguirlos correctamente.

COMPARACIÓN DE LAS METODOLOGÍAS Y MODELOS DE SOFTWARE EDUCATIVO		
METODOLOGÍA, MODELO DE "SOFTWARE EDUCATIVO"	CARACTERÍSTICAS	AREAS
MERISE	Es aplicado para cubrir las necesidades tanto de administración como también de lo que involucran las empresas como los niveles conceptuales, organizaciones y operacionales Es una Metodología de desarrollo de Software utilizado para el desarrollo de sistemas de información en instituciones o empresas Merise puede ser utilizado en el desarrollo de todo tipo de sistema de información.	Educación
Adaptive Software Development (Asd)	Es un modelo de implementación de patrones ágiles para desarrollo de software. es utilizada para obtener excelentes resultados, pero debido a las características que maneja es más factible usarla para proyectos pequeños y medianos, para adquirir práctica y experiencia	Ingeniería de sistemas
Metodología del software educativo por Álvaro Galvis.	tiene como objetivo principal el dominio de teorías para el diseño de ambientes educativos computarizados (MEC).	áreas de educación
Dynamic Systems Development Method (DSDM)	generar y proporcionar métodos de desarrollo rápido de aplicaciones con el fin de ofrecer gestión y entrega de soluciones a las necesidades de la población. disminuir las falencias y proporcionar un incremento de las técnicas de las personas para realizar proyectos rápidamente.	Gestión de proyectos.
Feature-Driven Development (FDD).	Su característica principal se centra en el enfoque ágil para el desarrollo de sistemas es una metodología ágil diseñada para el desarrollo de software de alta calidad. no requiere la utilización de ningún modelo de proceso específico.	Ingeniería de sistemas.
Programación Extrema XP.	es un enfoque de la ingeniería de software diseñado para establecer buenas prácticas en el desarrollo de proyectos es una metodología de programación organizada ya que ayuda a disminuir los rangos de errores en los proyectos.	Ingeniería de sistema, programación y áreas de educación superior
Ingeniería de Software Educativo con Modelaje Orientado por Objetos: Un Medio Para Desarrollar Micromundos Interactivos	Es una Ingeniería de software educativo orientada a objetos Permite crear ambientes basados en micromundos interactivos Es utilizado para el desarrollo de software de calidad.	Informática y áreas de educación
reModelo Sistémico de Calidad (MOSCA) del Software	tiene como objetivo mostrar la formulación y aplicación de un algoritmo que permite la operacionalización del Modelo Sistémico de Calidad (MOSCA) para la Estimación de la Calidad de los Sistemas de Software este modelo se basa principalmente en el dominio de teorías para el diseño de ambientes educativos computarizados.	Áreas De Educación
"Metodología THALES"	es un modelo de desarrollo de software educativo utilizado como apoyo al docente para la integración de la tecnología en el currículo sus fases de desarrollo son cíclicas, dada la dificultad para separar las tareas a ejecutar ya que se hacen revisiones constantes de los procesos y el producto final.	Áreas De Educación
Metodología Dinámica para el Desarrollo de Software Educativo	es un paradigma educativo que contempla la utilización de las nuevas tecnologías en las aulas de clases. el computador es utilizado como medio dinámico que permite impartir los procesos educativos.	Áreas De Educación
La metodología de Desarrollo de Software Educativo: una propuesta metodológica	su objetivo principal es lograr el uso eficiente de las tecnologías de la informática y las comunicaciones en el proceso enseñanza aprendizaje	Áreas De Educación
MODESEC	tiene como base fundamental el sistema de competencias para el desarrollo de software integra los componentes pedagógicos, didácticos, multimediales y de ingeniería de software en su creación de software. Es de fácil uso para implementar cualquier tipo de software educativo en las instituciones educativas donde se presente algún tipo de necesidad educativa.	Áreas De Educación

La diferencia más importante entre las metodologías está en las necesidades reales de los estudiantes ya que cada organización escolar debe manejar su propia información y buscar una posible solución a estas necesidades, todas las metodologías cubren

necesidades, es por esto, por lo que se desarrollan sistemas complejos para facilitar el aprendizaje y suplir las falencias en la educación para alcanzar los procesos de enseñanza-aprendizaje.

Cabe resaltar que las metodologías: Adaptive Software Development (ASD), Dynamic Systems Development Method (DSDM), Programación Extrema XP y Feature-Driven Development (FDD), son metodologías que propiamente no son desarrolladas para solucionar problemas de carácter educativo, pero tienen un enfoque asociado a las áreas de educación, ya que todas estas metodologías tiene como fin cubrir las necesidades que se presentan en los usuarios, de este modo estas metodologías desarrollan estrategias y se asignan actividades para la utilización eficaz, para así poder generar un impacto a la hora de ser implementado.

Del mismo modo, las metodologías, MODESEC, MOSCA, Thales y la Metodología Dinámica, son modelos analizados que cuentan con grandes ventajas como por ejemplo la facilidad en el diseño de sus contenidos para establecer las estrategias y las competencias a alcanzar. Por otro lado, su gran ventaja es el diseño pedagógico el cual basa en las necesidades educativas de los estudiantes y se adaptan a las características que presentan ellos.

Para concluir, MODESEC es un modelo que se convierte en una guía de fácil uso para implementar software educativo conforme a las necesidades de alguna institución educativa, porque deben intervenir personas actas y apropiadas en el área del problema, con expertos en pedagogía y didáctica, y con docentes expertos en informática. Este modelo presenta ventajas, ya que es el único modelo que se basa en el diseño de competencias y su metodología se ve reflejada en software educativo.

IV. REFERENCIAS

- Arias M., López A., (12 de junio de 2013), Honmy R. Metodología Dinámica para el Desarrollo de Software Educativo.
- Caro & Toscano. (2012). MODESEC: Modelo para el desarrollo de software educativo basado en competencias. Retrieved from www.edupmedia.org
- Caro, Toscano, Hernandez y David. (2012). MODESEC: Modelo para el desarrollo de software educativo basado en competencias. Recuperado de www.edupmedia.org/Modesecc.
- Figuroa, P. (1997). Metodología de desarrollo de software Orientado por Objetos. <http://agamenon.uniandes.edu.co:8088/~pfigueroa/soo/metod/>.
- Galvis, A.H (1992). Ingeniería de Software Educativo. Santafé de Bogotá: Ediciones Uniandes.
- Galvis, A.H (1.997). Micromundos lúdicos interactivos: Aspectos críticos en su diseño y desarrollo. Informática Educativa.
- Madueño, L. (2003). Modelo THALES de desarrollo de software educativo. Universidad de Zulia. Venezuela. Recuperado de [http://modelosdesarrollomdc.blogspot.com/search/label/THALES 11](http://modelosdesarrollomdc.blogspot.com/search/label/THALES%2011)
- Madueño, L. (noviembre de 1997). Diseño de un Sistema Automatizado para la Administración de la Evaluación de los Aprendizajes a Nivel Superior. Maracaibo-Venezuela. Universidad del Zulia
- Madueño, L. (24 de noviembre de 2003). Desarrollo de Software Educativo bajo Plataforma Web. Recupero de www.ucv.ve/edutec/Ponencias/12.doc.
- Marqués, Pere. (junio de 1993) El software educativo. Una metodología para su diseño y desarrollo. Tesis Doctoral. Departamento de Didáctica y Organización Escolar. Universidad de Barcelona. España.
- Marqués P. (junio de 1995). Metodología Para La Elaboración De Software Educativo. Recuperado de <http://blues.uab.es/home/material/programes/t023151/uabdisof.htm>.
- Matt Stephens, Doug Rosenberg, (1 enero 1970). Extreme Programming Refactored: The Case Against XP.
- Mendoza, Pérez, Grimán, (septiembre 2004). Prototipo de modelo sistémico de calidad (MOSCA) del software. Universidad Simón Bolívar. Caracas Venezuela.
- Molina, R. (junio de 2006). Modelo de desarrollo de software educativo.
- Omez, R., Galvis, Mariño, (enero de 2000). Ingeniería de software educativo y modelaje orientado a objetos. Un medio para desarrollar micro mundos interactivos. Universidad de Los Andes. Bogotá. Colombia.
- O'Reilly & Associates, (junio de 2003). Extreme Programming, recuperado de <http://www.extremeprogramming.org>.
- Quintero H., Portillo L., Luque F., G. M. (2005). Desarrollo de software educativo: una propuesta metodológica.
- Tuffs, Stapleton, West, Eason Inter-operability of DSDM with the Rational Unified Process, DSDM Consortium, Issue 1, p. 1-29 (1999)

Para citar este artículo / to cite this article:

Dueñas, D., Gómez, A., Caro, M. y Toscano, R. (2017). Sinopsis de metodologías y modelos de software educativo. Acta Scientiæ Informaticæ 1(1), 70-74.