Modelo del Negocio para Análisis del Dominio del Software Educativo: un enfoque centrado en la calidad del producto

Francisca Losavio
Facultad de Ciencias, UCV
francislosavio@gmail.com

Yuly Esteves
Instituto Pedagógico de Miranda, UPEL
yulyesteves@gmail.com

RESUMEN

En el diseño de Software Educativo (SE) deben conciliarse los aspectos tecnológicos y pedagógicos relevantes para el dominio; la aplicación de principios inherentes a la Ingeniería del Software en el proceso de desarrollo, garantizaría la obtención de un SE de calidad según ISO/IEC 25010. Este estudio de desarrollo teórico incluye el Modelo del Negocio, que recoge las necesidades del dominio, como paso inicial al Proceso Extendido de Chung, de Losavio, Matteo y Pacilli (2014). La propuesta es que la entrada de la etapa de Análisis del Dominio incorpore las reglas, políticas y restricciones del negocio que contemplan los elementos pedagógicos considerados por el desarrollador. Entre los resultados obtenidos se encuentran la categorización de tipos de SE incluyendo su arquitectura y el modelo de calidad adaptado al dominio del SE, lo que puede ser utilizado posteriormente para la especificación de una arquitectura de referencia para el dominio.

Palabras clave: Modelo del Negocio; Análisis del Dominio; Software Educativo; Calidad del Software; Proceso Extendido de Chung.

Recibido: Abril 2015
Aceptado: Julio 2015

Business Model to Analyze the Educational Software Domain: a Standpoint Focused in the Quality of the Product

ABSTRACT

In the educational software design (ES), relevant aspects such as technology and pedagogy should be reconciled to domain. The application of inherent principles to software engineering in the development process will ensure to obtain an ES with quality according to ISO/IEC25010. This research of theoretical development includes a business model which collect domain needs, as initial step to the extended process of Chung, de Losavio, Matteo and Pacilli (2014). The propose is that the entry of domain analysis stage incorporate business rules, policies and restrictions of the business taking in to account pedagogical elements considered by the developer. As result, we have types of ES categorization that include its architecture and the quality model adapted to the ES domain, and it could be used as a reference of architecture to the domain.

Key Word: Business Model, Domain Analysis, Educational Software, Software Quality, Extended Process of Chung.

Le modèle d'affaire pour l'analyse du domaine du didacticiel : une approche centré sur la qualité du produit

RÉSUMÉ

Le design Didacticiel pour l'enseignement assisté par ordinateur doit concilier des aspects technologiques et pédagogiques pertinents pour le domaine; l'application de principes inhérents à l'Ingénierie du Logiciel dans le processus de développement garantirait l'obtention de un Didacticiel de qualité selon l'ISO/IEC 25010. Cette étude de développement théorique inclut le Modèle d'Affaire, qui contient les besoins du domaine, comme une première étape du Processus Étendu de Chung (Proceso Extendido de Chung) de Losavio, Matteo et Pacilli (2014). Ils proposent que la première étape pour l'Analyse du Domaine incorpore les règles, les politiques et les restrictions de l'affaire qui contemplent les éléments pédagogiques considérés par le développeur. Parmi les résultats obtenus nous avons la catégorisation de types de Didacticiel en incluant son architecture et le modèle de qualité adapté au domaine du Didacticiel, ce qui peut être utilisé ensuite pour la spécification d'une architecture de référence pour le domaine.

Mots-clés : modèle d'affaire**,** analyse du domaine, didacticiel, qualité du logiciel, processus étendu de Chung

Introducción

El diseño y la selección de Software Educativo (SE) constituyen un tema de investigación abierto, especialmente al considerar el creciente interés de los educadores por el uso de tecnologías informáticas en sus prácticas docentes. Existen muchas acepciones de SE, en este estudio se tomará aquella que considera al SE como una aplicación electrónica especialmente diseñada para mediar el proceso de enseñanza-aprendizaje. En este sentido, la adecuación del SE a los requisitos del usuario final (profesores, estudiantes) merece especial atención. El problema de traducir los requisitos del usuario en funcionalidades del sistema ha conducido a la búsqueda de soluciones en el proceso de la captura de requisitos. Este proceso involucra la captura, el modelado y el análisis de los requisitos, tanto del diseñador como del usuario final, así como de las reglas propias del *negocio educativo*, dadas muchas veces por las teorías pedagógicas subyacentes al diseño del software. En Ingeniería del Software, todo lo concerniente a captura y especificación de requisitos de un sistema de software se realiza en la denominada etapa de Análisis, una de las primeras etapas dentro del ciclo de vida del desarrollo del software.

Ahora bien, en nuestro contexto donde debemos caracterizar y generalizar arquitecturas de familias de sistemas, esta etapa se denomina Análisis del Domino (AD), dentro de la disciplina denominada Ingeniería del Dominio (Krueger, 2002; Pohl, Böckle y van der Linden, 2005), y los SE forman parte del dominio que se tratará. En esta etapa se comienza a definir la arquitectura de un sistema de software, o conjunto de componentes y conectores que ofrecen un cierto comportamiento (Garlan y Shaw, 1994), la cual es en gran parte responsable del comportamiento y de la calidad global de todo el sistema. Las actuales tendencias en la Ingeniería del Dominio son de incorporar un Modelo de Negocio (MN), el cual permite expresar procesos organizacionales en términos de objetos, comportamientos y objetivos o metas, capturando todos los aspectos relativos al (los) sistema (s) de software que soportará (n) el desarrollo del negocio. Del MN se derivarán los requisitos globales del sistema de software relativos a un proceso de negocio.

El objetivo de este trabajo es por lo tanto incorporar un primer paso de modelado del negocio al *Proceso Extendido de Chung (PEC)*, adaptado en Losavio, Matteo y Pacilli (2014)de Supakkul y Chung (2005), con un paso de AD. PEC ofrece un importante aporte pues propone la vinculación de los *requisitos funcionales (RF)* con los *no funcionales (RNF)*, también denominados *requisitos de calidad* (ISO/IEC 25010, 2011), integrados en el desarrollo de software centrado en la arquitectura, ya que el sistema se articula sobre esta estructura de base (del inglés "*baseline architecture*"). La norma ISO/

IEC 25010 (2011) ofrece una terminología unificada sobre la calidad del producto de software y es la que se utilizará en este trabajo para especificar los requisitos no funcionales (RNF), generalmente asociados con requisitos de calidad. Esta norma internacional define las características que describen la calidad de un producto de software mediante un modelo de estructura jerárquica. Se consideran ocho *características de calidad* de alto nivel de abstracción, generalmente no medibles, que representan la calidad del producto de software desde el punto de vista interno (durante el proceso de desarrollo) y externo (en un ambiente de prueba) del software: *funcionalidad* (adecuación del software con las funcionalidades requeridas, es decir que realmente cumple con los requisitos funcionales exigidos), *compatibilidad*, *seguridad, confiabilidad, eficiencia, facilidad de mantenimiento, facilidad de uso* y *portabilidad*. Estas características son refinadas en sub-características; por ejemplo en compatibilidad se tienen las sub-características coexistencia e interoperabilidad; el refinamiento puede abarcar varios niveles de sub-sub-características, hasta llegar a los *atributos de calidad*, que indican como deben ser medidas por métricas específicas, las sub-características y por ende la característica. El modelo de calidad puede utilizarse para la especificación (cualitativa y/o cuantitativa) de las características de calidad relacionadas con RF y RNF de un producto de software.

Por lo anterior, el análisis de requisitos se considera crucial para el diseño o selección de la arquitectura sobre la cual se articulará el software, pues los requisitos serán implementados en las operaciones que se realizan o requisitos funcionales (RF) o podrían responder a características de calidad (RNF) exigidas por los RF y serán implementados como "funcionalidades implícitas" (ISO/IEC 25010, 2011). En esta etapa se consideran taxonomías, como son, entre otras, las categorías de RF, RNF y las reglas del negocio, que pueden ser capturadas a partir del MN, el cual proporciona maneras de expresar procesos organizacionales en términos de objetos, comportamientos y metas, capturando todos los aspectos relativos al sistema el cual soportará el desarrollo del negocio. Si los modelos del negocio no se producen, se corre el riesgo que los desarrolladores no presten suficiente atención a la manera cómo se hace el negocio, creando inconsistencias (Arias, 2005). En este estudio se propone entonces agregar al PEC de Losavio, Matteo y Pacilli (2014) una primera etapa adicional de modelado del negocio, como entrada al proceso de AD, obteniendo así los elementos necesarios que permitan completar PEC con la etapa de construcción de una arquitectura genérica que traiga inmersa las características de calidad y los estilos arquitectónicos propios del dominio del SE.

Para tal fin, este artículo se estructura de la siguiente manera, además de la introducción y la conclusión: en la segunda sección se discute en torno a los fundamentos teóricos que soportan la investigación, tales como Ingeniería del Software, Calidad del Software, Modelo del Negocio y Análisis del Dominio. En la tercera sección se presenta la propuesta del proceso que, partiendo del modelo de negocio, considera el análisis del dominio para el aseguramiento de la calidad. En la cuarta sección se aplica este proceso a un caso de estudio para ilustrar el modelado del negocio en ese dominio, por lo que se presentan brevemente consideraciones en torno al SE.

Contexto y terminología

En esta sección se presentan los fundamentos básicos que se toman en consideración en el desarrollo de la propuesta, especialmente las relacionadas con la Ingeniería del Software y los conceptos fundamentales.

Ingeniería del Software

En las últimas décadas, uno de los avances más significativos de la humanidad lo constituye indiscutiblemente la *tecnología de la información y la comunicación (TIC)*, y dentro de ella, sin duda alguna, el desarrollo de sistemas de software que cumplan con la calidad exigida por estas tecnologías. ¿Quién no tiene que ver con aplicaciones de software en la vida diaria? En el área educativa los encontramos desde las inscripciones estudiantiles en una universidad, hasta la mediación del aprendizaje a través de aplicaciones electrónicas. Así por el estilo, casi cualquier actividad educativa en el mundo está ligada con las aplicaciones de software.

Es una nueva cultura: la cultura de la información y en consecuencia un nuevo tipo de sociedad, donde cada día las fallas en sistemas de software afectan a un número mayor de usuarios, lo que ha ocasionado que las organizaciones requieran, para su desarrollo informático, la formación de talentos en el área de la Ingeniería del Software. Lo que busca la disciplina de la *Ingeniería de Software (IS)* (Pressman, 2001), en la cual se pretende pasar de técnicas artesanales a técnicas de ingeniería, es desarrollar una solución de software, sistema de software, aplicación o producto de calidad, generalmente complejo, que considere convenientemente las necesidades y recursos, así como los procesos y procedimientos organizacionales que pueden ser automatizados, pues, mediante un buen análisis se podrá llegar

a obtener una buena herramienta, reduciendo costos y aprovechando al máximo los recursos.

Para ello, la IS maneja cuatro pilares fundamentales: aseguramiento de la calidad, establecimiento de procesos, desarrollo de métodos y selección de herramientas que permitan la toma de decisiones adecuadas, basados en procedimientos previamente estudiados, repetibles y probados como métodos de ingeniería.

En el área educativa, sin embargo, se aprecia la poca vinculación existente entre los mencionados pilares de la IS y los procesos de desarrollo que se siguen, muchos de los cuales suelen realizarse sin tomar en cuenta todas las aristas de este complejo problema, pues se requiere conciliar procesos inherentes al área educativa, con aquellos suficientemente probados y recomendados en el área computacional. El resultado es la existencia de sistemas heterogéneos, generalmente realizados "*ad hoc*" y sin contemplar la calidad de producto. En el negocio educativo, los productos de software desarrollados deben satisfacer los requerimientos pedagógicos manifiestos en las bases curriculares de la institución para la cual se diseña, además de satisfacer, en primera instancia, las políticas educativas de su localidad.

En esta investigación, se parte de la siguiente premisa: el aseguramiento de la calidad de los SE puede lograrse partiendo del modelo de negocio, en el cual se identifican reglas de negocio específicas. Una *regla de negocio* se considera como una política o un reglamento que debe ser cumplido en el dominio del negocio; por ejemplo el uso de software libre para la selección o el desarrollo de sistemas de software en instituciones públicas de gobierno, o el uso de una plataforma específica como Linux/Unix, Mac o Windows. En este sentido, en el dominio de SE consideramos que una regla de negocio podría ser constituida por restricciones curriculares y consideraciones pedagógicas de alto nivel, que deben tenerse en cuenta en el desarrollo de un SE de calidad.

Calidad del software

Una de las grandes metas que se plantean los desarrolladores de software es la satisfacción del usuario, la cual depende en gran medida de la comunicación que se establece entre los diferentes especialistas vinculados en el proceso de desarrollo. La satisfacción del usuario, puede lograrse al incorporar estándares de calidad cuando se desarrollan sistemas de software, lo cual se enmarca en las "buenas prácticas" recomendadas por la IS. Un

estándar de calidad es un conjunto de lineamientos o normas diseñadas para que las aplicaciones de software puedan ser evaluadas al final del proceso y logren satisfactoriamente cada objetivo para el cual fueron desarrolladas, como se mencionó en la Introducción. Uno de los estándares más utilizados es la norma ISO/IEC 25010 (2011), que reemplaza y actualiza el estándar ISO 9126-1 (2006). Define, desde el punto de vista del usuario final:

> Un *modelo de calidad del producto* se compone de ocho características de alto nivel de abstracción, que se subdividen en sub-características, mencionadas anteriormente. Se refieren a las propiedades *inherentes* al software (las características o requisitos de calidad, las cuales no cambian aunque el software cambie); las propiedades *asignadas*, que pueden cambiar aunque el software no cambie, como costo y tiempo de introducción del software al mercado, son requisitos no funcionales que no son considerados parte del modelo de calidad. El modelo es aplicable a los productos de software y sistemas informáticos.
>
> Un modelo de calidad en uso se compone de cuatro características: productividad, efectividad, seguridad y satisfacción, algunas de las cuales se subdividen en subcaracterísticas. Se relacionan con el resultado de la interacción con los usuarios finales, cuando un sistema de software se emplea en un contexto particular de uso (p. 4).

Las características definidas por ambos modelos son relevantes para todos los productos de software y sistemas informáticos. Las características y sub-características de calidad del producto (ver Figura 1) proporcionan coherencia terminológica para especificar, medir y evaluar la calidad del producto de software y sistemas informáticos (Alfonzo y Mariño, 2013), ISO/IEC 25010 (2011).

En la Ingeniería de Software, existe gran cantidad de riesgos e imponderables que pueden atentar contra el desarrollo de soluciones de software realmente efectivas, aún con el uso adecuado de los estándares de calidad; sin embargo los requisitos de calidad ayudan a detectar esos riesgos; por ejemplo si se controla la tolerancia a fallas en un sistema, si puede disminuir ese tipo de riesgo. Esto pudiera estar relacionado con problemas en el proceso de desarrollo, o bien, con problemas al momento de realizar la determinación y análisis de requisitos; también pueden surgir problemas debido al poco conocimiento que se tenga del negocio para el cual se está elaborando el software. En este sentido, elaborar el modelo del negocio como punto de partida en el proceso de determinación y análisis de requisitos, puede contribuir a resolver, en parte, la detección y solución de conflictos durante el desarrollo.

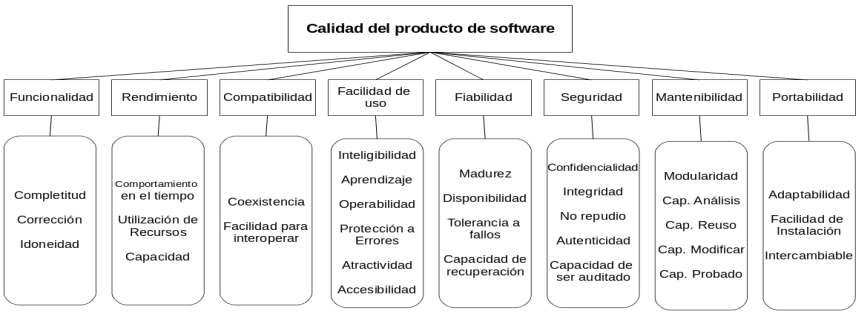

Figura 1. **Modelo de Calidad ISO/IEC 20510. Características con sus subcaracterísticas de calidad (ISO/IEC 25010, 2011)**

Modelo del negocio (MN)

El MN es una vista interna de la organización, sus procesos y sistemas, donde se aprecia el acoplamiento entre los diferentes modelos del sistema organizacional (Rodríguez, Fernández y Pattini, 2005). Estos modelos pueden ser representados utilizando UML 2.0 o alguna de sus extensiones o bien utilizando la propuesta del *Business Process Management Initiative* (BPMI; Delphi Group, 2011; Owen y Raj, 2005) con su notación *Business Process Modelling Notation (BPMN)* (BPMI, 2004). Las tendencias actuales del desarrollo de software incluyen el modelado del negocio como una etapa importante para una mejor captura y especificación de los requisitos que debe satisfacer el futuro sistema de software, el cual soportará total o parcialmente los procesos del negocio; también considerando las TICs. La tendencia actual es traducir los procesos de negocios a composición de "servicios" Web, los cuales ofrecen soluciones rápidas para la automatización de estos procesos. Sin embargo, la completitud y trazabilidad de los requisitos son cruciales para garantizar la flexibilidad del sistema para soportar cambios; esta propiedad es expresada por la característica de calidad denominada *mantenibilidad* en ISO/IEC 25010 (2011); de esta forma se podrán obtener productos capaces de mejorar en el tiempo. Las relaciones entre las entidades involucradas y los objetivos que los procesos deben satisfacer, tienen que ser especificados en el MN. Mediante una correspondencia adecuada entre los MN y el sistema, se incrementará la capacidad de evolución. Cuando haya una variación en el negocio, podrá ser reflejada rápidamente en el sistema, facilitando así su evolución.

Uno de los problemas del desarrollo de aplicaciones educativas es precisamente el no contar con un proceso de modelado del negocio educativo ni con los criterios que permitan la trazabilidad de este modelo con las funcionalidades del sistema. Aún no está claro cómo debe realizarse la elección de las funcionalidades o RF, la determinación de los RNF prioritarios y el diseño de un marco arquitectural o arquitectura de referencia válida para cualquier familia de aplicaciones o sistemas del dominio del SE; es allí donde se ponen de manifiesto los conflictos entre las propiedades de calidad relativas a cada requisito. Por lo tanto, en las primeras fases del desarrollo es muy importante poder especificar estos aspectos, para proporcionar una arquitectura de referencia acorde con los requisitos iniciales. Una arquitectura de referencia (AR) comprende la estructura arquitectónica "baseline", también denominada plantilla, en inglés "framework" o plataforma (Pohl, Böckle y van der Linden , 2005), en donde se articulan los componentes de todo el sistema y que incluye una parte variable o modelo de variabilidad, que es la que se instancia para derivar a partir de ella, sistemas o productos de software concretos; AR por lo tanto es instanciada para ser reutilizada en la derivación de sistemas que comparten una estructura base similar. Es de hacer notar que las aplicaciones actuales que se basan en la comunicación electrónica, deben responder a la evolución tecnológica creciente; por lo tanto su arquitectura debe reflejar esta flexibilidad para su evolución, es decir la capacidad de ser modificados o mantenibilidad en ISO/IEC 25010 (2011). Por lo tanto, la capacidad del software de ser evolutivo, que se adquiere mediante la trazabilidad entre el MN y el modelo del sistema de software, corrobora esta afirmación y se recalca que la especificación de RF y RNF es crucial para conseguir dicha trazabilidad.

Análisis de Dominio

Hay enfoques para el proceso de desarrollo que consideran relacionar los requisitos funcionales con los requisitos de calidad u objetivos de calidad a los cuales éstos responden, entre estas se encuentra el de Losavio, Matteo y Pacilli (2014), quienes definen el proceso PEC basado en Supakkul y Chung (2005), incluyendo en la etapa inicial el AD para la reutilización del conocimiento sobre la arquitectura y la identificación de restricciones o propiedades globales sobre las familias de aplicaciones del dominio (ver Figura 2); este AD incluye la especificación de las propiedades de calidad derivadas de requisitos funcionales, arquitecturales y otras restricciones, mediante el estándar ISO/IEC 25010 (2011). Un *dominio*, según Bérard (1992), es el conjunto mínimo de propiedades que definen una familia de problemas

para los cuales se requieren soluciones computacionales. El resultado de este AD es el punto de partida para la justificación de los requisitos globales de la aplicación o sistema a ser construido utilizando estándares (ISO/IEC 20510, 2011; Losavio, Chirinos, Matteo y Ramdane-Cherif, 2004; Losavio, Matteo y Pacilli, 2014).

Proceso MN-PEC

En este estudio, se toma PEC, propuesto por Losavio, Matteo y Pacilli (2014) que se muestra en la Figura 2, incorporando el MN, ver Figura 3. Se considera que MN es de vital importancia en el dominio educativo para garantizar que el producto de software satisfaga los requisitos del usuario y de calidad, además de satisfacer tanto las políticas educativas y los enfoques pedagógicos válidos en el contexto para el cual se desarrolla; es en este modelo donde se pueden especificar los propósitos y las restricciones curriculares.

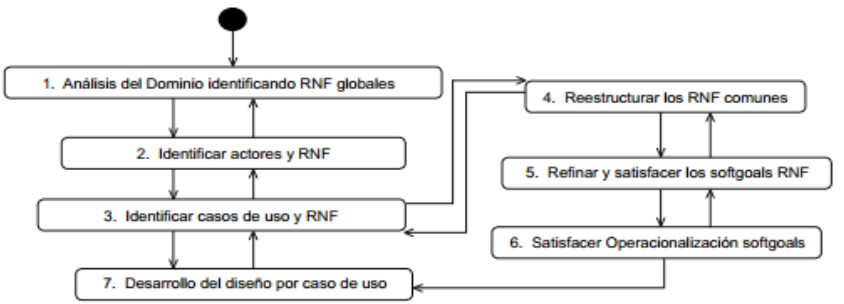

***Figura2*. Diagrama de Actividad describiendo PEC**
Tomado de Losavio, Matteo y Pacilli (2014, p. 18)

MN-PEC es un proceso de desarrollo que se define en el presente trabajo que permitirá obtener una AR para SE que satisfaga todas las características de calidad deseables en el dominio. En lo que sigue se desarrollarán los pasos 1 y 2 de MN-PEC:

MN-PEC - Paso 1. Modelado del Negocio

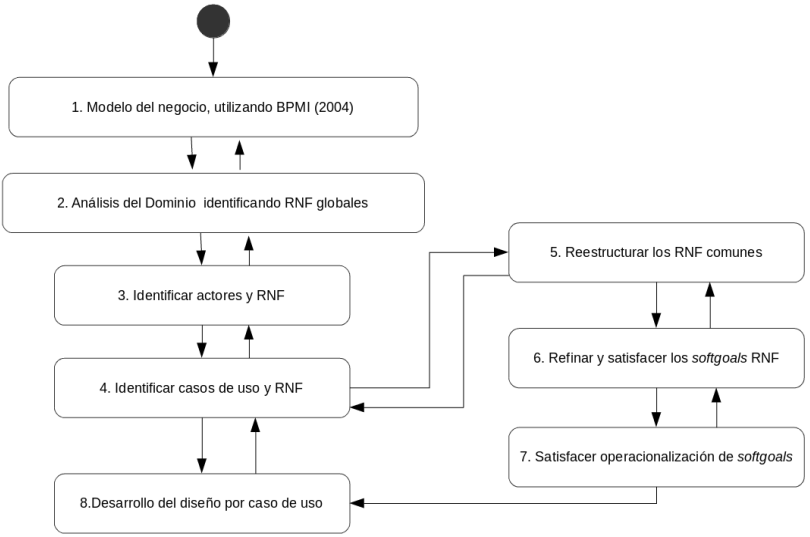

***Figura 3*.Propuesta de PEC extendido con modelo de negocio: MN-PEC**

Entrada: descripción textual del problema que permita la determinación del modelo de negocio propio del dominio.

Especificar los procesos relativos al negocio siguiendo las fases para la gestión de los procesos de negocio de una organización BPMI (BPMI, 2004; Fuentes y Sánchez, 2005), Berrocal, García y Murillo (2005). Estas fases y actividades establecen el ciclo de vida (ver Figura 4) que se debe seguir para alcanzar los objetivos y beneficios perseguidos por el modelado de los procesos de negocio:

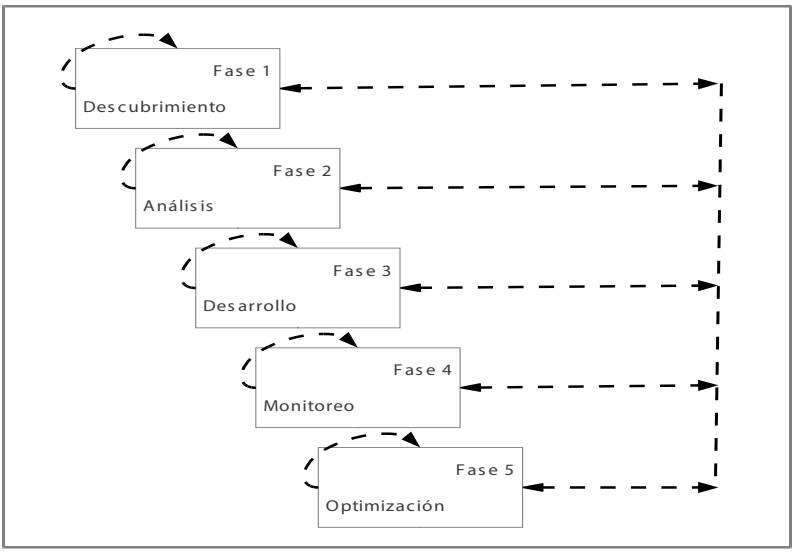

***Figura 4*. Ciclo de Vida de los Procesos de Negocio (las líneas segmentadas indican relaciones bidireccionales entre cada fase). Berrocal, García y Murillo (2005)**

1.1 Descubrimiento

El principal objetivo es identificar y entender cada uno de los procesos de negocio que forman la organización. Especificando todos los detalles de cada uno de los requisitos exigidos y centrándose, principalmente, en las funcionalidades clave del sistema; en este dominio estas funcionalidades, en gran medida, se derivan de las reglas del negocio, las cuales vienen dadas por las decisiones curriculares del contexto educativo donde se diseña. Esto es, por ejemplo, teoría pedagógica subyacente, competencias genéricas a desarrollar, entre otros.

a. Identificar procesos de negocio: tales como *Añadir Conocimiento* (vinculado con las teorías pedagógicas seleccionadas), *Realizar consulta* (relacionado con el rol que se espera que asuma el estudiante), *Validar respuestas* (es necesario fijar posición con respecto al manejo de los errores), entre otros procesos.

b. Especificar procesos de negocio: Se hará utilizando la notación BPMN; adicionalmente se debe especificar **la meta del proceso**, que en general es un RF y los **RNF asociados con esa meta.**

1.2 Análisis

En esta fase se estudia cada uno de los procesos de negocio del sistema, modelándolos con las nuevas características y reglas que deben seguir para obtener una mayor productividad. Una vez identificado los procesos del negocio, desde el punto de vista pedagógico, se asocian con los RF y los RNF que pudieran ser particulares del dominio, o bien del desarrollo en particular.

1.3 Desarrollo

Se depuran los procesos de negocio analizados y diseñados en la etapa anterior y se establecen las especificaciones para su ejecución ya sea automatizada o no.

1.4Monitoreo

Cada proceso de negocio debe ser controlable y por lo tanto medible, para saber su grado de éxito y calidad con el que ha sido llevado a cabo; de esta forma, se pueden analizar los resultados de cada uno de los procesos para que puedan ser redefinidos y optimizados.

1.5 Optimización:

Aquellos procesos que no han cumplido las expectativas deseadas, son rediseñados para que puedan mejorar su rendimiento y así también el de la empresa.

Salida: reglas del negocio, MN en BPMN, RF, RNF

MN-PEC – Paso 2. Análisis de dominio

Entrada: MN especificado en BPMN, reglas del negocio, RF, RNF

a. Identificar el (los) estilo (s) arquitectural (es) y/o soluciones arquitecturales del dominio
b. Definir la funcionalidad del dominio: listar las funcionalidades principales para una familia; esta etapa se facilita con los resultados del MN del paso 1.
c. Definir el modelo de calidad del dominio, utilizando el estándar ISO/IEC 20510 (puede utilizarse cualquier otro estándar); esta etapa se facilita con los resultados del MN del paso 1.
    i. Se especifica la calidad arquitectural del dominio, utilizando los objetivos de calidad críticos (prioritarios) de una solución arquitectural genérica o estilo para una familia del dominio

ii. Se especifica la calidad funcional del dominio, donde se identifican las funcionalidades propias a la familia y a cada funcionalidad se asocian los requisitos de calidad, como objetivos que deben cumplirse para lograr las funcionalidades.

iii. Salida: modelo de calidad de la familia del dominio

2. Identificación de RNF globales

* identificar RNF a los cuales debe responder todo el sistema

3. Identificación de actores y RNF

* identificar los participantes como usuarios del sistema y a identificar los RNF del sistema

4. Identificación de Casos de Uso y RNF

* reestructurar los RNF comunes
* refinar y satisfacer RNF expresados como "*softgoals*"
* identificar las operacionalizaciones de los *softgoals* que representa soluciones arquitecturales para satisfacer los RNF expresados como <<include>> en el modelo de casos de uso

5. Realizar el diseño por casos de uso

* construir AR, haciendo corresponder los casos de uso y las operacionalizaciones obtenidas en el paso 4 con componentes arquitecturales
* determinar el modelo de variabilidad de AR identificando componentes comunes y variantes

En este trabajo solo trataremos los pasos (1) Modelado del Negocio y (2) Análisis del Dominio del MN-PEC; en el paso 2 solo nos referiremos a la identificación de los principales estilos y/o soluciones arquitecturales.

Aplicación de MN-PEC al caso de estudio: Software Educativo

En esta sección se valida MN-PEC respecto al dominio de SE. Se comienza describiendo el modelo de negocio, usando BPMI (2004), para

simplificar la lectura, sólo se desarrollarán las fases de Descubrimiento y Análisis, en trabajos posteriores, se analizará el dominio, llegando hasta la propuesta de las características de calidad prioritarias en el dominio.

MN-PEC - Paso 1. Modelado del Negocio del SE

Entrada: Descripción textual del problema: Software Educativo

No existe una acepción única para el término SE. Para Sánchez (2011), es "cualquier programa computacional cuyas características estructurales y funcionales sirvan de apoyo al proceso de enseñar, aprender y administrar". Un concepto más restringido sería aquel que lo define como "cualquier material de aprendizaje especialmente diseñado para ser utilizado con una computadora en los procesos de enseñanza y aprendizaje". Según Rodríguez (2000) "es una aplicación informática, que soportada sobre una estrategia pedagógica bien definida, apoya directamente el proceso de enseñanza aprendizaje constituyendo un efectivo instrumento para el desarrollo educacional del hombre del próximo siglo".

Lo anterior evidencia la necesidad de caracterizar y clasificar al SE de manera tal que permita distinguir entre éstos y las herramientas computacionales útiles en el proceso enseñanza-aprendizaje. En este sentido, uno de los principales aspectos a considerar en el momento de diseñar o elegir un SE, son los elementos fundamentales de la teoría pedagógica afín al docente; por ejemplo, en este trabajo resulta relevante que en el diseño o selección se tenga presente:

1. Qué significa aprender
2. El rol del estudiante en el proceso enseñanza-aprendizaje
3. Las teorías pedagógicas subyacentes en el proceso
4. Las estrategias de enseñanza preferentes en el área de estudio

Además, un SE debe vincularse directamente con los propósitos instruccionales para los cuales ha sido diseñado, incorporando las estrategias educacionales y de evaluación acordes a tales propósitos. Sin embargo, es importante contemplar la posibilidad de que no todas las exigencias educativas pueden satisfacerse plenamente. En este orden de ideas, para Giacosa, Giuliano, Giorgi y Concari (2010) al momento de seleccionar / diseñar un SE se deben tomar en cuenta, además de los criterios pedagógicos mencionados, criterios que hacen referencia a aspectos funcionales y, si fuera comercial, también habría que contemplar el económico.

Reglas del negocio

Con respecto a las características propias de las plataformas para el aprendizaje electrónico, surgen algunas reglas del negocio que toda institución educativa debe considerar (Álvarez, 2003):

1. *Proporcionar acceso al contenido* desde cualquier lugar, a través de un navegador de Internet.
2. El *contenido* (tema o asunto que se analiza en el curso) debe ser *interoperable* para poder ser compartido y *portable* para ser independiente de la plataforma, de tal manera de poder utilizar diferentes plataformas para acceder un mismo contenido.
3. El *entorno de aprendizaje* (lugar, espacio, comunidad o sucesión de hechos que promueven el aprendizaje) debe ser *adaptable* o personalizable.
4. Se deben *diseñar contenidos que puedan ser utilizados una y otra vez* en diferentes asignaturas, cursos o programas educativos, es decir los contenidos deben ser *reutilizables*.

Estas reglas del negocio, generales, seguirán analizándose y depurándose, para ser adaptadas de acuerdo con las restricciones curriculares reinantes en la institución para la cual se diseña el software. En la especificación de cada proceso se precisará la influencia de estas reglas.

Modelo de negocio

1.1 Descubrimiento

Procesos: *Realizar Consulta*, *Añadir Conocimiento*, *Validar Respuestas*, son los más relevantes, entre otros que no se especificarán en este artículo.

1.2 Análisis

La siguiente descripción del flujo de ejecución del proceso de negocio: *Realizar Consulta* (ver Figura 5) se da como un ejemplo. En este proceso intervienen el estudiante con el sistema, la participación del docente se realiza en el proceso previo: *Añadir Conocimiento*, y en otro posterior, *Validar Respuestas*.

1. Los actores a considerar son el *sistema* y los *estudiantes*.
2. El proceso inicia cuando los estudiantes solicitan acceder al software para consultar un tema en específico.
3. El sistema recibe la solicitud y analiza la lista de temas.
4. El sistema muestra los contenidos disponibles.

5. El estudiante selecciona el contenido con el que va a trabajar
6. El sistema muestra el contenido y las preguntas relacionadas.
7. El estudiante responde la pregunta.
8. El sistema indica si es o no correcta y presenta la opción de responder otra pregunta.
9. Finaliza el proceso cuando el estudiante recibe la estadística de respuestas acertadas.

Requisitos Funcionales: *Realizar Consulta*

Requisitos No Funcionales o metas de calidad: *disponibilidad del sistemas, seguridad de acceso, eficiencia en tiempo, precisión (corrección) respecto a las respuestas, completitud de las tareas realizadas, facilidad de uso para el estudiante*

Reglas de negocio que pueden afectar al proceso: *Proporcionar acceso al contenido.*

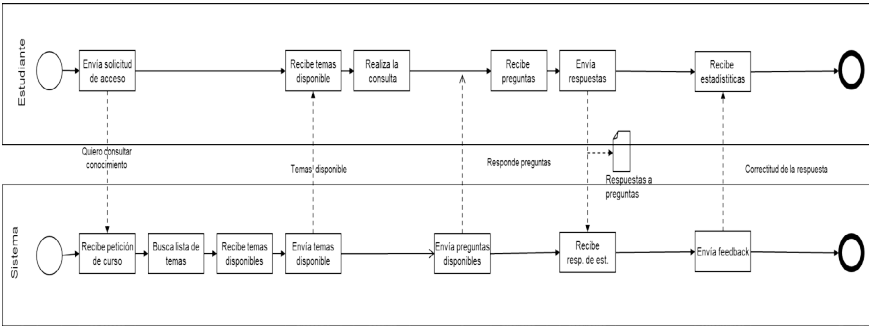

Figura 5. **Modelo de Negocio usando BPMN Proceso: Realizar consulta**

2. Análisis de dominio

a. Estilos y/o soluciones arquitecturales para Software Educativo

Los modelos y arquitecturas de referencia que particularizan un estilo, imponen una serie de restricciones sobre el mismo y realizan por lo general una descomposición y definición estándar de componentes. En el dominio de los SE resulta complicado el generalizar las características arquitecturales que permitan establecer una arquitectura de referencia única para este tipo de aplicaciones electrónicas, pues no existen estándares que guíen el proceso de desarrollo, es escasa la documentación inherente a los procesos que implican la toma de decisiones arquitecturales y, lo más serio aún, las

particularidades del negocio contempla restricciones locales, regionales y nacionales, resultando casi imposible hablar de una arquitectura de referencia única para este tipo de aplicaciones. De las reglas del negocios formuladas en el paso anterior, con respecto a la arquitectura, se desprende que se requieren aplicaciones Web, que utilizan un navegador, es decir que son cliente-servidor/SOA[1]/capas: capa de proceso con servidores de aplicaciones para la gestión de contenidos, capa de datos para la base de datos, capa de presentación con despliegue de páginas en navegador y finalmente la capa de transmisión para la gestión de la comunicación, que concierne o entrecruza a las demás capas.

Al analizar los diferentes tipos de SE, puede observarse que en general, poseen tres módulos principales claramente definidos, (Marqués, citado en Vidal, Gómez y Ruiz, 2010):

a. El *módulo o componente que gestiona la comunicación* (sistema input/output), a través del cual los programas establecen el diálogo con los usuarios y es el que posibilita la interactividad característica de este tipo de software. Corresponde a la capa de presentación.
b. El *módulo o componente que contiene los contenidos informativos del programa* (bases de datos), donde se obtiene la información específica que se presentará a los estudiantes; corresponde a la capa de Datos, y
c. El *módulo o componente que gestiona las actuaciones y las respuestas a las acciones de los usuarios (motor)*. El motor o algoritmo, corresponde a la capa de proceso y gestiona las secuencias en que se presenta la información de las bases de datos y las actividades que pueden realizar.

En este estudio se hace un ejercicio respecto a las soluciones arquitectónicas deseables para cinco tipos de SE, clasificados de acuerdo con sus propósitos y modos de interactuar, advirtiendo al lector que pudieran encontrarse tantas propuestas, como objetivos persigan los docentes involucrados en un proceso enseñanza-aprendizaje que utilice esta herramienta tecnológica. En el Cuadro 1 se describen algunos SE con las características/sub-características de calidad ISO/IEC 2510, deseable en cada caso.

1. Service Oriented Architecture http://www.soaagenda.com/

Cuadro 1.
Caracterización del Software Educativo

Tipo de SE	Características	Clasificación	Descripción	Características de calidad prioritarias	Comentarios	Estilo/ Solución arquitectónica
Programas tutoriales (Programas didácticos, plataformas e-learning, entre otros)	Dirigen el trabajo de los estudiantes y controlan en todo momento su actividad	Programas lineales	Son poco interactivos, se limitan a presentar una secuencia de información o ejercicios.	- Funcionalidad - Facilidad de uso - Portabilidad (adaptabilidad) - Mantenibilidad	Su diseño sigue procesos menos formales. Se pueden realizar para satisfacer necesidades instruccionales particulares	Capas, MVC, sistema "stand-alone"
		Programas ramificados	Ofrecen más opciones de interacción, al ordenar el material según la complejidad (son multinivel).	- Funcionalidad - Facilidad de uso - Rendimiento (Tiempo) - Portabilidad (adaptabilidad) - Mantenibilidad		Capas, MVC, sistema "stand-alone"
		Entornos virtuales	Proporcionan herramientas de búsqueda y de proceso de la información que pueden utilizarse libremente para construir las respuestas. Un ejemplo de este tipo de software, son los entornos de resolución de problemas.	-Funcionalidad - Facilidad de uso - Rendimiento (tiempo) - Portabilidad (adaptabilidad) - Fiabilidad (Disponibilidad)	Ofrecen la posibilidad de digitalizar la realidad para compartir información y comunicarse con otros entornos	Event-driven (SOA); Web Services

Continuación. Cuadro 1.

Tipo de SE	Características	Clasificación	Descripción	Características de calidad prioritarias	Comentarios	Estilo/ Solución arquitectónica
		Sistemas tutoriales expertos	Tienden a reproducir un diálogo auténtico entre el programa y el estudiante; pretenden comportarse como lo haría un tutor humano: guían al estudiante paso a paso en el proceso, analizan su estilo de aprendizaje y sus errores, proporcionando respuestas en función de esa interacción. Están basados en Inteligencia Artificial.	-Funcionalidad - Facilidad de uso - Rendimiento (tiempo) - Compatibilidad (interoperabilidad) - Seguridad - Fiabilidad (Disponibilidad)	Son los más utilizados para mediar el aprendizaje electrónico (e-learning), han avanzado desde las plataformas para mediar el aprendizaje de contenidos, hasta los campus virtuales.	(SOA); Web Services
Bases de datos (Google scholar, EBSCO, entre otros)	Proporcionan datos organizados, en un entorno estático, según determinados criterios; facilitan su exploración y consulta selectiva.	Bases de datos convencionales	Almacenan la información en ficheros, mapas o gráficos. El usuario la utiliza según su criterio.	- Seguridad - Fiabilidad (disponibilidad) - Mantenibilidad - Portabilidad (adaptabilidad)	Suelen utilizarse en las bibliotecas para organizar la información presentada en sus ficheros.	Repositorio
		Bases de datos tipo sistema experto	Recopilan la información existente de un tema concreto y además asesoran al usuario cuando accede buscando determinadas respuestas.	- Seguridad - Fiabilidad (disponibilidad) - Mantenibilidad - Facilidad de uso - Rendimiento (tiempo) -	Los repositorios de datos, que preservan la memoria académica de las instituciones educativas, son cada vez más utilizados	Repositorio

Continuación. Cuadro 1.

Tipo de SE	Características	Clasificación	Descripción	Características de calidad prioritarias	Comentarios	Estilo/ Solución arquitectónica
Simuladores (***Physlets,*** realidad virtual, etc)	Presentan un modelo o entorno dinámico; facilitan la exploración y modificación de datos de situaciones difícilmente accesibles en la realidad.	Modelos físico-matemáticos	Obedecen a leyes representadas por un sistema de ecuaciones deterministas.	- Funcionalidad - Facilidad de uso - Rendimiento (tiempo) - Compatibilidad (interoperabilidad) - Seguridad - Fiabilidad (disponibilidad)	Útiles para modelar fenómenos difíciles de reproducir en un laboratorio	Event-driven (SOA); Web Services
		Entornos sociales	Presentan una realidad regidas por leyes no del todo deterministas, se incluyen juegos de estrategias, por ejemplo, que exigen acciones cambiantes.	- Funcionalidad - Facilidad de uso - Rendimiento (tiempo) - Compatibilidad (interoperabilidad) - Seguridad	Los simuladores de vuelo, así como algunos juegos colectivos, útiles para el aprendizaje de las normas deportivas.	Event-driven (SOA); Web Services
Construc-tores *(Visual basic, etc)*	Poseen un entorno programable que facilita elementos para construir entornos más complejos.	Constructores específicos	Ponen a disposición una serie de mecanismos de actuación que les permiten llevar a cabo operaciones de un cierto grado de complejidad mediante la construcción de determinados entornos, modelos o estructuras	- Funcionalidad - Rendimiento (tiempo) - Facilidad de uso - Mantenibilidad - Portabilidad (adaptabilidad)	Especies de plantillas utilizadas para generar recursos instruccionales	Capas, MVC, sistema "stand-alone"

Continuación. Cuadro 1.

Tipo de SE	Características	Clasificación	Descripción	Características de calidad prioritarias	Comentarios	Estilo/ Solución arquitectónica
		Lenguaje de programación	Como LOGO, PASCAL, BASIC..., que ofrecen unos "laboratorios simbólicos" en los que se pueden construir un número ilimitado de entornos.	- Funcionalidad - Compatibilidad (interoperabilidad) - Fiabilidad (disponibilidad, madurez, tolerancia a fallas) - Portabilidad (adaptabilidad) - Rendimiento (tiempo)	Mayormente utilizados en las aulas de computación o, en general, de informática educativa o aplicada	Máquina virtual, pipe-filter (compilador)
Herramientas *(Office Libre Office, Open Office,* etc)	Proporcionan un entorno instrumental con el cual se facilita la realización de ciertos trabajos generales de tratamiento de la información: escribir, organizar, calcular, dibujar, transmitir, captar datos, entre otros.	Procesadores de texto	Con la ayuda de una impresora, convierten el ordenador en una máquina de escribir	- Facilidad de uso Rendimiento (tiempo) - Compatibilidad (interoperabilidad) - Fiabilidad (disponibilidad, madurez, tolerancia a fallas)	Son utilizados de manera generaliza-da; su aprendizaje inicia desde las etapas tempranas de formación	Capas, MVC, sistema "standalone", plugins.
		Gestor de base de datos	Sirven para generar potentes sistemas de archivo ya que permiten almacenar información de manera organizada y posteriormente recuperarla y modificarla.	- Facilidad de uso - Compatibilidad (interoperabilidad) - Rendimiento (tiempo)	Son utilizados de manera generaliza-da; su aprendizaje inicia desde las etapas tempranas de formación.	Capas con tres niveles: extremo, interno y conceptual

Continuación. Cuadro 1.

Tipo de SE	Características	Clasificación	Descripción	Características de calidad prioritarias	Comentarios	Estilo/ Solución arquitectónica
		Hojas de cálculo	Convierten el ordenador en una versátil y rápida calculadora programable, facilitando la realización de actividades que requieran efectuar muchos cálculos matemáticos.	- Funcionalidad (correctitud) - Facilidad de uso Rendimiento (tiempo) - Compatibilidad (interoperabilidad) - Fiabilidad (disponibilidad, madurez, tolerancia a fallas)	Son utilizados de manera generaliza-da; su aprendizaje inicia desde las etapas tempranas de formación	Capas MVC, sistema "stanalone)
		Editores gráficos	Se emplean desde un punto de vista instrumental para realizar dibujos, portadas para los trabajos, murales, anuncios, etc.	- Funcionalidad (correctitud) - Facilidad de uso Rendimiento (tiempo) - Compatibilidad (interoperabilidad) - Fiabilidad (disponibilidad, madurez, tolerancia a fallas)	Son utilizados de manera generaliza-da; su aprendizaje inicia desde las etapas tempranas de formación.	
		Programas de mensajería	Permiten que equipos lejanos (si disponen de módem) se comuniquen entre sí a través de las líneas telefónicas y puedan enviarse mensajes y gráficos, programas. Ejemplos: "chats", Facebook, wikis, entre otros.	- Seguridad - Rendimiento (tiempo) - Compatibilidad (interoperabilidad) - Fiabilidad (disponibilidad)	Se usan de manera cotidiana, con o sin fines educativos; muy popular entre los adolescentes	SOA / Web *services*
		Programas de experimentación asistida	Recogen datos sobre el comportamiento de las variables que inciden en determinados fenómenos. Permiten construir tablas y elaborar representaciones gráficas de las variables estudiada	- Funcionalidad - Compatibilidad (interoperabilidad) - Rendimiento (en tiempo)	Utilizadas mayormente en clases de laboratorio	SOA / Web *services*

Nota. Los tipos de SE y sus características fueron adaptados de Marqués, 2010

Como puede verse en el cuadro anterior, son muchas las propuesta referidas a SE, muchos tipos de sistemas de software son utilizados para fines educacionales, aunque no fueron construidos para eso, y como se decía anteriormente, esta clasificación podría ampliarse al realizar una revisión más exhaustiva de los SE que se encuentran disponibles, aun así, podrían encontrarse propuestas no comerciales que utilicen otro tipo de arquitectura en su desarrollo.

b. Definir las funcionalidades del dominio

Algunas funcionalidades del dominio están relacionadas con: Identificar usuarios; Registrar la sesión de trabajo; Ofrecer respuesta en cada interacción; Ingresar, modificar o eliminar nuevo contenido; Emitir reporte de usuarios; Registrar estadísticas de uso, entre otros.

c. Construcción del modelo de calidad

Tomando como base lo anterior y los resultados del MN del paso 1, los SE estudiados posean las siguientes características de calidad prioritarias:

- *Funcionalidad*, determina la capacidad del producto para proveer funciones que cumplan con necesidades específicas o implícitas, cuando el software es utilizado bajo ciertas condiciones. Sub-características: adecuación al propósito (completitud) y precisión.
- *Facilidad de uso,* para que un SE sea útil para el aprendizaje, debe generar actividades que motiven y mantengan la atención, a la vez que respondan a diferentes estilos de aprendizaje. Esta característica, garantiza que el SE sea atractivo, entendible, aprendido y utilizado bajo las condiciones para las cuales se diseña. Sub-características: facilidad de comprensión, uso y operación. Debe observarse que la arquitectura no es responsable de hacer cumplir esta característica.
- *Confiabilidad*, debido a que es importante que funcione bajo las condiciones establecidas y mantenga un nivel específico de rendimiento, para garantizar un ambiente adecuado bajo condiciones específicas. Sub-características deseables: madurez, disponibilidad y tolerancia a fallas. Es claro que en el caso de aplicaciones Web, la confiabilidad depende de la disponibilidad de la conexión a la red; si no hay conexión, el sistema no cumplirá con sus compromisos.

- *Portabilidad*, se espera que el software pueda ser ejecutado en diferentes plataformas.
- *Compatibilidad,* se espera que los diferentes componentes del SE puedan ser combinados con otros. Sub-características deseables: interoperabilidad
- *Mantenibilidad*, se requiere que el SE pueda ser modificado para que pueda evolucionar: corregir fallas, mejorar su rendimiento u otros atributos, o para adaptarlo a un entorno cambiante. Sub-características deseables: modificabilidad, modularidad, reutilización.
- *Seguridad*, debe garantizar la confidencialidad de la información, por lo que requiere la autentificación de los usuarios.
- *Eficiencia*, respecto a la capacidad para manejar el volumen de usuarios simultáneos.

Conclusiones

Es poco frecuente encontrar en el desarrollo de SE procesos que concilien elementos pedagógicos con los tecnológicos, por lo que los productos obtenidos suelen tener un tiempo de vida muy corto; bien sea porque son difíciles de mantener, o bien porque no son escalables. Por otro lado, cuando se consideran solo los procesos inherentes a la Ingeniería del Software, se corre el riesgo que los productos no satisfagan las necesidades instruccionales de los docentes. Ambos problemas se pueden resolver considerando un proceso de desarrollo que considere el análisis del dominio y parta del modelo del negocio, que en el caso educativo, es donde se ven reflejadas las necesidades curriculares inherentes al proceso de enseñanza-aprendizaje. Todo esto sería la entrada para la delimitación de las características de calidad arquitecturales y permitiría obtener una arquitectura de referencia específica para el dominio.

Reconocimiento

Queremos hacer un especial reconocimiento al Postgrado en Ciencias de la Computación, Facultad de Ciencias, Universidad Central de Venezuela y al Banco Central de Venezuela, por su colaboración en esta investigación.

Referencias

Alfonzo Pedro y Mariño Sonia (2013). Los estándares internacionales y su importancia para la industria del software. *Ciencia y Técnica Administrativa*, 12 (2), N°2. Buenos Aires. [Documento en línea] Disponible en: http://www.cyta.com.ar/ta1202/v12n2a3.htm [Consulta: 2015, marzo 29]

Álvarez José (2003). Uso de estándares e-learning en espacios educativos. *Revista Fuentes*, 5,153-172. España: Universidad de Sevilla.

Arias Jesús (2005). *Definición de un modelo para la verificación formal de procesos de negocio*. Madrid: Universidad Carlos III. [Documento en línea]. Disponible en: http://www.it.uc3m.es/jaf/tesis/tesis.pdf [Consulta: 2015, enero 19].

Bérard Edward (1992). Testing of object-oriented software. *Proceedings of the eighth international conference on Technology of object oriented languages and systems.* EEUU: Prentice-Hall, Inc.

Berrocal Javier, García José y Murillo Juan (2005). *Hacia una gestión del proceso software dirigida por Procesos de Negocio.* [Documento en línea]. Disponible en: http://alarcos.esi.uclm.es/pnis/articulos/pnis-07-Berrocal-GPSDPN.pdf [Consulta: 2015, febrero 20].

Business Process Management Initiative, BPMI (2004). *Business Process Modeling Notation (BPMN)* [Documento en línea]. Disponible en: http://www.omg.org/spec/BPMN/2.0/ [Consulta: 2015, marzo 02].

Delphi Group. BPM2002. (2011) *Marke Milestone Report*. [Documento en línea]. Disponible en: http://www.fujitsu.com/downloads/SG/fapl/workflow/iflow_delphi_report.pdf[Consulta: 2014, diciembre 17].

Fuentes Lidia y Sánchez Pablo (2005). *Desarrollo de software con aspectos dirigido por modelos*.[Documento en línea]. Disponible en:http://www.dsi.uclm.es/personal/elenanavarro/dsoa/papersCR/Sanchez_desarrollo.pdf [Consulta: 2015, marzo 29].

Garlan David y Shaw Mary (1994). *An Introduction to Software Architecture*. *Advances in Software Engineering and Knowledge Engineering*, Volume I. New Jersey: World Scientific Publishing Company.

Giacosa Norah, Giuliano Mónica, Giorgi Silvia, y Concari Sonia (2010). *Criterios para evaluar applets de física universitaria*. [Documento en línea]. Disponible en: http://www.cbc.uba.ar/noti/ [Consulta: 2010, diciembre 14].

ISO/IEC 25010 (2011). *Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE)*. Systems and software quality models, ISO/IEC JTC1/SC7/WG6. Ginebra: autor.

ISO/IEC (2006). *Systems and software engineering*. Systems and software Quality Requirements and Evaluation (SQuaRE). Ginebra: autor.

Krueger Charles (2002). *Easing the Transition to Software Mass Customization*. Berlin: F. van der Linden (Ed.).

Losavio Francisca, Chirinos Lenis, Matteo Alfredo, Lévy Ramdane-Cherif (2004). Designing Quality Architecture: Incorporating ISO Standards into the Unified Process. *Journal of ISYM*, 21(1), 27-44. Johannes Gutenberg: Universität Mainz.

Losavio Francisca, Matteo Alfredo y Pacilli Irma (2014). Unified Process for Domain Analysis integrating Quality, Aspects and Goals**.** *CLEI Electronic Journal*, 17(2), paper 1. [Documento en línea]. Disponible en: http://www.clei.org/cleiej/paper.php?id=298 [Consulta: 2015, febrero 20].

Vidal María, Gómez Freddy y Ruiz Alina (2010). Software educativo. Educational softwares**.** *Educación Médica Superior,* 24(1). Cuba: Universidad de La Habana.

Owen Martin y Raj Jog (2005). *BPMN and Business Process Management An Introduction to the New Business Process Modeling Standard*. [Documento en línea]. Disponible en: http://www.bptrends.com/publicationfiles/03-04%20WP%20BPMN%20and%20BPM%20Owen-Raj.pdf [Consulta: 2014, noviembre 20].

Pohl Klaus, Böckle Günter y van der Linden Frank (2005). SPL engineering - foundations, principles, and techniques. *Springer IXXVI*, pp. 1-467. Germany: Universität Triet.

Pressman Roger (2001). *Ingeniería del Software: Un enfoque práctico*. México: McGraw Hill.

Rodríguez Raúl (2000). La Informática educativa en el contexto actual. *Edutec. Revista Electrónica de Tecnología Educativa*, 13. [Documento en línea]. Disponible en: http://edutec.rediris.es/Revelec2/Revelec13/Rlamas.html [Consulta: 2014, marzo 15]

Rodríguez Alfonso, Fernández Eduardo y Pattini Mario (2005). *Hacia la definición de Procesos de Negocios Seguros basados en una Arquitectura Dirigida por Modelos*. [Documento en línea]. Disponible en: http://www.face.ubiobio.cl/~alfonso/Publicaciones_Conferencias.html [Consulta: 2015, enero 14].

Sánchez Jaime (2011). *Informática Educativa*. [Documento en línea]. Disponible en: http://www.c5.cl/ie/ [Consulta: 2011, enero 19].

LAS AUTORAS

FRANCISCA LOSAVIO

Profesor titular, jubilado, de la Facultad de Ciencias de la UCV. Doctor en Ciencias de la Computación. Investigador adscrito al Laboratorio MoST, Escuela de Computación UCV.

Área de investigación: Ingeniería del Software. Autora de múltiples artículos de investigación en el área

YULY ESTEVES

Profesor agregado de la UPEL-Miranda. Magister en Educación, mención Enseñanza de la Física. Doctorando en Ciencias de la Computación, área Ingeniería del Software. Escuela de Computación-UCV. Investigador del CIJuMaC, UPEL-IPMJMSM. Área de investigación: Ingeniería del Software. Autora y ponente en eventos tanto en el área educativa, como en el de las Ciencias de la Computación.