

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

Propuesta de metodología de desarrollo de software para objetos virtuales de aprendizaje -MESOVA-¹

Methodology Proposal of Software Development for Virtual Learning Objects - MESOVA -

Proposition de méthodologie de développement des logiciels pour objets virtuels d'apprentissage - MESOVA -

Eucario Parra Castrillón

Ingeniero de sistemas

Magíster en Educación, Universidad de Manizales

Magíster en Software Libre, Universitat Oberta de Catalunya

Docente, Universidad de San Buenaventura

Investigador Católica del Norte Fundación Universitaria

Programa de Ingeniería Informática

Correo: eucarioparra5@gmail.com

Tipo de artículo: Artículo de investigación científica y tecnológica.

Recepción: 2011-07-30

Revisión: 2011-08-11

Aprobación: 2011-08-17

Contenido

1. Introducción
2. Metodologías y modelos de desarrollo de software
3. Ingeniería del software educativo
4. Objetos virtuales de aprendizaje
5. Metodología propuesta
 - 5.1. Principios de MESOVA

¹ Proyecto de investigación: *Diseño y despliegue de objetos de aprendizaje basados en problemas, para la formación en áreas específicas de ingeniería informática*, desarrollado en la Fundación Universitaria Católica del Norte. Fecha de inicio de la primera fase del proyecto: Junio de 2009. Fecha de terminación de esta fase: julio de 2010.

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

- 5.2. Fases y actividades
- 5.3. Puntos de control
- 5.4. Gestión del proyecto
- 5.5. Experimentación y pruebas de MESOVA
- 6. Conclusiones
- 7. Lista de referencias

Resumen

La producción de objetos virtuales de aprendizaje es un problema de los métodos formales de ingeniería del software, ya que su calidad depende del rigor de las técnicas, modelos, métodos y herramientas que se incorporen en su ciclo de vida. Acorde con lo anterior, en la primera parte del artículo se hace una referencia sobre metodologías para desarrollar software, diferenciándose los métodos ágiles, los métodos pesados y los de proceso unificado, y se plantean diferencias entre los modelos denominados en cascada, en espiral y evolutivos. Se abordan los conceptos de ingeniería de software educativo y de objetos virtuales de aprendizaje. Producto de la investigación, el proceso que se propone comprende las fases de concepción, diseño, desarrollo, integración, despliegue, pruebas de aprendizaje y consolidación.

Palabras clave

Ciclo de vida de desarrollo del software, Ingeniería del software, Ingeniería del software educativo, Metodología de desarrollo del software, Métodos de desarrollo del software, Objetos de aprendizaje.

Abstract

The production of virtual learning objects is a problem of the formal methods of software engineering, because its quality depends on the rigor of the techniques, models, methods and tools that join in its life cycle. In accordance with that mentioned above, in the first part of the article a reference is made about methodologies to develop software, being different from light methods, the heavy methods and those of unified process, and differences are proposed between the so-called cascaded models, in spiral and evolutionary. Concepts of educational software engineering and virtual learning objects are treated. As a result, the proposed process has the stages of conception, design, development, integration, deployment, learning tests and consolidation.

Keywords

Life cycle of software development, Software engineering, Educational software engineering, Software development methodology, Software development methods, Learning objects.

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

Résumé

La production d'objets virtuels d'apprentissage est un problème des méthodes formelles de génie des logiciels, puisque sa qualité dépend de la rigueur des techniques, modèles, méthodes et outils qui s'incorporent dans son cycle de vie. Comme déjà mentionné, dans la première partie de l'article on fait allusion aux méthodologies pour développer des logiciels, en étant différentes les méthodes légères, les méthodes lourdes et ces de procès unifiés, et on propose des différences entre les modèles appelés *en cascade*, en spirale et évolutifs. On aborde les concepts de génie des logiciels éducatifs et d'objets virtuels d'apprentissage. Comme résultat de la recherche, le processus qui est proposé à las phases de conception, intégration, déploiement, tests d'apprentissage et consolidation.

Mots-clés

Cycle de vie du développement des logiciels, Génie des logiciels, Génies des logiciels éducatifs, Méthodologie du développement des logiciels, Méthodes de développement, Objets d'apprentissage.

1. Introducción

Una de las dificultades serias en la construcción de objetos virtuales de aprendizaje, es la tendencia a la disociación con la ingeniería del software, originada porque los proyectos generalmente son liderados por profesionales que enfatizan más en el producto que en el método. Esto se evidencia por ejemplo, cuando los esfuerzos se centralizan en las técnicas y tecnologías para producir multimedia o cuando se proponen soluciones, sin considerar estructuralmente análisis de contexto, nivel educativo, ambientes apropiados o particularidades de las temáticas.

Como resultado, las soluciones quedan expuestas al ensayo y error, esto es, a veces aun sin una metodología rigurosa se alcanzan soluciones de alta calidad, pero también es probable que se propongan objetos que difícilmente son incorporados a los procesos de aprendizaje, porque no llenan las expectativas de los profesores y los estudiantes.

En el primer caso, si bien los objetos pueden ser ampliamente aplicados, no queda la documentación suficiente para garantizar su soporte o para ampliar su capacidad, ni subsisten históricos sobre su validación y verificación. En el segundo caso, al no ser aplicados los objetos en los procesos de aprendizaje, lo que se derivan son problemas de dilapidación de recursos, pérdida de credibilidad, ausencia de tecnología para el aprendizaje o inadecuada utilización de infraestructura disponible.

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

Pensando en estos problemas se propone una metodología de desarrollo de software para objetos virtuales de aprendizaje, con el objetivo de formalizar un marco de trabajo cuyas fases y actividades permitan definir ciclos de vida que aseguren metodológicamente su calidad.

Tres aspectos son fundamentales en la metodología propuesta: la especificación, antes del análisis y diseño, de la caracterización de la temática, el nivel educativo y la especificidad pedagógica, como parte de los requisitos no funcionales; la especificidad de la didáctica e instrucción, como parte de los requisitos funcionales; y la evaluación experimental del objeto, luego de su integración, como parte esencial de su desarrollo.

Es importante resaltar que la metodología ya fue probada con la construcción de 13 objetos de aprendizaje para el área de la programación de computadores, una las áreas más críticas de la ingeniería de sistemas, por su nivel de abstracción.

2. Metodologías y modelos de desarrollo de software

La construcción de software es el evento fundamental de la ingeniería de software. Los programadores trabajan construyendo e integrando programas a través de técnicas de codificación, validación y pruebas. Pero ese carácter esencial no minimiza fases tan cruciales como la planeación del proyecto, el análisis de requerimientos, el diseño y la gestión de la calidad.

En el caso del software educativo igualmente deben tenerse en cuenta estas consideraciones. Pues como en todo proyecto de software, el ciclo de vida consiste en convertir los requerimientos planteados por los usuarios en un producto de software con funcionalidades específicas.

En los proyectos de desarrollo de software es primordial la definición de la metodología. Esta se define según la forma como se asuman las distintas actividades para la consecución del producto final de software. Las metodologías aplican distintos modelos de desarrollo tales como el de cascada, el incremental, el evolutivo y el de espiral (Aycart, Ginestá & Hernández, 2007).

El modelo en cascada determina cuatro fases terminales del ciclo de vida, con unos hitos específicos al final de cada una (toma de requisitos, análisis, diseño e implementación). Los modelos incremental y evolutivo permiten la creación de productos en etapas parciales, donde cada etapa agrega funcionalidad a la anterior e incluye las fases del ciclo de vida. El modelo en espiral incluye la creación de prototipos del proyecto que pasan cíclicamente por las fases del ciclo de vida,

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

hasta llegar paulatinamente al producto final, después de validarse repetidamente los requisitos y diseños del proyecto.

Con respecto a las metodologías, en la fabricación del software se conocen dos corrientes, los llamados métodos pesados y los métodos ágiles. La diferencia fundamental está en que mientras que los primeros buscan conseguir el producto de software por medio del orden y la documentación, los otros buscan conseguir el producto de software utilizando la comunicación directa entre las personas que intervienen en el proceso (Molpeceres, 2002).

La programación extrema (XP) es una metodología ágil, se caracteriza porque los requerimientos del software se plantean como escenarios llamados historias de usuarios, los cuales se implementan como una serie de tareas que deben desarrollarse. Aplicando esta metodología los programadores trabajando en parejas hacen pruebas antes de escribir los programas. En general XP como método ágil de desarrollo de software implica las siguientes prácticas (Sommerville, 2005):

- a) Entregas pequeñas y frecuentes a los clientes, en vez de la entrega del sistema en su totalidad.
- b) Participación directa de los clientes o sus representantes en el equipo de desarrollo del proyecto. Además, ellos son responsables de la aceptación de pruebas aplicadas al software.
- c) Cuidado en el proceso de desarrollo, de manera que se priorice el trabajo en parejas, la responsabilidad colectiva sobre el programa y racionalización de las cargas de trabajo evitándose jornadas excesivas.
- d) Desarrollos previamente probados de las entregas (*release*), antes de su integración al sistema global.
- e) Utilización de diseños sencillos y reutilización de programas o partes de éstos.

Los métodos ágiles se originaron en el año 2001 por la inestabilidad del entorno técnico y porque el cliente a veces es incapaz de definir con exactitud los requisitos del proyecto de software. El término ágil se relaciona con la capacidad de adaptarse a los cambios de contexto y de especificaciones que ocurren durante el proceso de desarrollo. Estas metodologías se caracterizan por lo siguiente (kioskea.net, 2008):

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

- a) Individuos e interacciones en lugar de procesos y herramientas. Las personas son el factor de éxito más importante en un proyecto de software. Lo apropiado es elegir adecuadamente el equipo de trabajo y que éste configure su entorno.
- b) Desarrollo de software en lugar de documentación exhaustiva. La documentación debe ser corta y centrarse solo en lo fundamental.
- c) Trabajo con el cliente en lugar de negociaciones contractuales. Debe existir una colaboración constante entre el cliente y el equipo de desarrollo.
- d) Apertura para los cambios en lugar de cumplimiento de planes poco flexibles. El éxito o el fracaso de un proyecto depende de la capacidad de adaptación a los cambios en los requisitos, la tecnología y el equipo de desarrollo.

Con los métodos ágiles el cliente tiene control del proyecto y consigue una rápida implementación del software.

Según Amaro y Valderde (2007), los cuatro valores anteriores son el fundamento de los principios del desarrollo ágil, según los cuales lo prioritario es satisfacer al cliente con entregas parciales de software que le generen algún valor, promover cambios cuando representen ventaja competitiva para el cliente y entregar software funcionando en el menor intervalo posible. Además se plantea también como principio, que desarrolladores y clientes deben trabajar juntos en el proyecto, construir en un entorno de motivación y apoyo para los desarrolladores, fomentar el diálogo cara a cara entre los integrantes del equipo de desarrollo, procurar la armonía permanente entre promotores, desarrolladores y usuarios y prestar atención continua a la calidad técnica. Son principios también la funcionalidad del software como el indicador fundamental del proyecto, la simplicidad de los diseños y la existencia de ambientes para la reflexión del equipo de desarrollo, en torno a cómo ser más efectivo.

Los llamados métodos pesados se caracterizan por su rigidez metodológica y la exhaustiva documentación. Uno de ellos es *Rational Unified Process (RUP)*, basado en *Unified Modeling Language (UML)* para la ingeniería de sistemas y de software.

Según Nextel Engineering (2011), para la metodología RUP el ciclo de vida de un proyecto se divide en las fases de concepción, elaboración, construcción y transición. En la concepción se define el alcance del sistema, la estimación de costos, la síntesis de la arquitectura candidata y la organización del proyecto. La fase de elaboración comprende las actividades para la refinación y validación de la

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

arquitectura y los componentes. Dentro de la fase de construcción se gestionan los recursos, optimización y control de los procesos y se completa el desarrollo de componentes. En la fase de transición se ejecutan los planes de implementación, se terminan los manuales de usuario y técnicos, se integra el sistema y se ajusta según la validación que hace el usuario.

Estas cuatro fases son complejas, exhaustivas, desarrolladas de acuerdo a la escritura de procesos que pretenden asegurar la calidad cuando el producto le llegue al usuario.

Por sus características, RUP está enfocado para grandes proyectos de software y equipos de trabajo de amplio alcance, al contrario de las metodologías ágiles que se ajustan mejor para proyectos y equipos de desarrollo pequeños. Asimismo, mientras que RUP presta interés especial a la calidad de los procesos y a la comunicación con los clientes desde la formulación de contratos, las metodologías ágiles abogan por una comunicación directa y continua. Por otra parte, la planificación en RUP establece inflexibles elementos y puntos de control, mientras que en las metodologías ágiles el énfasis está en entregas rápidas y útiles para los clientes, sin que esto signifique ausencia de planes de los proyectos de software.

Para Calero, Moraga y Piatini (2010) se evidencian diferencias en las metodologías tradicionales y ágiles en la documentación del desarrollo de software. En los métodos ágiles es más concisa y navegable, pero permanece menos actualizada que en los métodos tradicionales. Este aspecto es importante considerarlo, ya que para que la información sea útil, el atributo de calidad más importante es su nivel de actualidad, aunque usualmente se piense que el mantenimiento de la información le quita tiempo a los desarrolladores de software que pueden invertir en tareas propiamente de producción.

Otra metodología es el denominado proceso unificado (UP). Ésta plantea como esencia del desarrollo de software la especificación de requerimientos de un proyecto mediante casos de uso. Tiene una visión que parte de la arquitectura, para llegar al producto final siguiendo rigurosamente procesos iterativos e incrementales. Se basa en los siguientes principios (Weitzenfeld, 2005):

- a) El éxito de un proyecto de software depende del conocimiento exacto de lo que quieren y necesitan los usuarios.
- b) Las arquitecturas de los sistemas basados en software deben permitir visualizaciones desde distintas perspectivas.

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

- c) Una buena práctica en el desarrollo de productos de software es dividir el trabajo en etapas, rigurosamente planeadas, donde cada hito representa un incremento significativo del proyecto.

Para Galindo, Simó, Prieto y Segret (2010), en el desarrollo de proyecto de software la rigidez y la meticulosidad de las metodologías UP pueden ser excesivas, sin que realmente sea necesario realizar todas las actividades propuestas y en el nivel de detalle que se propone. Es así como en reacción a estos métodos precisos surgen las metodologías ágiles.

El proceso unificado se caracteriza porque describe el conjunto de actividades que se deben realizar para transformar sistémicamente los requisitos en un sistema de software. Esta metodología de desarrollo de ciclo de vida de proyectos de software, define seis flujos de trabajo fundamentales: modelado de negocio, captura de requisitos, análisis y diseño, implementación, pruebas y despliegue. Además provee tres flujos de soporte relacionados con la configuración y administración de cambios, gestión de proyectos y ambiente (Amó y Martínez, 2005). Según estos autores, el ciclo de vida del software hace referencia al conjunto de etapas por las que debe pasar un sistema, desde su concepción hasta su producción y despliegue final.

Las metodologías y métodos que se mencionan tienen validez general para el desarrollo de software independiente de la especificidad, uno de los cuales es el software educativo, de aplicación tan vasta como el software financiero, contable, industrial o de relacionamiento de clientes.

3. Ingeniería del software educativo

Puede definirse la ingeniería de software educativo como el estudio de métodos, modelos, técnicas y herramientas para el análisis, diseño, producción, implementación y evaluación de productos de software destinado a facilitar procesos de aprendizaje. Esta definición se fundamenta en los conceptos previos sobre modelos y metodologías de desarrollos de software, las cuales aunque con presupuestos y enfoques distintos, coinciden en que la producción de software de calidad implica la definición de un método y la determinación de requisitos como punto de control de la funcionalidad final del producto.

El desarrollo de software educativo precisa de una estrecha relación de cooperación y conjunción de conceptos entre la práctica docente, la didáctica y la ingeniería del software (Sierra, 2005).

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

Orjuela y Rojas (2008) propusieron una metodología ágil para el desarrollo de software educativo que permite guiar el desarrollo de productos de software que apoyen los procesos de enseñanza. Esta propuesta está orientada a la producción de prototipos funcionales en cada iteración (ciclo de desarrollo) y según las necesidades de los clientes puede adaptarse a la producción de artefactos. Es aplicable a proyectos de diferentes dimensiones, características, niveles de formación y tipologías como micromundos, multimedia, videojuegos, animación, tutoriales.

Abud (2009) propone una Metodología de Ingeniería de Software Educativo (MeISE) que comprende la definición de requisitos y un análisis y diseño preliminar. En esta etapa se determinan las características del producto de software esperado, los requisitos pedagógicos y de comunicación, la arquitectura tecnológica y el plan de iteraciones para alcanzar los objetivos didácticos del software.

El trabajo referenciado de Abud es valioso por cuanto alcanza a definir meticulosamente las actividades que deben cumplirse y los artefactos (hitos) esperados en cada una. Define estos artefactos de forma precisa, de manera que en el momento de aplicar la metodología los desarrolladores encuentran la ruta adecuada. De esta forma se crea una estructura formal para las fases conceptuales, de análisis y diseño inicial, de diseño computacional, de desarrollo y de despliegue. Aparte de esto, propone unos instrumentos para los procesos de documentación y formalización de los procesos del ciclo de vida del software.

Galvis, Gómez y Mariño (1998), presentaron una metodología para el desarrollo de software educativo (ISE) basada en el paradigma de la programación orientada a objetos. Según ellos el ciclo de vida puede tener dos rutas: una en función de los resultados de la etapa de análisis, el ciclo consiste en analizar, desarrollar y probar con base en la necesidad del software. La otra comprende la prueba piloto, y la prueba de campo del producto, de acuerdo con el análisis de las necesidades.

Si bien en su momento la incorporación de los conceptos de orientación a objetos al software educativo podría considerarse como innovación, en el momento actual este concepto es dominante en la ingeniería del software. Pero es un aporte valioso ya que dio una mirada ingenieril a las tareas de construcción de software educativo, tradicionalmente encargada a los especialistas en didáctica y multimedia.

Un aspecto fundamental en la ingeniería del software es el *testing* que consiste en probar el producto de software desde su funcionalidad y su arquitectura. Hay unos principios básicos en los procesos de validación y verificación del software que tiene que ver con lo siguiente (Aycart, Ginestá y Hernández, 2007):

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

- a) Es imperioso disponer en detalle de los requisitos del sistema y las necesidades de los usuarios. El producto final del software debe cumplir con ellos.
- b) Los procesos de *testing* deben hacer parte de las distintas etapas del proyecto, para garantizar la validez de cada una.
- c) El equipo que testea el software debe ser distinto al equipo de desarrollo. La persona o grupo que desarrolla no debe ser quien aplica las pruebas.
- d) El plan de pruebas debe comprender los guiones para probar los componentes del software. Esto hace parte del sistema de calidad del proyecto.

En el ciclo de vida de desarrollo del software, el *testing* demarca ciertas especificidades para el software educativo, al igual que el análisis de requerimientos. Esto se sustenta en el hecho que este tipo de software involucra aspectos didácticos, comunicativos, pedagógicos y ambientales que no siempre son fáciles de formalizar, condición esta que es fundamental en las metodologías.

La dificultad para formalizar requerimientos y necesidades del software es proporcional a la calidad del producto final y es determinante para la configuración de los guiones de pruebas, o sea, si los requisitos no se pueden especificar en detalle, entonces no es posible delinear un buen plan de pruebas, pues éstas se delinearán con base a aquellos.

Una de las aplicaciones de la ingeniería del software educativo, son los objetos virtuales de aprendizaje, cuya aplicación tiene amplia vigencia en la actualidad, por las tendencias de expansión de la educación virtual y los procesos de educación presencial asistidos con recursos tecnológicos.

4. Objetos virtuales de aprendizaje

Según el Ministerio de Educación Nacional (2006), un objeto virtual de aprendizaje (OVA), es un material estructurado de forma significativa, digital, dirigido al aprendizaje, que puede ser distribuido y consultado a través de Internet; además, un OVA debe contar metadatos (fichas de registro) consistente en listados de atributos que describen el uso del objeto y permiten su catalogación e intercambio. Bajo estos conceptos el Ministerio consideró la siguiente tipología de objetos: simuladores, cursos, multimedia, tutoriales, multimedia, animaciones, videos, documentos interactivos y colecciones de imágenes estáticas.

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

Sin embargo, más adelante a raíz de las ambigüedades de esta definición, propuso la siguiente conceptualización: "Un objeto de aprendizaje es un conjunto de recursos digitales, autocontenible y reutilizable, con un propósito educativo y constituido por al menos tres componentes internos: contenidos, actividades de aprendizaje y elementos de contextualización. El objeto de aprendizaje debe tener una estructura de información externa (metadatos) que facilite su almacenamiento, identificación y recuperación" (tomado de Colombia Aprende).

Para Stephens (2004), un objeto de aprendizaje (OA) es un componente digital de contenidos que puede ser utilizado en diferentes contextos. Es una unidad digital significativa de aprendizaje. Son unidades autosuficientes e independientes para el aprendizaje. Las propiedades más importantes de un OA son las siguientes:

- a) Son bloques independientes
- b) Son de dimensiones distintas
- c) Pueden ser reutilizados
- d) Se pueden ensamblar de distinta manera
- e) Manejan un único estándar de ensamblaje

Tal como se presentan, no hay diferencias entre OA y OVA. En este artículo, se tratarán equivalentemente ambos términos, bajo dos supuestos:

- a) Los objetos de aprendizaje tienen estructura digital
- b) Los objetos de aprendizaje deben tener acceso remoto utilizando la web

Los objetos de aprendizaje tienen dos indicadores de uso importantes: la interoperabilidad y la reusabilidad. Según *SCORM (Sharable Content Object Reference Model)*, éstos deben ser reusables, interoperables, durables y accesibles (ADL, 2004). Reusabilidad es la flexibilidad que permite integrar componentes distintos dentro de múltiples contextos y aplicaciones; interoperabilidad es la capacidad de los objetos de utilizarse sobre distintas plataformas tecnológicas. Durabilidad es la capacidad del objeto para resistir a evoluciones tecnológicas, sin necesitar de reconfiguraciones estructurales. Por su parte, accesibilidad es la capacidad de acceder a los componentes de enseñanza desde sitios remotos a través de tecnologías web, así como la facilidad para distribuirlos a otros sitios. Además de lo anterior, los OA deben ser adaptables, o sea que se pueda personalizar la formación en función de las necesidades de las personas y organizaciones.

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

Sin embargo al tema de los objetos de aprendizaje no le atañen únicamente aspectos tecnológicos y de diseño de la usabilidad. La otra dimensión a considerar tiene que ver con la concepción pedagógica, la cual debe considerarse en el diseño y desarrollo. Al respecto, en (Ossandón y Castillo, 2006), citando a Kolb (1984) se propone un diseño de objetos fundamentado en modos de aprender (experiencia concreta, observación reflexiva, conceptualización abstracta, experimentación activa) que determinan a su vez los siguientes estilos de aprendizaje: divergente, asimilador, convergente y acomodador. El primero basado en actividades que comprenden observación de experiencias e imaginaciones reflexivas; el segundo propicio para combinar la conceptualización abstracta con eventos de observación; el siguiente fundamentado en la aplicación e implementación de los aprendizajes y el último encaminado a la actitud propositiva para la adaptación de la información en nuevas experiencias.

La exploración conceptual deja claro que los proyectos de elaboración de objetos de aprendizaje debe considerar aspectos de pedagogía, tecnología y protocolización de procesos. Considerando esto, para su construcción de una manera estructurada, dentro de cánones de ingeniería del software, a continuación se presenta una metodología de desarrollo.

5. Metodología propuesta

La Metodología de Desarrollo de Software para Objetos Virtuales de Aprendizaje - MESOVA - que se propone a continuación, recoge distintos elementos de los marcos de trabajo XP (*Extreme Programming*), RUP (*Rational Unified Process*) y UP (*Unified Process*). Con respecto al flujo del ciclo de vida, reúne conceptos de los modelos en espiral, incremental y evolutivo, dándose especial importancia a la construcción ágil de prototipos.

5.1. Principios de MESOVA

MESOVA asume los siguientes principios de desarrollo:

P1. La temática, el nivel educativo y la intencionalidad pedagógica agrupan requisitos no funcionales básicos. Además de éstos, puede haber otros relacionados con las comunicaciones, la plataforma tecnológica o la gestión del proyecto.

P2. La didáctica, los servicios y las operaciones para interactuar con el objeto de aprendizaje agrupan requisitos funcionales básicos. Tienen que ver con las funciones que el usuario realizar para la aplicación del software.

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

P3. Más importante que la documentación es la funcionalidad del objeto de aprendizaje, sin embargo, ésta hace parte del proceso de desarrollo. Cada actividad incluida en la metodología debe tener el respaldo documental, con parámetros de simplicidad y fácil navegación.

P4. El diseño modular es imperativo para la evolución y despliegue de cada módulo. Se entiende por módulo un subsistema con entradas, salidas y almacenamientos propios y representa una parte temática o metodológica del objeto de aprendizaje.

P5. El prototipo guía el desarrollo y la comunicación con el usuario. Las decisiones sobre funcionalidades deben tomarse en conjunto con los usuarios, con base en la evaluación del respectivo prototipo.

P7. La usabilidad del objeto debe garantizar fácil aprendizaje, manejo y recordación para los usuarios que interactúan con el software.

P8. El usuario es actor central de las pruebas intermedias del ciclo de vida. Estas pruebas validan los requisitos y funcionalidades parciales del objeto.

P9. La calidad del objeto de aprendizaje se deduce, si al confrontarse su aplicación con un grupo experimental, se demuestran aprendizajes iguales o superiores al de un grupo de control. Para esto se deben diseñar y aplicar las pruebas de aprendizaje.

P10. Las pruebas de aprendizaje son posibles sólo cuando el objeto está totalmente integrado y en el ambiente final del usuario, pues la intencionalidad pedagógica y temática no es posible parcializarla.

5.2. Fases y actividades

La Metodología de Desarrollo de Software para Objetos Virtuales de Aprendizaje - MESOVA - tiene la siguiente estructura secuencial, anotando que al interior de cada fase hay actividades planteadas, las cuales son posibles llevar a cabo en secuencia estricta o en acciones paralelas, dependiendo de la naturaleza del proyecto y de la disposición del equipo de trabajo. Véanse la figura 1 y las tablas 1-5.

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

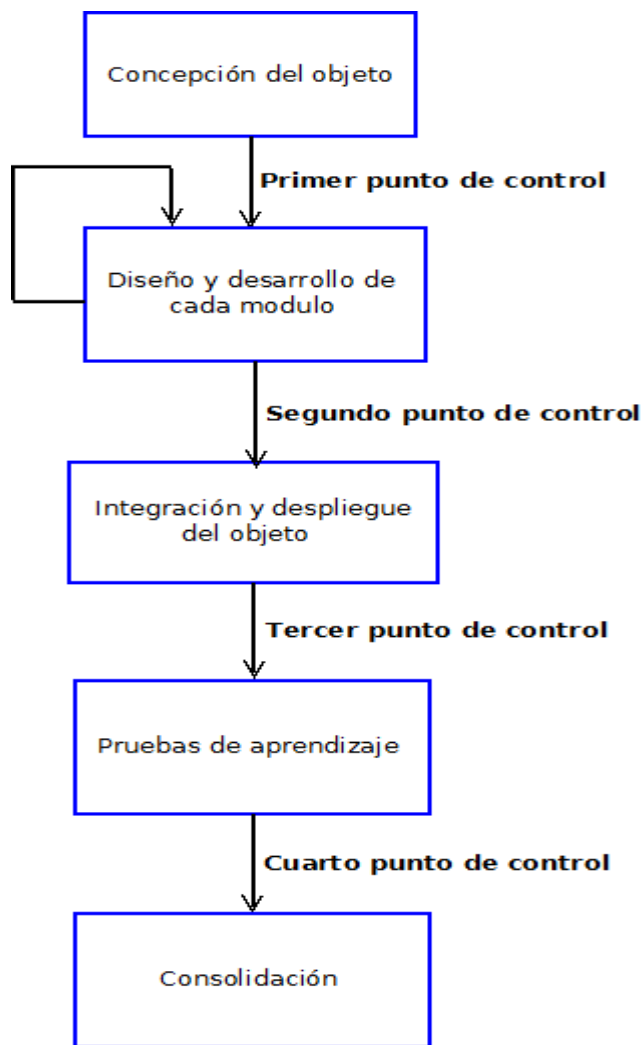


Figura 1. Fases de la metodología MESOVA.

La metodología MESOVA está estructurada con base en seis fases, cada una con sus respectivas actividades y especificación de resultados. Éstas son: concepción del objeto, diseño y desarrollo modular evolutivo, integración y despliegue, pruebas de aprendizaje, consolidación.

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

Fase 1. Concepción del objeto	
Actividades	Resultados
C1. Caracterización de temática y nivel educativo.	Aspectos que delimitan la temática del objeto de aprendizaje. Aspectos que delimitan el nivel educativo del usuario del objeto de aprendizaje.
C2. Especificidad pedagógica.	Descripción simplificada de la intencionalidad pedagógica.
C3. Especificidad de requisitos funcionales.	Precondiciones, didáctica, datos de entrada, estímulos al objeto, respuestas de objeto, almacenamientos y postcondiciones.
C4. Especificidad de requisitos no funcionales.	Necesidades pedagógicas, temáticas, educativas, comunicacionales y ambientales del objeto de aprendizaje.
C5. Diseño modular global.	Perfil gráfico del objeto visualizando los módulos constitutivos o subsistemas.
C6. Modelado de casos de uso.	Modelos de UML (<i>Unified Modeling Language</i>) describiendo las interacciones entre los objetos de aprendizaje y los usuarios.
C7. Modelado transición de estados.	Modelos de UML (<i>Unified Modeling Language</i>) describiendo las secuencias de estados por las que pasan los casos de uso.
C8. Caracterización tecnológica.	Definición de plataforma y dispositivos de hardware y software que se eligió para construir el objeto.
C9. Evaluación y validación del análisis inicial.	Pruebas intermedias. Análisis de instrumento aplicados a los usuarios para validar los resultados de las actividades C1 a C7.

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

	Primer punto de control.
C10. Construcción de un prototipo del primer módulo.	Implementación de las funcionalidades básicas e interfaces de un módulo.
C11. Documentación formal de la concepción y análisis.	Documentos integrados de las actividades C1 a C8, de acuerdo con un sistema de calidad ágil.

Tabla 1. Fase de concepción del objeto.

Fase 2. Diseño y desarrollo modular evolutivo	
Actividades	Resultados
D1. Evolución del prototipo de módulo.	Evolución en implementación de las funcionalidades básicas e interfaces de un módulo.
D2. Evaluación y validación del prototipo.	Pruebas intermedias. Análisis de la evaluación del usuario sobre el prototipo. Segundo punto de control.
D3. Estado final del módulo.	Módulo terminado completamente. Evolución de las funcionalidades del prototipo hasta la consecución final del módulo.
D4. Instalación del módulo en ambiente de usuario final.	Módulo disponible para la interacción con el usuario y su explotación.
D5. Diseño y desarrollo del siguiente módulo.	Hasta que se terminen los módulos planeados se repiten las actividades D1, D2, D3, D4.
D6. Documentación formal del diseño y desarrollo.	Documentos integrados de las actividades D1 a D5, de acuerdo con un sistema de calidad ágil.

Tabla 2. Fase de diseño y desarrollo del objeto.

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

Fase 3. Integración y despliegue	
Actividades	Resultados
I1. Configuración del ambiente.	Consecución de los requisitos técnicos y funcionales del ambiente donde se hospedará el objeto de aprendizaje.
I2. Integración de los módulos.	Integración y despliegue de los módulos en el ambiente final de usuario.
I3. Evaluación y validación del sistema.	Pruebas intermedias. Análisis de la evaluación del usuario sobre el objeto final integrado. Tercer punto de control.
I4. Documentación formal del diseño y desarrollo.	Documentos integrados de las actividades I1 a I3, de acuerdo con un sistema de calidad ágil.

Tabla 3. Fase de integración y despliegue del objeto. Fuente: elaboración propia.

Fase 4. Pruebas de aprendizaje	
Actividades	Resultados
P1. Selección de usuarios de la prueba.	Integración y caracterización de los grupos experimentales y de control.
P2. Construcción de la prueba.	Caracterización y definición de la prueba y del ambiente de aplicación.
P3. Aplicación de la prueba.	Medición, análisis e interpretación de los resultados en los grupos experimental y de control. Cuarto punto de control.
P4. Documentación formal de las pruebas de aprendizaje.	Documentos integrados de las actividades P1 a P3, de acuerdo con un sistema de calidad ágil.

Tabla 4. Fase de pruebas de aprendizaje del objeto. Fuente: elaboración propia.

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

Fase 5. Consolidación	
Actividades	Resultados
S1. Despliegue después de prueba de aprendizaje.	Instalación y configuración definitiva después del cuarto punto de control.
S2. Plan de soporte.	Documento y contrato en que se definan los compromisos con el usuario frente a soporte técnico del objeto creado, la atención a fallos o errores, la incorporación de requerimientos adicionales o las posibilidades de expansión y la documentación sobre procesos, arquitectura y manejo del objeto.
S3. Documentación del objeto.	Resumen ejecutivo de los documentos de las fases anteriores. Manual de usuario del objeto.

Tabla 5. Fase de consolidación del objeto. Fuente: elaboración propia.

5.3. Puntos de control

Primer punto de control: pruebas intermedias de evaluación y validación del análisis inicial.

El objetivo es verificar la consistencia de los requisitos funcionales y no funcionales de acuerdo con las necesidades planteadas por el usuario, en aspectos temáticos, pedagógicos y comunicativos. Sobre el prototipo inicial se hace la verificación y validación y se decide sobre la viabilidad del proyecto, la concreción de sus alcances, las posibilidades tecnológicas y la capacidad del objeto para adaptar la intencionalidad pedagógica y la funcionalidad didáctica.

El equipo de desarrollo y los usuarios deciden la continuidad o no del proyecto o las modificaciones sustanciales o ligeras que se deben hacer con respecto a la intención inicial.

Segundo punto de control: pruebas intermedias de evaluación y validación del prototipo de cada módulo.

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

Para cada uno de los módulos se deben diseñar instrumentos para medición de indicadores de visibilidad, funcionalidad y usabilidad de los prototipos. El diseño de los instrumentos debe validarse con pares técnicos especializados en software.

El equipo de desarrollo y los usuarios deciden dar continuidad o no a los prototipos presentados o plantear las modificaciones sustanciales o ligeras que se deben hacer con respecto a las temáticas, nivel educativo, intencionalidad pedagógica, criterios de comunicación y funcionalidad didáctica.

Tercer punto de control: pruebas intermedias de evaluación y validación del sistema.

Se debe diseñar instrumentos para medición del sistema integrado por todos los módulos, en cuanto a su consistencia, secuenciación, flexibilidad y navegabilidad.

El equipo de desarrollo y los usuarios deciden la aceptación o no del sistema, o plantean las modificaciones puntuales que deben hacerse sobre el sistema integrado.

Cuarto punto de control: prueba de aprendizaje.

Se deben seleccionar grupos experimentales y de control, diseñar instrumentos de aplicación, elegir los ambientes adecuados y aplicar pruebas de aprendizaje aplicando el sistema que integra al objeto virtual de aprendizaje.

Las respuestas de aprendizaje del grupo experimental deben ser iguales o superiores a las del grupo de control (aunque en el diseño y análisis de las pruebas debe considerarse que puede haber diferencias objetivas entre los estudiantes y factores internos y externos de influencia en ellos). El equipo de desarrollo y los usuarios analizan los resultados de las pruebas y definen la entrega definitiva del objeto virtual de aprendizaje o si es preciso la revisión de módulos y de la plataforma integradora.

Este cuarto punto de control reviste algunas características especiales:

- a) Se debe aplicar al final, cuando ya el objeto esté integrado, ya que la medición del aprendizaje se hace es sobre la totalidad del objeto. Así como las pruebas intermedias validan funcionalidades, la prueba final de aprendizaje evalúa es la capacidad del objeto para facilitar los procesos de aprendizaje, o sea, su efectividad para facilitar los aprendizajes. Es de anotar que los módulos, aunque son unidades independientes, pedagógica y didácticamente están integrados desde la concepción inicial de las intencionalidades formativas.

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

- b) Es posible que el objeto amerite una reingeniería significativa después de los resultados de la prueba. En este sentido, por encima de otra circunstancia está la utilidad del objeto para los procesos de aprendizaje.

5.4. Gestión del proyecto

MESOVA tal como está concebido es una metodología que se inspira en los conceptos sobre ciclo de vida de ingeniería de software, con la posibilidad de la siguiente tipología:

- a) Aplicación de herramientas genéricas para la integración y manipulación de componentes de objeto de aprendizaje.
- b) Creación de componentes del objeto de aprendizaje aplicando herramientas para edición y generación de código fuente e interfaces de usuario.
- c) Ajuste de parámetros de herramientas especializadas en tareas específicas, para las funcionalidades a la medida del objeto de aprendizaje.
- d) Mixturas de las anteriores que implican herramientas para integración y manipulación de componentes, herramientas para la creación de componentes y herramientas para adecuar parámetros de productos especializados.

Teniendo en cuenta esta clasificación, se deben hacer análisis de alcance, tiempo, integración, costos, gestión de calidad, recursos necesarios, criterios de comunicación, gestión de riesgos y gestión de suministros.

Con respecto al equipo de desarrollo del proyecto, es indispensable la definición de los siguientes roles:

- a) Líder del proyecto, con competencias en ingeniería del software.
- b) Usuario del proyecto, con competencias en la temática del objeto de aprendizaje y con conocimientos sobre el nivel educativo de actuación del proyecto.
- c) Asesor pedagógico, competente para ilustrar sobre la intencionalidad pedagógica y la usabilidad didáctica del objeto.
- d) Técnicos expertos en herramientas de software y plataformas para hospedaje de objetos.

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

El desarrollo es en equipo, con pausas suficientes para racionalizar el dialogo entre estos actores. Sin demérito de la utilización de las tecnologías de la información y la comunicación, debe procurarse mayores tiempos para el diálogo cara a cara y la comunicación directa y pedagógica entre los miembros del equipo.

Con respecto al soporte, deben quedar compromisos documentales y contratos, donde al usuario se le garantice el soporte técnico del objeto creado, la corrección de errores y fallos, las posibilidades de expansión del objeto, la documentación de los procesos y la arquitectura del objeto y los manuales sobre el funcionamiento.

5.5. Experimentación y prueba de MESOVA

La metodología propuesta se experimentó en el proyecto *Diseño y despliegue de objetos de aprendizaje basados en problemas, para la formación en áreas específicas de ingeniería informática*, realizado en la Fundación Universitaria Católica del Norte, entre junio de 2009 y junio de 2011. Como resultado se integraron 13 objetos virtuales de aprendizaje con aplicación al área de programación de computadores.

Los objetivos de este proyecto fueron:

- Identificar las características de los aprendizajes en las áreas específicas de la ingeniería informática, en ambientes soportados por las TIC.
- Definir características de problemas asociados con la tipología de saberes conceptuales, procedimentales y propositivos de la ingeniería de sistemas e informática, para el desarrollo de objetos de aprendizaje.
- Diseñar objetos de aprendizaje, aplicando conceptos de UML² para el desarrollo de sistemas y aspectos metodológicos de aprendizaje basado en problemas.
- Diseñar un modelo de ciclo de vida para la creación de objetos de aprendizaje, bajo la metodología de aprendizaje basado en problemas.
- Diferenciar procesos de aprendizaje basado en problemas, de otros procesos secuenciales de aprendizaje, dentro del contexto de la ingeniería de sistemas e informática.

² Unified Modeling Language.

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

El producto final de este proyecto integró los siguientes módulos:

Módulos	Número de objetos
Representación de variables	1
Entrada y salida de datos	1
Asignación de expresiones	1
Condicionales	1
Ciclos <i>Para</i>	1
Ciclos Mientras	1
Vectores – Creación y llenado –	1
Suma de vectores	1
Matrices – Creación y llenado –	1
Suma de matrices	1
Ordenamiento por el método de la burbuja	1
Búsqueda secuencial en un vector	1
Búsqueda binario en un vector	1

Tabla 6. Módulos del proyecto.

El proyecto se orientó de acuerdo con los siguientes criterios y medios:

Temática	Lógica para programación de computadores.
Nivel educativo	Formación profesional en ingeniería informática.

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

Intencionalidad pedagógica	Aprendizaje basado en problemas - ABP
Tecnología utilizada	<ul style="list-style-type: none"> • Programa <i>Camtasia Studio 5.0</i> para la edición de vídeo. • <i>Dream Weaver</i> y <i>Flash CS 3.0</i>. <i>Action Script, html, Xml, Java Script</i> y <i>Photoshop</i>, para el diseño y montaje de los objetos virtuales en la web. • Solución de problemas mediante lenguaje de programación <i>Java</i>, utilizando la versión 6.8 de la plataforma <i>Netbeans</i>. • La herramienta <i>Google Blogs</i> para la creación de los <i>blogs</i> de cada objeto.

Tabla 7. Criterios y medios del proyecto.

6. Conclusiones

Los objetos virtuales de aprendizaje (OVAS) son esencialmente soluciones de software y por ello su planeación, análisis, diseño, desarrollo, implementación y evaluación deben recoger técnicas, tecnología, métodos y nociones de la ingeniería.

La construcción de OVAS debe enfocarse como un trabajo interdisciplinario pero orientado desde la ingeniería, pues básicamente se trata es de producir tecnología para aplicar en procesos de aprendizaje.

Una característica esencial de los OA es su reutilización, es decir, su capacidad para ser adaptados a distintos contextos educativos, a través de la edición, sustitución o acomodo de sus componentes. MESOVA se ajusta a estas condiciones, ya que el desarrollo del objeto se hace por módulos separables, con posibilidad de aislarse del producto final, con la posibilidad de ser evaluables independientemente.

Acorde con lo anterior, la metodología MESOVA que se acaba de presentar reúne conceptos de la ingeniería del software, sobre la especificidad de unos

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

componentes básicos: principios, estructura de fases y actividades, puntos de control, gestión y experimentación y pruebas.

En una primera experimentación se aplicó para construir objetos de aprendizaje para el área de la programación de computadores y como trabajos futuros para su validación teórica y experimental se aplicará con otros tipos de objetos de aprendizaje para distintas, áreas posiblemente de las matemáticas y las humanidades.

7. Lista de referencias

Abud, F. (2009). MeISE: Metodología de Ingeniería de Software Educativo. *Revista Internacional de Educación en Ingeniería*, 2 (1).

ADL - Advanced Distributed Learning. (2004). SCORM® Overview Version 1.3.1.

Amó, A. y Martínez, N. (2005). *Introducción a la ingeniería del software*. Delta Publicaciones.

Amaro, S. y Valverde, J. (2007). *Metodologías ágiles*. Universidad Nacional de Trujillo. Facultad de Ciencias Físicas y Matemáticas. Escuela de Informática.

Aycart, D., Ginestá, M. y Hernández, M. (2007). *Ingeniería de Software en Entornos de SL*. Barcelona: Universitat Oberta de Catalunya..

Calero, C., Moraga, M. y Piatini, M. (2010). *Calidad del producto y proceso de software*. Editorial Ra-Ma.

Colombia Aprende. Ministerio de Educación Nacional. (s.f.). Banco Nacional de Recursos Educativos. ¿Qué es un objeto de aprendizaje? Recuperado de: <http://www.colombiaprende.edu.co/html/directivos/1598/article-99393.html>.

Galindo, M., Simó M., Prieto, J. y Segret, R. (2010). *Escaneando la informática*. Universitat Oberta de Catalunya.

Galvis, P., Gómez, C. y Mariño, D. (1998). Ingeniería de software educativo con modelaje orientado por objetos: un medio para desarrollar micromundos interactivos. *Informática Educativa*, 11 (1), 30.

Kioska.Net. (2008). *Métodos rápidos (RAD, XP)*. Recuperado de: <http://es.kioskea.net/contents/genie-logiciel/>.

Ministerio de Educación Nacional. (2006). *Concurso de méritos: objetos virtuales de aprendizaje*. Bogotá: MEN.

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

Molpeceres, A. (2002). *Procesos de desarrollo RUP, XP y FDD. AT Javahispano*. Recuperado de: <http://www.javahispano.org/contenidos>.

Nextel. (2011). *Control del ciclo de vida del desarrollo software*. Recuperado de: <http://www.nexteleng.es/ingenieria/ibmrational.asp>.

Orjuela, A. y Rojas, M. (2008). La metodología de desarrollo ágil como una oportunidad para la ingeniería del software educativo. *Revista Avances en Sistemas e Informática*, 5 (2).

Ossandón, Y., y Castillo, P. (2006). Propuesta para el diseño de objetos de aprendizaje. *Revista Facultad de Ingenierías*, 14 (1), p. 36-48.

Sierra, J. (2005). *Estudio de la influencia de un entorno de simulación por ordenador en el aprendizaje por investigación de la Física en Bachillerato*. Ministerio de Educación y Ciencia de España.

Sommerville, I. (2005). *Ingeniería del software*. Pearson Educación.

Stephens, A. (2004). *Implementación de un estudio de caso usando learning objects para determinar la interoperabilidad entre diferentes plataformas e-learning*. Medellín: Universidad Eafit.

Weitzenfeld, A. (2005). *Ingeniería de software orientada a objetos con UML, Java e Internet*. Cengage Learning Editores.