

Estrategia didáctica virtual en Procesos de diseño de software

Nayibe Ruíz Chagüi
Universidad Industrial de Santander
ruiznayibe@gmail.com

Luis Carlos Gómez Flórez
Universidad Industrial de Santander
lcgomezf@uis.edu.co

Sonia Gamboa Sarmiento
Universidad Industrial de Santander
scgamboa@uis.edu.co

Resumen

El trabajo de investigación muestra cómo a través de un *dispositivo* computacional incorporado dentro del entorno virtual de aprendizaje de distribución libre *moodle*, se implementó la estrategia de *portafolios*, para un curso de Ingeniería del software I, del programa de pregrado de ingeniería de sistemas e Informática de la Universidad Industrial de Santander (UIS) - Colombia.

La estrategia de *portafolios* permite a los estudiantes identificar habilidades, estrategias y debilidades, para comprender su aprendizaje y potenciar habilidades de *modelado* y *representación*, haciendo uso de las TIC. Al profesor le permite hacer seguimiento y realimentación de las evidencias que realizan los estudiantes en el diseño de un proyecto softwar

El programa de Ingeniería de Sistemas e Informática de la UIS, cuenta con un *dispositivo* computacional incorporado en el VLE de distribución libre *moodle*, el cual implementa la estrategia didáctica de *portafolios*; para apoyar procesos de diseño del curso de Ingeniería de Software I.

La estrategia didáctica virtual “*portafolios*” es un mecanismo de realimentación y seguimiento que permite tanto al profesor como el estudiante validar y reflexionar sobre los procesos de aprendizaje. Además los estudiantes responden de manera favorable a propuestas de aprendizaje en entornos que son apoyados por las TIC.

Palabras clave Ingeniería de software, modelado, didáctica, *blended learning*, entorno virtual de aprendizaje.

Introducción

En ingeniería de sistemas una de las formas para dar solución a un problema es a través de la construcción de *software*; esto requiere hacer diferentes abstracciones del problema, tomar de la realidad lo relevante y plasmarlo en modelos a través de la *representación*; por otra parte la construcción de *software* implica realizar diferentes etapas, tales como la planificación, la cual abarca la obtención de requerimientos, análisis y diseño de la solución, actividades que requieren procesos de *modelado*. En el mercado existen herramientas privativas y de uso libre que apoyan el proceso de *modelado* pero estas no garantizan que se realice una adecuada *representación* de la realidad.

La presente propuesta de investigación busca contribuir al área de conocimiento de *diseño de software* propuesta por el SweBOK¹ (IEEE, 2004, pág. 2), a través de la

¹ Guide to the Software Engineering Body of knowledge

informática educativa, la estrategia de *portafolios* y la didáctica en ingeniería haciendo uso de las tecnologías de información y comunicación –TIC–.

SITUACIÓN DE INTERÉS

En 1968 durante la primera conferencia de la OTAN² se tocó por primera vez el término *crisis del Software* el cual hacía referencia a la gran dificultad de construir programas sin errores, verificables sin mencionar las demoras en tiempos de entregas y la insatisfacción de los usuarios; esto dio lugar a que por primera vez se hablará de *Ingeniería de Software*. Esta es una disciplina cuyo objeto principal es el diseño y construcción de Software. El desarrollo de software comprende dos complejidades inherentes a este (Weitzenfeld, 2005, pág. 13), por una parte está la complejidad del problema que tiene que ver con la funcionalidad del sistema, es decir a mayor número de requerimientos o funcionalidades que deba ofrecer la aplicación, mayor será el tamaño del sistema; por otra parte está la complejidad del diseño del sistema el cual debe satisfacer las funcionalidades mencionadas anteriormente y da como resultado el software, el cual es la solución a dicha problemática, con el diseño de un sistema software busca disminuir la complejidad del problema; por otra parte se tiene que para algunas problemáticas los requerimientos son cambiantes, lo cual hace que la complejidad aumente.

Otro aporte es el realizado en 1975 por Frederick Brooks (Weitzenfeld, 2005, pág. 16), sobre los inconvenientes para desarrollar software; a través del ensayo “*No Silver Bullet*”³, él menciona los desafíos que debe enfrentar la *Ingeniería de software*, debido a los inconvenientes generados por los elementos inherentes y abstractos del software; así como los generados en el desarrollo del mismo, ya que son productos intangibles.

² Organización del Tratado del Atlántico Norte

³ Frederick Brooks. “No Silver Bullet”, Memorias de la Décima Conferencia en computación Mundial. IFIP, editada por H.J. Kugler, 1986, pág.69-76.

Brooks propuso una regla empírica que permite estimar el tiempo para el desarrollo de un proyecto software, el cual se distribuye así: para la planeación se estima un $\frac{1}{6}$ de tiempo, $\frac{1}{6}$ para codificación, $\frac{1}{4}$ para realizar pruebas de componentes, $\frac{1}{4}$ para pruebas del sistema. (Ver figura 1), es decir la mitad del tiempo se invierte en actividades no esenciales como lo son las pruebas; si a esto le sumamos la codificación, otra actividad no esencial, se estaría dando $\frac{2}{3}$ de tiempo a actividades no esenciales y se está dedicando sólo $\frac{1}{3}$ a la planificación que es una actividad esencial para el desarrollo del proyecto.

Figura 1. Tiempo estimado para desarrollar un proyecto software



Fuente: Adaptado de (Weitzenfeld, 2005, pág. 18)

La crisis del software y el reto de construir software de buena calidad y sin errores, ha motivado que se implementen herramientas, metodologías, estándares, entre otros; en aras de resolver la problemática planteada, lo cual dejó de ser una preocupación en gran parte de la industria y pasó a condicionar el conocimiento.

Es en ese momento, donde los estamentos educativos empiezan a tomar parte en la solución de dicha problemática y el problema dentro de la disciplina *ingeniería de software* pasa a verse como una situación de interés dentro de la enseñanza de la misma.

¿QUÉ SE HA HECHO?

Los esfuerzos en producción de software han sido apalancados por herramientas, estándares, metodologías de desarrollo y lenguajes de programación, los cuales cada vez facilitan el nivel de abstracción y las tareas al programador, ver tabla 1.

Tabla 1. Nuevos paradigmas y Tecnologías

| Objetivo | Paradigma | Tecnología |
|--|---|--|
| Aumentar productividad | Desarrollo Dirigido por Modelos (DDM) | MDA – Model-driven Architecture ... |
| Mejorar integración y tiempo de respuesta | Orientación a Servicios (SOC) | SOA – Service-oriented Architecture ... |
| Mejor adaptación a las necesidades y cambios de las empresas | Orientación a los Procesos de Negocio (BPM) | BPMS – Business Processes Management Systems |

Fuente: (Ruiz, 2007, pág. 3)

A pesar de los avances en los lenguajes de programación, los problemas a resolver cada vez son más complejos y requieren de profesionales con habilidades no sólo en el área de programación sino en las áreas de planificación y diseño.

Según Zapata Jaramillo profesor de la universidad Nacional de Medellín, el cual ha realizado varios trabajos conducentes a mejorar los procesos de enseñanza y aprendizaje en *Ingeniería de software* (Zapata Jaramillo, 2007, págs. 3-12), muestra que a pesar de la evolución de la disciplina, su enseñanza, no ha tenido cambios significativos; esta se ha enfocado a la explicación de los contenidos teóricos, de forma expositiva por parte del profesor y a la realización de proyectos de clase por parte de los estudiantes, que distan de las características y de la magnitud de una situación ubicada en un contexto real; además de ser una enseñanza centrada en el profesor, ha sido reevaluada por no favorecer el desarrollo de habilidades de planeación y diseño en el futuro profesional.

Según Fairley (Fairley, 1986, págs. 39-52), las habilidades demandadas por la industria y que debe poseer un profesional del área de *software* son especiales, ya que debe tener conocimientos de ingeniería y de diferentes disciplinas, tales como ciencias de la computación, sistemas; tecnologías de la información; habilidades de administración, planeación y programación, una mezcla difícil de garantizar por parte de las universidades. Para solventar esta situación se han incorporado dentro de la enseñanza estrategias tales como, la realización de convenios universidad-industria, bajo la figura de *prácticas* o *pasantías*; el uso de simuladores y video juegos dentro de las clases; estudios de caso; y otras estrategias para el desarrollo de habilidades cognitivas particulares en el estudiante tales como los juegos no tecnológicos, la educación personalizada, los laboratorios, entre otros (Zapata Jaramillo, 2007).

Estamentos académicos y organizaciones internacionales como ACM⁴ y la IEEE⁵ (2001) jalonan la definición del cuerpo de la disciplina *Ingeniería de software*, inicialmente el currículo estaba ligado a la disciplina *Ciencias de la computación*, a partir de ACM (2005) la *Ingeniería de software* es definida como una disciplina independiente.

Cabe resaltar los esfuerzos por contribuir al cuerpo de conocimiento del *Software Engineering Institute* de *Carnegie Mellon University*, donde se exploró el currículo que debe hacer parte de la enseñanza de la *Ingeniería de Software*, especificando las habilidades que deben primar en el ingeniero de *software* (Bagert *et al.*, 1999). Este estudio es retomado por Budgen y Tomayko en el 2005 para contrastarlo con la manera como se enseña la *Ingeniería de software* en la actualidad. Kitchenham *et al.* a través de una encuesta buscaron determinar la forma como el currículo de *Ingeniería de Software* se adapta a las necesidades de las organizaciones, encontrando que se requiere profesionales no sólo con habilidades matemáticas sino del área de administración y comunicación, así como el aporte realizado por Cowling, el cual era incluir el

⁴ Association Computing Machine

⁵ Institute of Electrical and Electronics Engineers

modelamiento como una área de conocimiento importante dentro del currículo de formación.

¿CUÁL ES LA VISIÓN?

El auge de las TIC ha cambiado notablemente la manera en que se realizan las actividades, las formas para comunicarse, el lenguaje utilizado, las nuevas maneras de hacer negocios, el manejo de las transacciones, el volumen de datos e información, el almacenamiento y procesamiento de los mismos, así como los procesos educativos (Gamboa S, 2010).

El estudiante juega un papel diferente en el proceso educativo, y la universidad no puede ser ajena a ello. Así, las TIC, tales como la Internet, las redes sociales, las plataformas que apoyan el aprendizaje (*e-learning*) configuran, en la actualidad, el panorama de las herramientas que apoyan los procesos de enseñanza y aprendizaje; por una parte el profesor pasa a tener un rol de diseñador de ambientes de aprendizaje, a ser su guía y facilitador; y por otra parte el estudiante pasa a tener un rol más activo como protagonista de dichos procesos.

Se entiende por formación, enseñanza en línea o *e-learning* al espacio donde profesores y estudiantes a través de la participación de forma remota, de internet y del uso de las bondades de las TIC, logran un ambiente educativo interactivo. El termino *e-learning* ha sido más utilizado en el entorno empresarial donde ha sido usado para el entrenamiento de los empleados en las organizaciones y en el ámbito educativo se utiliza el término virtualización o educación virtual (Ruipérez, 2003, pág. 18).

Estos son algunos de los beneficios proporcionados por el *e-learning* (Rosenberg, 2002, págs. 30-31).

- Reduce costos, tales como gastos de desplazamiento, al no requerir de una infraestructura física.

- Permite trabajar con un gran volumen de usuarios conectados virtualmente de forma simultánea, lo cual lo hace una solución altamente escalable que no requiere de una gran inversión.
- Cuenta con elementos multimediales como textos, imágenes, animaciones, vídeos, sonido que motivan a los usuarios en la realización de sus actividades.
- Al estar soportado por internet permite fácilmente la actualización y acceso a contenidos y actividades; logrando que la información sea coherente, oportuna, confiable y disponible en cualquier momento.
- Promueve el trabajo en equipo ya que cuenta con herramientas de comunicación entre los usuarios y permite construir comunidades de práctica, lo cual también contribuye al aprendizaje organizacional.
- Permite hacer seguimiento a los estudiantes, a través de actividades de formación planificadas y pueden ser realimentadas.

Por otra parte la formación presencial permite la interacción física y creación de vínculos, factor motivante en los estudiantes; además ofrece la posibilidad de realizar actividades complejas que se dificultarían de forma virtual, estos son algunos factores que no han permitido el éxito total del *e-learning*, como lo muestran estudios realizados por APeL⁶ donde se muestra que las instituciones de educación superior han reducido costos en proyectos relacionados, ya que las expectativas de matrículas de usuarios y finalización de los cursos no se han cumplido (cf. Bartolomé, 2004, pág. 2).

Debido a lo anterior aparece el término “*blended learning*” o *b-learning*, cuyos orígenes son del campo de la Psicología escolar y existen otros antecedentes como lo son el de “Enseñanza asistida por computador” en inglés (CAI) o “Aprendizaje basado en el computador” en inglés (BCL) (Bartolomé, 2004, pág. 5).

Como lo señala (Brodsky, 2003) *blended learning* no es un concepto nuevo. Durante años se ha estado combinando las clases magistrales con los ejercicios, los estudios de caso,

⁶ Asociación de proveedores de e-learning

juegos de rol y las grabaciones de vídeo y audio, así como el asesoramiento y la tutoría; también se le conoce como “Modelo Híbrido” (Marsh, 2003)

El *b-learning*; consiste en una educación semipresencial; esto significa que un curso incluye tanto momentos presenciales como actividades de *e-learning*. Este modelo de formación hace uso de las ventajas de la formación presencial y de la formación en línea; este deberá incluir tanto actividades en línea como presenciales, las cuales deben estar bien estructuradas, de modo que se facilite alcanzar los objetivos de aprendizaje propuestos.

El uso de ambientes o entornos virtuales de aprendizaje (*Virtual Learning Environment – VLE–*) cada vez es mayor, también son conocidos como LMS – *Learning Management System*, Sistema Gestor de Aprendizaje, Ambiente Virtual de Aprendizaje (AVA) o Espacio virtual de aprendizaje y corresponden a un espacio donde se llevan a cabo actividades involucradas en el proceso de enseñanza y aprendizaje. Cuentan con servicios que permiten gestionar cursos y usuarios, hacer seguimiento del aprendizaje, generar informes, realizar evaluaciones, foros de discusión, charlas, videoconferencias, pizarras electrónicas, entre otros; estos elementos permiten diseñar estrategias didácticas pensadas para el aprendizaje de temáticas específicas y con unos propósitos claros, apoyados por las TIC y guiados por la experiencia del profesor, además hace que se cuente con la posibilidad de que los estudiantes desarrollen las competencias necesarias para los cuales fueron diseñados (Corredor Montagut, 2010, pág. 31). En este proyecto el término utilizado es Entorno Virtual de Aprendizaje (VLE).

EL MODELADO EN EL DISEÑO DE SOFTWARE

En el diseño de software se requiere comprender el dominio del problema, dar solución a este y comprender el ambiente en que operará el sistema. Por lo tanto modelar es la acción que permite hacer una descripción lo más exacta posible del sistema y de las actividades llevadas a cabo en él. “Un modelo es una descripción de la estructura genérica

y del significado de un sistema” (Rumbaugh, Jacobson, & Booch, 2000, pág. 17). Un modelo es una *representación* abstracta de un sistema que permite comprender y hacerse preguntas acerca de este, es decir, un modelo es una abstracción a cierto nivel (Bruegge & Dutoit, 2010, pág. 6), que corresponde a una *representación* simplificada de la realidad, un ejemplo de ello es el globo terráqueo, el cual es la manera más exacta de representar al planeta Tierra, pero es menos práctico que un mapa, en caso de ser el soporte a la geografía.

Uno de los primeros pasos para resolver un problema es la creación de una *representación* del problema. La *representación* según (Chi, P.J., & R., 1981, pág. 121) es "una estructura cognitiva que corresponde a un problema dado...construido por el solucionador sobre la base de conocimientos relacionados con el dominio y su organización". La *representación* que se crea se basa en la percepción inicial que tiene la persona respecto al problema que desea resolver.

Herbert Simon en su libro “Las ciencias de lo artificial” (Simon, 2006, pág. 158) muestra un ejemplo para resolver el juego “*scrabble* numérico”, en el que existen dos jugadores cada uno con nueve cartas, cada jugador debe colocar las cartas mirando hacia arriba e ir en cada turno eliminando una carta, el ganador será el primero que complete la suma de quince, con tres de sus cartas; una estrategia para este juego es pensar en un cuadrado mágico donde hay números del uno al nueve, cada fila, columna o diagonal formada por una terna de números da quince, ver la figura 2, entonces al cambiar de *representación* se hace más fácil el juego ya que para muchos el juego de “tres en línea” es más conocido.

Figura 2. Estrategia del cuadrado mágico

| | | |
|---|---|---|
| 4 | 9 | 2 |
| 3 | 5 | 7 |
| 8 | 1 | 6 |

Fuente: Adaptado de (Simon, 2006), pág. 158

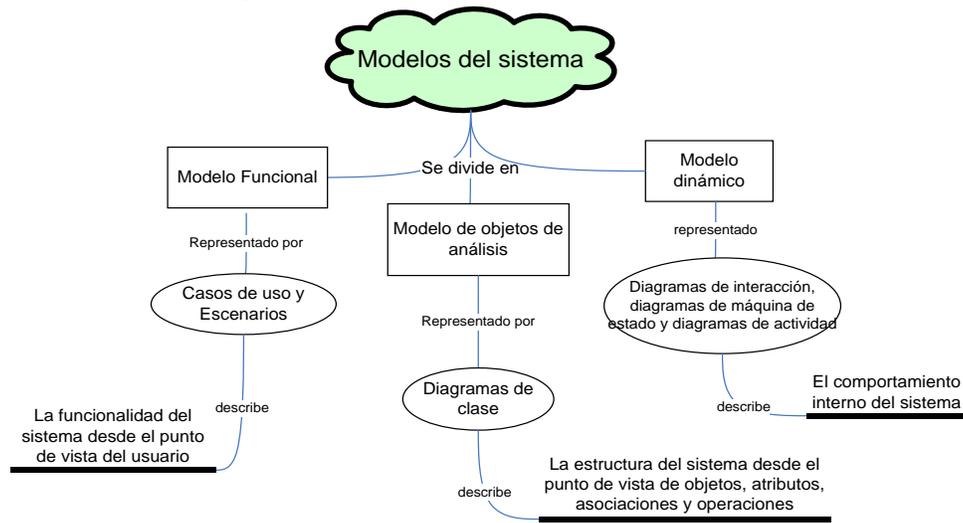
El *modelado* toma los aspectos relevantes de la realidad para resolver un problema e ignora los aspectos no esenciales, a esta capacidad para “encapsular y aislar la información del diseño y ejecución” se le llama *abstracción*.

La *abstracción* es esencial para diseñar software, es la clasificación de los fenómenos en conceptos. El *modelado* es el desarrollo de abstracciones que sirven para contestar preguntas específicas acerca de un conjunto de fenómenos. Esta ha tenido una evolución desde la aparición de los lenguajes de programación; algunos de los mecanismos utilizados para hacer abstracción son los procedimientos o funciones, los módulos, los tipos abstractos de datos y los objetos.

Los problemas pueden ser descritos a través del lenguaje natural, de las matemáticas, formalismos clásicos del álgebra, la geometría, la teoría de conjuntos, el análisis o la topología, entre otros, si dichos problemas corresponden a objetos físicos se pueden representar mediante planos, mapas, dibujos de ingeniería, prototipos o modelos tridimensionales; si los problemas están relacionados con actividades o acciones pueden representarse con diagramas de flujo, flujos de procesos, entre otros; un sistema software puede ser representado a través del lenguaje de modelado UML, este es el resultado de la unificación de la técnica de modelado de objetos (OMT) de Rumbaugh (1991), Booch (1994) e *Ingeniería de software orientada a objetos* (OOSE- Jacobson, 1992) y tiene como objetivo proporcionar una notación estándar que puede ser usada por todos los métodos orientados a objetos (Bruegge & Dutoit, 2010, pág. 30).

El desarrollo de un sistema puede ser visto en tres modelos diferentes como se muestra en la figura 3.

Figura 3. Modelos de un sistema software



Los diagramas UML para la representación de dichos modelos son.

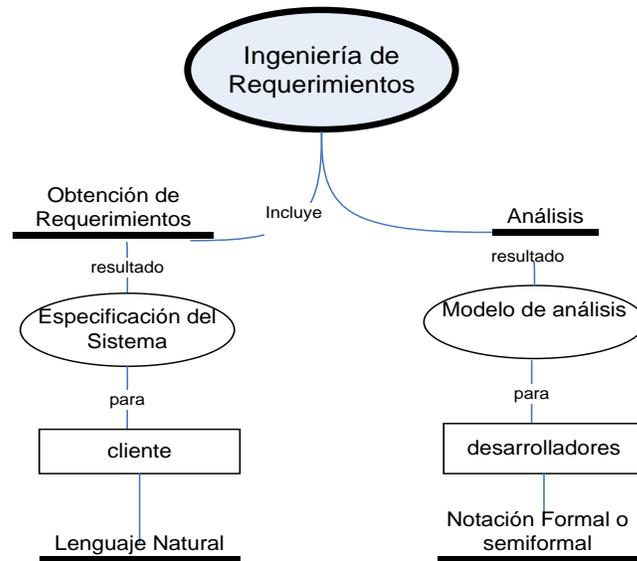
- **Diagramas de caso de uso:** Representan la funcionalidad del sistema desde el punto de vista del usuario.
- **Diagramas de clase:** Representan la estructura estática del sistema en términos de objetos, sus atributos y relaciones.
- **Diagramas de interacción:** Representa el comportamiento del sistema y visualiza la comunicación entre objetos.
- **Diagramas Máquina de estado:** Describe el comportamiento dinámico de un objeto individual como un número de estados y las transiciones entre estados.
- **Diagramas de actividad:** Describe el comportamiento del sistema en términos de actividades.

Las actividades de desarrollo de software, buscan manejar la complejidad mediante la construcción de modelos de los dominios del problema o del sistema y estas son: *Obtención de requerimientos, Análisis, Diseño del sistema, Implementación y Pruebas.*

A continuación se describen algunos aspectos importantes de las tres primeras actividades que son pertinentes dentro de la presente investigación ya que estas permiten modelar un conjunto de componentes y las relaciones entre estos, lo cual da una visión general denominada *Arquitectura del sistema.*

• **Obtención de requerimientos.** Los requerimientos son las características y restricciones que posee el sistema que se desea construir y que satisfacen las necesidades del cliente a esto se le llama *Ingeniería de requerimientos* y se puede dividir en dos actividades (Ver figura 4).

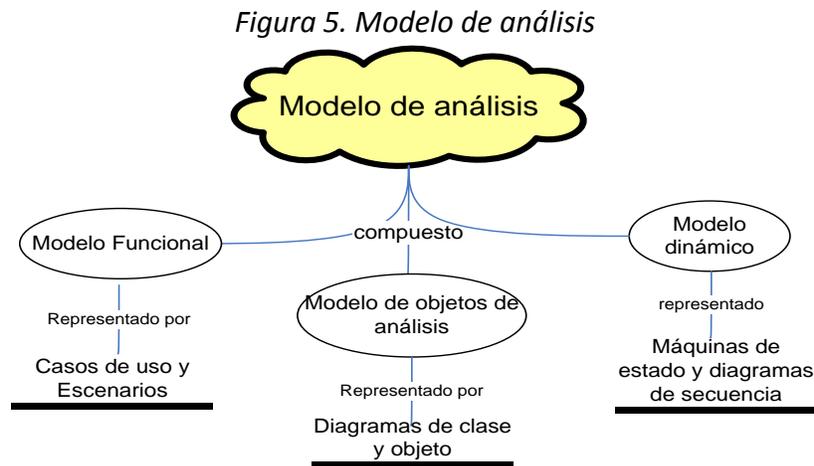
Figura 4. Ingeniería de Requerimientos



La *obtención de requerimientos* da como resultado la especificación del sistema, es la más compleja debido a que requiere de varios participantes, tanto usuarios como clientes tienen diferentes niveles de conocimiento sobre el sistema, para solventar lo anterior usan los *escenarios* y los *casos de uso*. El *análisis* da como resultado un modelo de análisis para los desarrolladores.

• **Análisis.** Busca obtener un modelo del sistema que pretende ser correcto, completo, consistente y sin ambigüedades. Aquí se formaliza la especificación de requerimientos y se examina en más detalle las condiciones de frontera. Los desarrolladores en el *análisis orientado a objetos* construyen un *modelo* que describe el dominio de la aplicación. El *modelo de análisis* describe el sistema por completo desde el punto de vista de los actores

y sirve como la base de la comunicación entre el cliente y los desarrolladores. Está compuesto por tres modelos ver la figura 5.



- **Diseño del sistema.** Durante esta etapa los desarrolladores transforman el *modelo de análisis* en un *modelo de diseño* del sistema tomando los requerimientos no funcionales. Busca tener información acerca de la estructura interna del sistema, definir los objetivos de diseño del proyecto y descomponer el sistema en subsistemas.

El *modelo de diseño* refleja las estrategias para la construcción del sistema, los subsistemas y la correspondencia entre el hardware y el software.

El ingeniero de sistemas se ocupa de proponer *modelos* o *representaciones* del mundo real a partir de las cuales se planteen sistemas de información o de gestión de conocimiento que satisfagan situaciones problemáticas.

En este contexto, el concepto de *representación* en procesos de desarrollo de software tendría que ser aprendido, no solamente como un conocimiento propio de la *ingeniería del software*, sino como una habilidad que debe ser desarrollada. El mercado ofrece herramientas informáticas que permiten hacer *modelado* bajo los estándares de

representación que ofrece la notación UML⁷, pero su uso eficiente, la elaboración de diagramas UML, requiere de que quien los realiza tenga tanto conocimientos como habilidades de *modelado* que deben ser aprendidas en los espacios académicos diseñados para este fin, de manera que un estudiante frente a una situación problemática, lleve a cabo una adecuada *representación* de dicha situación que pueda ser plasmada en un modelo, el cual describe la situación en ese lenguaje simbólico.

DESCRIPCIÓN DE LA PROPUESTA DE INVESTIGACIÓN

Esta propuesta de investigación se realiza en el marco de la maestría de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander –UIS– y del macroproyecto “Desarrollo y validación de una estrategia de aprendizaje basada en el uso de plataformas virtuales para el curso de ingeniería del software I”, financiado por la Vicerrectora de Investigación y Extensión de la UIS.

Esta investigación se ubica en el campo de la informática educativa, y tiene como fin plasmar la estrategia didáctica de *portafolios* en un dispositivo didáctico computacional que apoye el aprendizaje del concepto de *representación* en procesos de diseño de software, necesario para llevar a cabo procesos de creación de software, usando como espacio la asignatura presencial “Ingeniería del Software I” del programa de pregrado de Ingeniería de Sistemas e Informática de la UIS.

La didáctica según (Brousseau, 1990, pág. 259) es usada para designar “el arte de enseñar”, es decir, el conjunto de medios y procedimientos que buscan conocer, saber algo, una lengua, un arte; su esencia es la reflexión y el análisis del proceso de enseñanza aprendizaje. Se ocupa del qué, el para qué y el cómo enseñar.

La universidad, como entorno social y centro que privilegia, no solo la producción de conocimiento, sino su enseñanza, es un espacio propicio para que se reflexione sobre el

⁷ Lenguaje de modelado unificado

conocimiento y se genere un valor agregado que contribuya al desarrollo social y económico de la sociedad. Uno de los propósitos en la *didáctica* es pensar las maneras en que el conocimiento científico pueda ser enseñado; según (Vargas Guillén G. G., 2008, págs. 51-59) en la *condición postmoderna* el diseño de estrategias didácticas consiste en diseñar ambientes de aprendizaje: espacios en los que profesores y estudiantes interactúen con elementos pensados para favorecer el aprendizaje de temas específicos; además se dan dos cambios, por una parte se va de la enseñanza al aprendizaje y por otra parte del profesor instructor al orientador (Vargas G, Gamboa S, & Reeder, 2008, pág. 18). La estrategia *didáctica* planteada en la investigación está dispuesta a que los estudiantes adquieran habilidades para realizar abstracciones que le permitan representar la realidad en modelos coherentes a una solución informática.

Cuando se habla de *dispositivo*, el DRAE⁸ lo define, como un adjetivo que indica “que dispone” y como sustantivo que indica “mecanismo, aparato, artefacto, órgano, elemento o artificio de un sistema, dispuesto para producir una acción prevista”. Un *dispositivo* es un *objeto tecnológico* y eso denota dos cosas: que es materialmente real y que se puede interactuar con él; y lo tecnológico implica que responde a una construcción basada en una metodología probada, sistemática y aceptada como válida con el fin de que resuelva un problema específico o actúe de una manera específica en una situación dada. No existe un dispositivo general, todo depende de la disciplina, de las temáticas específicas, del nivel de los estudiantes y de la experiencia del profesor.

Según ;Error! No se encuentra el origen de la referencia. el término *dispositivo* se puede tomar como un artefacto tecnológico que está ahí, dispuesto porque se requiere para cumplir un fin específico con criterios de eficiencia, vigencia, practicidad, estética, costos, entre otros.

Para Estela Quintar el uso del concepto *dispositivo* “es un artificio que tiene como función articular, conectar, poner en movimiento sistemas orgánicos, materiales o simbólicos”. Lo

⁸ Diccionario de la Real Academia Española

cual le da un grado de complejidad y su función es la de “*bisagra* entre mundos simbólicos y/o sistemas materiales u orgánicos”, lo cual le da el carácter de unir y colocar en acción, es decir es un activador de procesos subjetivos e intersubjetivos (Salcedo, 2009).

El *dispositivo* está incorporado en el VLE de distribución libre *moodle* en un hosting público de propiedad del grupo de investigación y desarrollo en Sistemas y Tecnologías de la Información –STI– al cual pertenecen los autores de la propuesta.

El *dispositivo computacional* incorpora la estrategia didáctica de *portafolios*, la cual es un mecanismo de realimentación y seguimiento que permite tanto al profesor como el estudiante validar y reflexionar sobre los procesos de aprendizaje (Klenowski, 2005) .

La estrategia de *portafolios* se define como “[...] una colección de trabajos del estudiante que nos cuenta la historia de sus esfuerzos, su progreso y logros en un área determinada. Esta colección debe incluir la participación del estudiante en la selección del contenido del portafolio, las guías para la selección, los criterios para juzgar méritos y la prueba de su autorreflexión” (Artes y Spandel, 1992, pág. 36). Es utilizado en diferentes ámbitos a nivel cultural, artístico, arquitectura, entre otros; en el campo educativo es usado para la evaluación, para procesos de promoción de profesores y en la parte vocacional para la evaluación por competencias.

Entre las ventajas de la estrategia de *portafolios* se encuentra la de permitir a los estudiantes reflexionar sobre su propio proceso de aprendizaje, identificar habilidades, estrategias y debilidades que les permiten ser autorreguladores de dicho aprendizaje de una forma continua, es decir, les proporciona herramientas cognitivas para comprender su propio aprendizaje; a los profesores les permite llevar un seguimiento y hacer realimentación a las evidencias que realizan los estudiantes. Es una estrategia que no esta exenta de inconvenientes como son el trabajo y tiempo de dedicación a la valoración, revisión y realimentación por parte del profesor; así como no permitir fácilmente obtener una apreciación generalizada de todo un grupo de estudiantes.

Es necesario, entonces, diseñar ambientes didácticos en los cuales se propicie el aprendizaje tanto de conceptos como de habilidades que permita a los estudiantes realizar abstracciones adecuadas y plantear modelos que correspondan a situaciones reales.

El ambiente de aprendizaje permite engranar la pedagogía, la didáctica y las TIC; para el diseño de la estrategia *didáctica* se han tomado los principios de la teoría de aprendizaje constructivista que busca que el estudiante construya su propio conocimiento; los planteamientos de Piaget según los cuales, el pensamiento responde a la formación de unas estructuras mentales de cada persona, las cuales van equilibrando su conocimiento, a través de su evolución cronológica, esto permitió caracterizar los estudiantes a los cuales va dirigido el curso; los planteamientos de Vygotsky, tales como la *zona de desarrollo próximo*, donde el estudiante adquiere su conocimiento a través de su experiencia y con la ayuda de otros; y los postulados de Ausubel que busca que los estudiantes relacionen los conocimientos previos con la nueva información y puedan lograr un *aprendizaje significativo*.

El proyecto software es planteado y definido al inicio del curso con un propósito de aprendizaje, este es llevado a cabo a lo largo del curso y es desarrollado en grupos de trabajo. Los estudiantes pueden modificar, agregar elementos y recursos complemento que se requieran elaborar para las evidencias solicitadas, tales como encuestas, entrevistas, guías, entre otros; además el estudiante cuenta con material complementario dentro del VLE y los heurísticos propuestos por (Bruegge & Dutoit, 2010) que apoyan los proceso de diseño de software, cada evidencia tiene su respectivo registro y el responsable de la tarea realizada, para ello se ha establecido una guía para el nombre y formato de los recursos, ver tabla 2.

El *portafolios* permite al estudiante llevar evidencias de las actividades de diseño del proyecto software, las cuales conforman la *Arquitectura del Sistema*. Cada etapa está incorporada a través de pestañas, donde el estudiante registrar las evidencias, así como

modificar y realizar los diagramas correspondientes a cada actividad (Ver figura 6 y 7). Las evidencias solicitadas al estudiante, corresponden a guías, plantillas documentos o modelos y diagramas propuestos en las figuras 4 y 5.

Tabla 2. Formato de nombres de archivos para las evidencias

| Dato | Identificador | Formato | Ejemplo |
|--------------------------------------|----------------------|---|---|
| Guías, plantillas, formas | GUIA | GUIA_Nombre_Fecha | <i>GUIA_Requerimientos_02-09-11.docx</i> |
| Informes de avance del equipo | INFO | INFO_Nombre_Fecha_v# (v indica la versión) | <i>INFO_Requerimientos_03-10-11_v1.docx</i> |
| Diseños, modelos, diagramas | DIAG | DIAG_Nombre_Fecha_v# | <i>DIAG_Requerimientos_03-10-11_v1.vsd</i> |
| Revisión por parte del docente | REVI | REVI_Nombre_Fecha_v# | <i>REVI_Requerimientos_05-10-11_v1.docx</i> |

Figura 6. Interfaz Principal del Estudiante

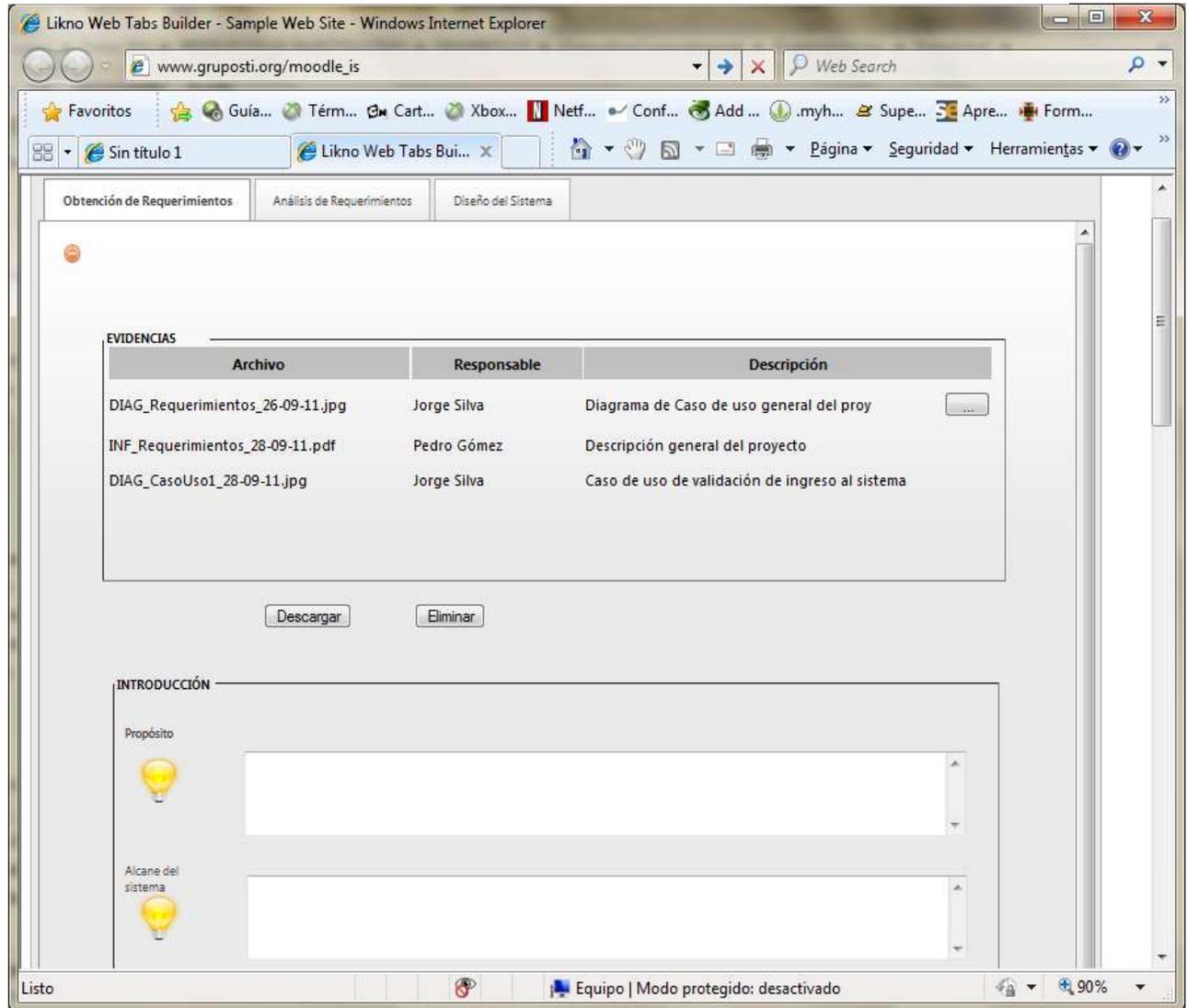
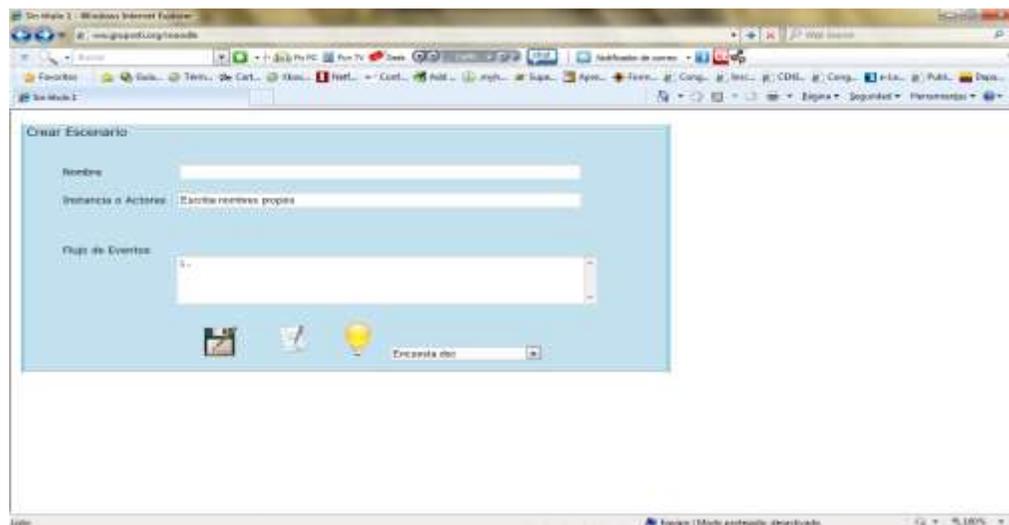


Figura 7. Interfaz Crear Escenario



El dispositivo permite el trabajo en equipo y la realimentación por parte del profesor durante todo el desarrollo del proyecto (Ver figura 8).

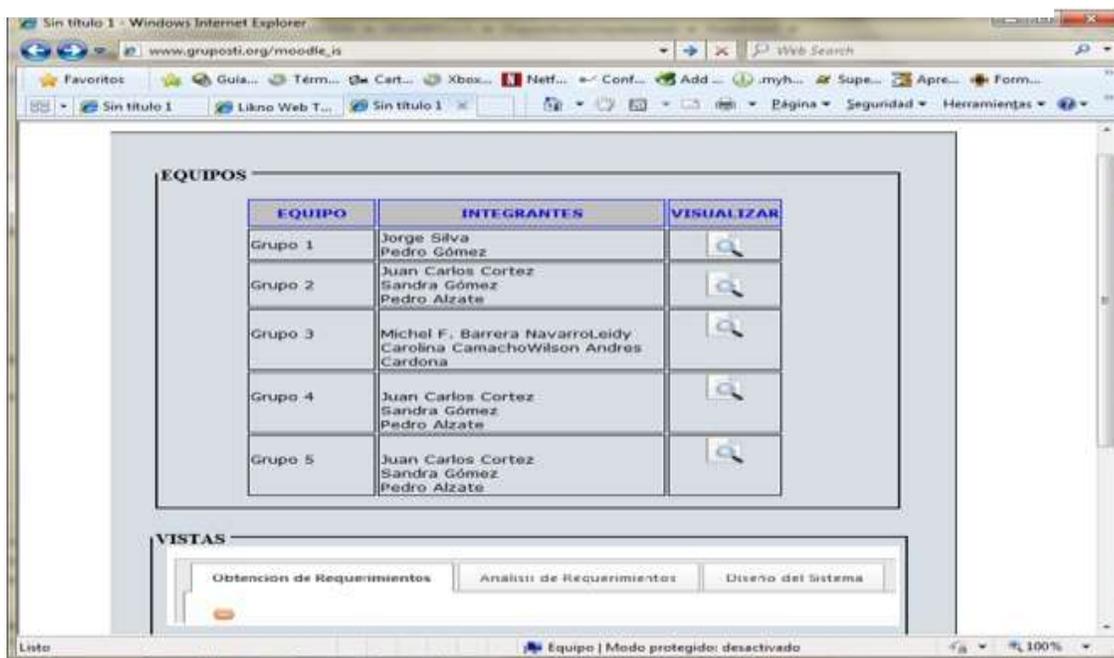


Figura 8. Interfaz del profesor

Además dentro de la estrategia didáctica virtual se contempla el uso de las bondades y recursos propios del VLE, como son el *wiki*⁹ y foros de discusión para la construcción de documentos y reflexión del proceso realizado, esto fomenta las habilidades de comunicación, lectura y escritura en los estudiantes; la descarga y subida de archivos, con el fin de contar con material complementario; los cuestionarios para realizar diferentes tipos de actividades evaluativas; encuestas para realimentar cada una de las etapas del *portafolios*, entre otras.

Conclusiones

La implementación de la estrategia de *portafolios* en un dispositivo computacional contribuye a que:

Los profesores cuenten con una gama de herramientas y recursos para realizar su labor de forma gestionada, en todos los momentos del proceso: la socialización, la evaluación y realimentación del conocimiento.

El profesor sea un gestor de la enseñanza y un guía en el aprendizaje. A su vez permitirá que se mejoren las prácticas docentes y se puedan realizar aportes en el área de la pedagogía.

- Los estudiantes encuentren en el dispositivo un ambiente familiar –la tecnología– con una variedad de recursos para lograr su aprendizaje. Por su parte las TIC les ofrecen fácil acceso a gran cantidad de información y los VLE cuentan con elementos tales como foros, wikis o chats, que fomentan en ellos habilidades comunicativas de redacción, lectoescritura y comprensión, así como el manejo de la argumentación y el discurso usando un lenguaje técnico propio del saber que se está aprendiendo.
- Los estudiantes de Ingeniería de Sistemas e Informática que cursan Ingeniería del Software I cuenten con una herramienta que contribuye al aprendizaje del concepto de *representación* en procesos de diseño de software, permitiendo que ellos puedan

⁹ Sitio Web que puede ser editado por varios usuarios

proponer, confrontar y realimentar sus modelos con otros y llevar la teoría a la práctica beneficiando su perfil profesional.

- A través de este proyecto la Escuela de Ingeniería de Sistemas e Informática y la maestría en Ingeniería de Sistemas e Informática de la UIS, contribuyan a los campos temáticos de la informática educativa y la pedagogía.
- El desarrollo de esta experiencia permitirá al grupo de investigación STI continuar trabajando en proyectos similares a nivel de pregrado y maestría para la aplicación y estudio del uso de las TIC en procesos de enseñanza y aprendizaje en ingeniería y ser referente a la comunidad universitaria.

Lo significativo de la investigación no está en el desarrollo del dispositivo computacional, sino que esta se basa en el diseño de una estrategia didáctica cuyo propósito está centrado en implementar una valiosa estrategia de enseñanza y aprendizaje como lo es el *portafolios* para la temática de *representación* en procesos de diseño de software haciendo del uso de las TIC. A su vez, el desarrollo de la investigación busca profundizar en el área de la informática educativa y espera que la divulgación de esta, muestre un panorama y algunas características a tener en cuenta en proyectos similares que busquen apoyar cursos de pregrado que trabajan en modalidad presencial apoyados por las TIC.

Bibliografía

- [1]. Bartolomé, A. (2004). Blended Learning. Conceptos básicos. *Píxel-Bit. Revista de Medios y Educación* , 23, 7-20.
- [2]. Brousseau, G. (1990). Qué pueden aportar a los enseñantes los diferentes enfoques de las didácticas de las Matemáticas. *Enseñanza de las Ciencias* , 259-267.
- [3]. Bruegge, B., & Dutoit, A. (2010). *Object-oriented Software Engineering* (Tercera ed.). Mexico: Pearson Education.

- [4]. Calzada Prado, F. J. (2010). *Repositorios, Bibliotecas Digitales y CRAI. Los objetos de aprendizaje en la educación superior*. Buenos Aires: Alfagrama.
- [5]. Chi, M., P.J., F., & R., G. (1981). Categorization and representation of physics problems by experts and novices. *Cognitive Science*, 5 (2), 121-152.
- [6]. Corredor Montagut, M. A. (2010). *Enseñanza en línea: otra opción para la formación en el ámbito universitario*. Bucaramanga: Publicaciones UIS.
- [7]. Fairley, R. (1978). Educational Issues in Software Engineering. En: Proceedings of the 1978 annual conference ACM/CSC-ER. Washington, D.C.
- [8]. Fairley, R. (1986). *The role of Academe in Software Engineering Education*. En: *Proceedings of the 1986 ACM fourteenth annual conference on Computer science*. Cincinnati.
- [9]. Gamboa S, S. (2010). La dimensión argumentativa de la formación. *Revista Itinerario Educativo*, 65-95.
- [10]. IEEE. (2004). *SMBOK - Guide to the Software Engineering Body of knowlegde*. California.
- [11]. Joyanes Aguilar, L. (2000). *Programación Orientada a Objetos*. Madrid: McGrawHill.
- [12]. Klenowski, V. (2005). *Desarrollo de portafolios. Para el aprendizaje y la evaluación*. Madrid: Narcea.
- [13]. Marsh, G. E. (2003). *Blended Instruction: Adapting Conventional Instruction for Large Classes*. Recuperado el 26 de Junio de 2011, de Journal of Distance Learning Administration, Number IV: [disponible]:<http://www.westga.edu/~distanc>
- [14]. Rosenberg, M. (2002). *E-learning. Estrategias para transmitir conocimiento en la era digital*. Bogotá: McGrawHill.
- [15]. Ruipérez, G. (2003). *Educación Virtual y eLearning*. Madrid: Fundación Auna.
- [16]. Ruiz, F. (2007). La Enseñanza de la Ingeniería del Software en el marco del Espacio Europeo de Educación Superior . *Ponencia en el II Congreso Español de Informática (CEDI)*, 1-20.
- [17]. Rumbaugh, J., Jacobson, I., & Booch, G. (2000). *El lenguaje unificado de modelado. Manual de Referencia*. Madrid: Pearson Education.
- [18]. Salcedo, A. (2009). Pedagogía de la potencia y didáctica no parametral: Entrevista con Estela Quintar. *Revista Interamericana de Educación de Adultos -Aula Magna*, 1-5.
- [19]. Simon, H. A. (2006). *Las ciencias de lo artificial*. Granada: COMARES.

- [20]. Vargas G, G., Gamboa S, S., & Reeder, H. P. (2008). *La humanización como formación*. Bogotá: San Pablo.
- [21]. Vargas Guillén, G. (2003). El disponer. *Folios, no. 11. DCS-UPN, Departamento de Ciencias Sociales, Facultad de Humanidades* . Bogotá, Colombia: Universidad pedagógica Nacional. Recuperado el 25 de Julio de 2011, de http://w3.pedagogica.edu.co/storage/folios/articulos/folios11_04arti.pdf: <http://profesorvargasguillen.wordpress.com/2011/06/>
- [22]. Vargas Guillén, G. G. (2008). Didáctica en la condición postmoderna. De las competencias a la cooperación. *Revista Científica Guillermo de Ockham* , 51-59.
- [23]. Weitzenfeld, A. (2005). *Ingeniería de Software orientada a Objetos con UML, Java e Internet*. México: Thomson.
- [24]. Zapata Jaramillo, C. M. (2007). *Los juegos de clase no tecnológicos como una estrategia didáctica para la enseñanza de la ingeniería del software*. Medellín, Colombia: Universidad Nacional.