

# SOFTWARE EDUCATIVO PARA FAVORECER LA APREHENSIÓN DE LOS CONTENIDOS DE INGENIERÍA DE SOFTWARE

## EDUCATIONAL SOFTWARE TO FACILITATE THE APPREHENSION OF THE CONTENTS OF SOFTWARE ENGINEERING

*Zenoyda Lujo Aliaga, Arianna Pérez Céspedes, Libely Victoria Cedeño Galindo, Carmen María Batista Díaz*

Facultad de Ciencias Técnicas, Departamento de Ingeniería Informática. Universidad de Las Tunas. Cuba  
E-mail: [zlujo, ariannapc, lcedeno, carmenbd]@ult.edu.cu

(Enviado Mayo 09, 2017; Aceptado Mayo 26, 2017)

### **Resumen**

Tomando como partida la consulta de materiales de varios autores y el diagnóstico realizado a los diferentes *softwares* educativos existentes para el apoyo a la docencia en la Facultad de Ciencias Técnicas de la Universidad de las Tunas, se realiza la siguiente investigación con el objetivo fundamental de: desarrollar un sistema de objeto de aprendizaje en condición de tutorial que, integrado desde la plataforma *Moodle*, permita favorecer la aprehensión de los contenidos de la asignatura Ingeniería de *Software* en el Tercer año de la carrera de Ingeniería Informática. Como resultado de esta investigación se obtiene un sistema tutorial, sustentándose en los referentes teóricos de la comunicación, del aprendizaje semipresencial, la didáctica y el diseño curricular. El mismo se desarrolló con el empleo del Sistema de Gestión de Contenidos *Drupal*, utilizando código *Javascript* para implementar algunas de las funcionalidades y la plataforma interactiva *Moodle* para el sistema de evaluaciones.

**Palabras clave:** *Ingeniería de Software, Moodle, Tutorial, Software Educativo.*

### **Abstract**

Taking as a starting point the consultation of materials of several authors and the diagnosis made to the different existing educational softwares for the support of teaching in the Facultad de Ciencias Técnicas of the University of Las Tunas, the following research is carried out with the fundamental objective of: develop a learning object system in a tutorial condition that, integrated from the Moodle platform, allows to favor the apprehension of the contents of the Software Engineering subject in the third year of the Computer Engineering degree. As a result of this research, a tutorial system is obtained, based on the theoretical referents of communication, blended learning, didactics and curricular design. It was developed with the use of the Drupal Content Management System, using Javascript code to implement some of the functionalities and the interactive Moodle platform for the evaluation system.

**Keywords:** *Software Engineering, Moodle, Educative Software.*

## 1 INTRODUCCIÓN

La carrera de Ingeniería Informática exige la formación de profesionales capaces de dar solución a cualquier problema de esta rama que se presente en la sociedad, lo que requiere del estudio, iniciativa y habilidades que adquieran los estudiantes en el trayecto de la misma. Muchas son las formas de enseñanza que se preparan, de modo tal, que propicien un eficiente desarrollo de la actividad intelectual de los estudiantes y oriente a los docentes como vía de estimulación del pensamiento creador. En este sentido cobra auge la introducción de las

Tecnologías de la Información y las Comunicaciones (TIC).

Las TIC, han sido conceptualizadas como la integración y convergencia de la computación microelectrónica, las telecomunicaciones y la técnica para el procesamiento de datos. Sus principales componentes son: el factor humano, los contenidos de la información, el equipamiento, la infraestructura material, el *software* y los mecanismos de intercambio electrónico de información, los elementos de política y regulaciones y los recursos financieros. Se encargan del estudio,

desarrollo, implementación, almacenamiento y distribución de la información mediante la utilización de *hardware* y *software* como medio de sistema informático. Uno de los campos donde han incursionado positivamente es en la educación, elemento clave en la construcción de la sociedad basada en la información, el conocimiento y el aprendizaje. Educar, en la Sociedad de la Información, significa mucho más que capacitar a las personas para el uso de las tecnologías de la información y comunicación [1].

En correspondencia con esto, en la Universidad de las Tunas se trabaja por incorporar las facilidades y ventajas del uso de las TIC en la vida cotidiana, en la formación de profesionales, en las investigaciones científicas, entre otras actividades que contribuyan a elevar el nivel técnico y humano, además, que promueven la creación de una cultura en su utilización. El plan de estudio de esta carrera tiene concebido la existencia de varias disciplinas, las cuales agrupan la totalidad de las asignaturas de la carrera.

Una de estas asignaturas que se imparten en dicha carrera es la Ingeniería de *Software*, columna vertebral en la formación de ingenieros, pues constituye uno de los principales espacios donde el estudiante puede aplicar en la práctica productiva, el resultado obtenido durante el proceso de enseñanza-aprendizaje.

En consideración a lo anterior, en la Facultad de Ciencias Técnicas de la Universidad de las Tunas, se realizó un estudio en la asignatura Ingeniería de *Software* a un grupo de estudiantes, pues además de lo antes expuesto, se evidenciaban algunas limitaciones de los mismos, al desempeñarse como desarrolladores de *software* a partir de la realización de las prácticas laborales y sus tesis de grado. Con dicho estudio se arribó a la conclusión de que evidentemente existían limitaciones por parte de estos en:

- Realizar el análisis de las clases en procesos de negocio de complejidad media.
- En la interpretación de los artefactos de diseño para estos casos.
- La generación de código de los métodos de las clases en tales condiciones.
- 

Lo que en otras palabras se puede generalizar en que los estudiantes presentan problemas al implementar las clases de análisis y diseño siguiendo el proceso que brinda la Ingeniería de *Software*.

Todas estas deficiencias permitieron plantear el siguiente objetivo: Desarrollar un *software* educativo en condición de tutorial que, integrado desde el *Moodle* (entorno virtual de aprendizaje utilizado en la Universidad de las Tunas para promover el aprendizaje cooperativo permitiendo la participación de estudiantes y docentes), permita favorecer la aprehensión de los contenidos de la asignatura Ingeniería de *Software* a los estudiantes del tercer año de la carrera de Ingeniería Informática en la Universidad de las Tunas, para aplicarlos en los

respectivos proyectos investigativos y de desarrollo de *software* a los que pertenecen.

## 2 DESARROLLO

### 2.1 Las TIC en el proceso de enseñanza-aprendizaje

En la actualidad no se puede hablar de educación, ni de cultura, ni de desarrollo si no se aplican eficientemente y en todas las esferas de la sociedad las TIC.

Las TIC se encargan del estudio, desarrollo, implementación, almacenamiento y distribución de la información mediante la utilización de *hardware* y *software* como medio de sistema informático [2].

Desde la perspectiva del aprendizaje elevan el interés y la motivación haciendo que los estudiantes dediquen más tiempo a trabajar y, por tanto, aprendan más, manteniendo una interacción continua de la actividad intelectual.

Dentro de estas tecnologías juegan un papel fundamental en la educación los medios de enseñanza y recursos del aprendizaje, considerando como *medios* aquellos que han sido diseñados para ser utilizados en los procesos educativos y como *recursos* los que tienen otros propósitos [3]. Un ejemplo de lo anterior expresado es la aparición y utilización del *software* educativo.

En los procesos de enseñanza y aprendizaje apoyados por el uso de las TIC se conoce el *software* educativo o programas informáticos orientados a fines educacionales a: aquellos que sean utilizados para apoyar o facilitar los diferentes procesos presentes en los sistemas educacionales.

Son disímiles las definiciones existentes sobre *software* educativos las cuales son tratadas según el contexto en el que se esté trabajando.

Pere Marqués lo clasifica en Programas tutoriales, Base de datos, Simuladores, Constructores, Programas Herramientas [4]:

**Programas tutoriales:** Son programas que en mayor o menor medida dirigen, tutorizan, el trabajo de los alumnos. Pretenden que, a partir de informaciones y la realización de ciertas actividades previstas de antemano, los estudiantes pongan en juego determinadas capacidades y aprendan o refuercen unos conocimientos y/o habilidades.

**Bases de datos:** proporcionan un entorno estático y facilitan la exploración y consulta selectiva.

**Simuladores:** proporcionan un entorno dinámico y facilitan la exploración y la manipulación del entorno.

**Constructores:** son entornos programables. Facilitan a los usuarios unos elementos simples con los cuales pueden construir elementos más complejos u otros entornos.

**Programas herramientas:** Son programas que proporcionan un entorno instrumental con el cual se facilita la realización de ciertos trabajos generales de tratamiento de la información: escribir, organizar, calcular, dibujar, transmitir, captar datos.

Los tutoriales son sistemas instructivos de autoaprendizaje que pretenden simular al maestro y muestran al usuario el desarrollo de algún procedimiento o los pasos para realizar determinada actividad [5]. Por este motivo, es mejor seguir los tutoriales en su secuencia lógica para que el usuario entienda todos los componentes. Cuando los tutoriales se limitan a proponer ejercicios de refuerzo sin proporcionar explicaciones conceptuales previas se denominan programas tutoriales de ejercitación.

Después de analizadas las clasificaciones anteriores, las autoras proponen como solución para su investigación la realización de un programa tutorial que permitirá en gran medida desarrollar didácticamente las habilidades de analizar, diseñar e implementar *software*, teniendo en cuenta los conocimientos adquiridos en la asignatura Ingeniería de *Software* a partir de este producto, de esta forma fomentar el proceso de enseñanza-aprendizaje en los estudiantes del tercer año de la carrera de informática, así como servir de bibliografía en las clase.

## 2.2 Desarrollo del tutorial

Para el desarrollo del *software* se tuvieron en cuenta cinco etapas. Una **inicial** donde se realizó una descripción completa del objeto de estudio, seleccionando aquellos contenidos que iban a ser presentados y la estructura de presentación de los mismos, describiendo además las herramientas que se utilizarían, el público al que va dirigido, los objetivos pedagógicos que se pretenden cumplir y el presupuesto necesario. La segunda etapa utilizada fue el **análisis y diseño**, en la misma se describen las funcionalidades a implementar y se define la organización interna del producto (directorios, archivos, interfaces de usuarios, apariencia, colores y arquitectura que se desea). La tercera etapa de desarrollo fue la **Construcción** en la que se implementaron todas las funcionalidades del tutorial. La cuarta etapa fue la de **Pruebas**, fase de vital importancia, ya que es donde se evalúa la efectividad del tutorial, realizando pruebas constantes e instrumentos que permitan la detección de errores para lograr un producto de calidad. Por último la etapa de **Despliegue y mantenimiento** que es donde se instala el *software*, se realiza una capacitación al personal que lo utilizará y se proponen las vías para la constante actualización del mismo.

La plataforma de desarrollo escogida para la elaboración del producto fue Drupal, un sistema de gestión de contenidos (CMS), que permite la creación y administración de contenidos principalmente en páginas Web.

Para las dos primeras etapas se tuvieron en cuenta tres aspectos que para las autoras son de vital importancia:

### a. La información, la sustancia del Software

Lo primero realizado fue la selección de los temas a tratar, los cuales fueron ubicados dentro del Plan y Programas de Estudios vigente de la asignatura Ingeniería de *Software*, para lo cual se hizo un análisis exhaustivo de los contenidos dados en cursos anteriores en dicha materia, así como búsquedas y revisiones bibliográficas de artículos científicos publicados en internet que dan al traste con los temas que se abordan en el mismo, de igual forma fueron revisadas todas las informaciones por los profesores que imparten la asignatura en la facultad, haciendo una selección detallada de aquellas que serían de mayor utilidad para crear los temas, quedando finalmente estructurado el tutorial en cuatro temas:

1. Proceso de Desarrollo de *Software*.
2. Análisis de sistemas.
3. Diseñando *software*.
4. Generando Código.

En cada uno se comienza con una breve introducción y como parte del desarrollo se le muestran ejemplos de lo explicado, para que les sea de mayor facilidad adquirir los conocimientos.

Posteriormente se determinó el camino a seguir para realizar el diseño, el cual cuenta con una estructura flexible en cuanto al diseño funcional, abierto en el aspecto de que el profesor pueda manipular datos que se manejan en dicho *software*, a fin de adecuarlo más a la temática en la que hará uso del mismo.

### b. Presentación de contenido

Para la presentación del contenido se analizó la estrategia para el desarrollo de de *software* educativo diseñada por Elsa Verónica González Robles del Departamento de Tecnología Educativa de la Secretaría de Educación Pública y Cultura del Estado de Sinaloa, quien plantea que en esta segunda parte del diseño es necesario concentrar procedimientos que necesiten de la habilidad y destreza de los alumnos, establecer un reto para manejar los conceptos tratados en el *software* [6].

Es necesario incluir juegos, imágenes llamativas, premios, llevar un registro de los mejores usuarios. Siguiendo esta estrategia los temas incluyen páginas en las que el contenido aparece en formato de texto, aunque en algunos casos fue necesario utilizar otro tipo de formato como: vídeo, gráficos, animaciones, y esquemas.

Se desarrollaron interfaces sencillas, con colores claros, logrando así que no existan elementos más importantes que el contenido que se expone.

### c. Las armas para el docente

Como un complemento para el mejor uso del tutorial terminado, es necesario proporcionar estrategias de cómo incorporarlo dentro de la planeación de una clase. Con esto se logran despejar algunas dudas de los profesores y en otros se despertará el interés de utilizarlo e

incorporarlo en su práctica, y por qué no posteriormente ser partícipes en la creación de nuevos diseños.

En el caso particular del *software* en cuestión, se buscó dentro del programa analítico de la asignatura en qué sección del Curso en el *Moodle* iba a ser insertado, teniendo en cuenta las particularidades de los contenidos que se modelarán en el *software* y que el estudiante necesita hacer mayor hincapié. Esta plataforma educativa al ser interactiva le permite al profesor realizar la evaluación de lo aprendido en el tutorial, a través de cuestionarios y tareas.

A continuación se muestra la página principal del tutorial.



Figura 1 Página Principal del tutorial.

La arquitectura del mismo es muy sencilla delimitada por cuatro áreas, las cuales se explican a continuación:

**d. Banner**

El banner está ubicado en la parte superior del *software* Fig. 2 y muestra a primera vista el título del contenido principal que se quiere transmitir a partir de una imagen estática tipo logo.

**e. Menú principal**

Está ubicado en el lateral izquierdo como se muestra en la Fig. 3 y se le domina *principal* porque es donde se ubicaron los cuatro temas fundamentales bajo los cuales se estructuró todo el contenido a tratar de forma coherente y jerárquica; llevando al estudiante a seguir una secuencia lógica de acceso al conocimiento brindado y la dependencia que existe entre los núcleos que lo componen.

**f. Menú superior**

El Menú superior se encuentra situado debajo del banner, en el mismo se visualizan los siguientes vínculos: Inicio, Mapa conceptual, Ejercicios, Videos, Animaciones, Ejemplos y Glosario de términos (Fig. 4). Este menú persigue el objetivo de minimizar el tiempo de búsqueda del estudiante de una determinada información y amplía las funcionalidades del tutorial.



Figura 2 Banner del tutorial de Ingeniería de Software.

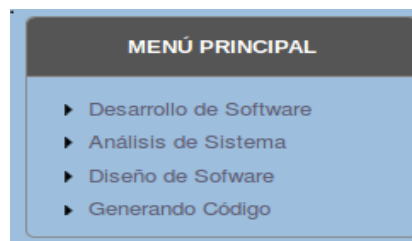


Figura 3 Menú Principal.



Figura 4 Menú superior del Tutorial.

En la página *Inicio* se da la bienvenida a los estudiantes al tutorial y se presentan los aspectos que se tratarán en el mismo.

Se usó el **Mapa Conceptual** para la representación gráfica de todo el contenido a través de nodos, garantizando de esta forma que el estudiante pueda navegar por todo el tutorial y remitirse al tema que sea de su interés de forma rápida.

En el vínculo **Videos** los estudiantes podrán encontrar videos que le muestran los pasos a tener en cuenta para la realización de los diferentes modelos a construir en el proceso del desarrollo de *software* usando la herramienta de modelación Visual Paradigm. Además se incluyen teleclases de algunos de los temas que se muestran en el tutorial.

Otros de los vínculos mostrados es **Animaciones**: aquí el estudiante podrá encontrar diferentes animaciones que muestran la representación de modelos usando la herramienta de modelación *Rational Rose*. De esta tendrá varias vías por las que pueda modelar.

En **Ejercicios** se enuncian varios ejercicios y cuestionarios que hacen vínculos al *Moodle* y que serán evaluados por el profesor desde el Entorno Virtual de Aprendizaje.

El **Glosario de términos** es una página con las terminologías y conceptos que pueden ocasionarles dudas a los estudiantes.

Por último en el vínculo *Ejemplos* se muestran ejemplos de ejercicios resueltos, con la finalidad de que el estudiante tenga una guía que le sirva en la realización de los ejercicios propuestos en cada tema y que posteriormente serán evaluados desde el *Moodle*.

### g. Área de Información

Esta sección se ubica a la derecha el menú principal y es donde se visualizan todos los contenidos. Es bueno resaltar el uso elementos denominados *las migajas de pan*, los cuales brindan al usuario en todo momento la ubicación del nodo donde se encuentra, aportando una mayor flexibilidad en la navegación.

En el tutorial desarrollado se utiliza código *Javascript* para implementar algunas de las funcionalidades de los módulos que integra el Drupal, como es el caso del *Jtooltips* (Fig. 5), módulo que se emplea para la explicación de conceptos de palabras claves dentro del cuerpo del *software*.

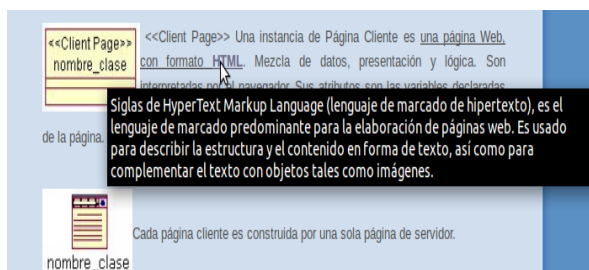


Figura 5 Uso de *Jtooltips*.

## 2.3 Orientaciones metodológicas de la aplicación

El tutorial propuesto fue creado como material de consulta para la asignatura de Ingeniería de *Software*. El hecho de estar integrado en la plataforma interactiva *Moodle* como un objeto de aprendizaje, le da la facilidad al estudiante de acceder en el horario que estimen conveniente desde cualquier puesto de trabajo.

A continuación se enuncian algunas orientaciones para un uso correcto del mismo, las cuales deben ser acogidas tanto por estudiantes y profesores en aras de lograr los objetivos previstos del *software* educativo.

Como se había planteado con anterioridad, en el tutorial se abordan cuatro temas por los cuales el estudiante debe transitar, teniendo en cuenta en todo momento una secuencia lógica.

### a. Desarrollo de *software*

En este tema se muestra un conjunto de conceptos, características, metodologías que lo introducen en el mundo del desarrollo de *software*, dotándolos de las habilidades que deben poseer para llevar a cabo dicho proceso. Donde el estudiante adquiere un conocimiento previo, permitiéndole hacer una lectura de la realidad y lo que determina qué información seleccionará, cómo la organizará y qué relaciones establecerá entre ellas.

En este caso se definen las etapas por las que debe transcurrir el desarrollo de un *software*, antes de llegar al Análisis, Diseño y Codificación, que son las etapas donde se hacen mayor hincapié en el tutorial, producto de las propias limitaciones detectadas.

Para enriquecer un poco más el tema, el profesor puede hacer uso de la bibliografía complementaria publicada en el *Moodle*, y utilizar los diferentes espacios que brinda este entorno (Foros, Cuestionarios, Tareas) y que fueron descritos con anterioridad para evaluar lo conocimientos adquiridos por los mismos.

### b. Análisis de sistemas

Una vez terminado el primer tema ya el estudiante debe estar capacitado para entender el proceso del *software* a partir de la Fase de Análisis, siendo este el próximo tema a tratar. Primeramente se enuncian los conceptos fundamentales que deben tener en cuenta para entender el proceso, luego a partir de un caso de estudio descrito se les explica los pasos a seguir para llevar a cabo la fase de análisis, mostrándoles los artefactos, trabajadores y principales modelos que se llevan a cabo en el proceso, haciendo mayor énfasis en los aspectos que dan lugar al diseño de clases.

Para poder entender todo este proceso pueden apoyarse en el ejemplo descrito, que con la ayuda de imágenes y videos publicados en el tutorial el estudiante podrá ejercitar lo aprendido.

La evaluación puede ser realizada por el profesor de la misma forma que en el tema I, no obstante cada tema tiene sus peculiaridades y por ende la forma de evaluarse es distinta, lo que conlleva que en cada caso estarán publicados vínculos a los ejercicios que les van a ser medidos.

### c. Diseñar *software*

Diseñar *software* es el tercer tema en cuestión, en este se comienza dando una panorámica de qué es el diseño y la importancia que tiene en el proceso de desarrollo de *software*. Se enuncian los principales conceptos y terminologías a utilizar. Para que el estudiante no pierda la secuencia, se trabaja con el mismo caso de estudio del tema anterior, tomando como partida el diagrama de análisis, de esta forma, pueden ir visualizando todo el proceso desde la etapa preliminar hasta donde se encuentran ubicados.

Al finalizar el tema los estudiantes deben ser capaces de conocer las particularidades de las diferentes clases que se usan en el diseño, seleccionar cuáles de estas son persistentes o no y la importancia que tiene en el proceso, de igual forma aprenderán las diferentes notaciones y estándares que brinda el Lenguaje Unificado de Modelado (UML) para la representación del modelo de diseño y cómo las actividades contempladas en el análisis dan lugar al diseño en la que, usando el lenguaje de los desarrolladores se refinan los requisitos y se estructuran en base a clases y paquetes.

#### d. Generando Código

El último tema a tratar es Generando código, para ello ya el estudiante debe haber aprendido a modelar los diferentes tipos de diagramas en las fases de análisis y diseño, lo que le da la secuencia lógica del contenido, de la misma forma deben saber reconocer el tipo de lenguaje de programación con el que están trabajando, por lo que se les explica los aspectos fundamentales que dan lugar a este proceso.

Para un mejor entendimiento de este tema, se publica un vínculo a la sección de videos donde se explica paso a paso cómo generar el código en la práctica, haciendo uso del Visual Paradigm: herramienta de código libre que sirve para el modelado de los diagramas que los estudiantes deben aprender a realizar con la ayuda del tutorial y que lo han usado en clases anteriores a estos temas descritos.

Al finalizar este tema los estudiantes deben haber adquirido todas las habilidades necesarias en la implementación de *software* y utilizar todos esos conocimientos en la actividad productiva.

#### 2.4 Validación de la efectividad del tutorial

Además de todas las técnicas utilizadas durante la fase de pruebas para encontrar errores y validarlo, una vez implementado y utilizado el tutorial en la facultad, se realizó la evaluación de la validez del producto mediante la aplicación de unos de los instrumentos utilizados en el diagnóstico inicial: la prueba pedagógica.

Dicha prueba fue aplicada a una muestra de 12 estudiantes del tercer año de la carrera, donde los resultados obtenidos permitieron hacer una comparación con los alcanzados inicialmente, observándose un cambio cualitativo referente a las deficiencias detectadas desde un principio, demostrando así un perfeccionamiento en el proceso de enseñanza-aprendizaje de la asignatura Ingeniería de *Software*.

Es un sistema seguro ya que al estar integrado a la plataforma *Moodle* como un objeto de enseñanza-aprendizaje sigue el protocolo HTTPS, que es una versión segura del protocolo HTTP, el cual implementa un canal de comunicación seguro basado en SSL (*Secure Socket Layers*) entre el navegador del cliente y el servidor HTTP.

Otros de los aspectos que lo hacen viable es que cuenta con una arquitectura que le permite al estudiante una mejor movilidad por el sistema sin necesidad de ir al inicio para llegar al contenido que desea.

Por otra parte no requiere de grandes conocimientos de informática para su uso, por lo que puede ser utilizado por cualquier persona.

El despliegue del producto es viable teniendo en cuenta que las condiciones tecnológicas están creadas en la facultad, lo que no quita que pueda ser desplegado en cualquier otro lugar, ya que solo se requiere de un

servidor Web Apache2 o superior, en el cual estará dispuesto el *Moodle* y el tutorial en cuestión.

Funciona como un medio bibliográfico disponible en la facultad, donde todos pueden acceder sin necesidad de realizar gastos en las cuotas de navegación de internet, de aquí la importancia que tiene desde el punto de vista económico.

El uso de componentes como los hipervínculos, logran dentro del *software* una conexión con otros programas o aplicaciones haciendo que se convierta en una aplicación interactiva.

Es una herramienta viable, a consideración de las autoras, debido a que toda la información y contenidos que se presentan fueron detallados y consultados por un grupo de profesores especialistas en el tema, de igual forma los materiales y bibliografías utilizadas son de fuentes confiables y están actualizadas.

### 3 CONCLUSIONES

Se logró concretar en una sola herramienta el proceso de análisis y diseño como un todo que era uno de los principales problemas detectados en el programa de la asignatura, de esta forma el estudiante a través de un único ejemplo puede visualizar el proceso de transformación de una etapa a la otra.

Su uso efectivo logra profesionalidad en los estudiantes ya que los dota de conocimientos que les sirven en la práctica para llevar a cabo los proyectos investigativos y productivos en la facultad, realizar su tesis de grado, prácticas profesionales en las empresas y desempeñarse profesionalmente en el desarrollo de *software* una vez graduado.

Contribuye al reforzamiento del estudio independiente en los estudiantes debido a los ejemplos y ejercicios que se proponen.

Por todo lo expresado anteriormente las autoras consideran que con el uso del *software* educativo implementado se le dio cumplimiento al objetivo propuesto en esta investigación.

### 4 REFERENCIAS

- [1] Piazza. Hacia la sociedad de la información. Guatemala, 2000.
- [2] UNESCO. Las tecnologías de la información y la comunicación (TIC) en la educación.  
URL:[http://www.unesco.org/new/es/unesco/themes/icts/\(10/08/2011\)](http://www.unesco.org/new/es/unesco/themes/icts/(10/08/2011)).
- [3] Cordero Escobar, I.; De la Paz Granados, M. E.; Cabeza Poblet, B. Odiseo, Revista Electrónica de Pedagogía: Influencia de la tecnología educativa en la enseñanza de la anestesiología, 4, 14(2010).  
URL:<http://www.odiseo.com.mx/bitacora-educativa/influencia-tecnologia-educativa-ensenanza-anestesiologia/>. (12/12/2011).

- [4] Marqués, P. El Software educativo. Universidad Autónoma de Barcelona. Barcelona, Estel, 2005.
- [5] SOMA 2007. Tutoriales. URL:  
<http://www.somacomunicaciones.com/WP/tutoriales/>.  
(13/10/2011).
- [6] González R. E; Ortiz, M. W. Estrategias para el diseño y desarrollo de software educativo. Departamento de Tecnología Educativa de la Secretaría de Educación Pública y Cultura del Estado de Sinaloa.  
URL:<http://bibliotecadigital.conevyt.org.mx/colecciones/documentos/somece/11.pdf>. (13/10/2011).