

Tecno Lógicas

ISSN: 0123-7799

tecnologicas@itm.edu.co

Instituto Tecnológico Metropolitano

Colombia

Guerrero-Peña, Diego Alejandro

Estrategias Didácticas y Plan de Actividades para el Diseño Curricular del Curso Básico de la
Enseñanza de la "Ingeniería del Software" a Partir del Proyecto SWEBOK

Tecno Lógicas, núm. 18, julio, 2007, pp. 187-219

Instituto Tecnológico Metropolitano

Medellín, Colombia

Disponible en: <http://www.redalyc.org/articulo.oa?id=344234311009>

- ▶ Cómo citar el artículo
- ▶ Número completo
- ▶ Más información del artículo
- ▶ Página de la revista en redalyc.org

ESTRATEGIAS DIDÁCTICAS Y PLAN DE ACTIVIDADES PARA EL DISEÑO CURRICULAR DEL CURSO BÁSICO DE LA ENSEÑANZA DE LA “INGENIERÍA DEL SOFTWARE” A PARTIR DEL PROYECTO SWEBOK

DIEGO ALEJANDRO GUERRERO PEÑA*

Resumen

Las instituciones de educación superior han tenido que evolucionar sus metodologías de enseñanza de la Ingeniería del Software, para gestar profesionales eficientes en el desarrollo de aplicativos informáticos. Este artículo pretende presentar una nueva didáctica y un diseño curricular de un curso de “Ingeniería del Software Básico”, basado en el modelo SWEBOK de la IEEE (Guide to the software engineering body of knowledge. Swebok) y adaptado al modelo por competencias presentes en el Modelo Pedagógico del INSTITUTO TECNOLÓGICO METROPOLITANO, como resultado del proyecto de investigación HICAP-IOI del ITM.

Palabras clave

Metodología de enseñanza aprendizaje, metodología de desarrollo de software, ingeniería del software, teoría de la actividad, teoría del caos.

* Ingeniero de sistemas, especialista en gerencia de proyectos, candidato a Magíster en Ingeniería Informática de EAFIT, docente investigador del ITM en asignaturas de la ingeniería del software y relacionadas.
diegoguerrero@itm.edu.co;
dguerrer@eafit.edu.co

Abstract

Higher education institutions have taken up new software engineering teaching methodologies in order to prepare professionals who are efficient in the development of informatics applications. This article aims at showing a new didactic, and a syllabus design for a course of "Basic Software Engineering", based on the SWEBOK model of IEEE (GUIDE TO THE SOFTWARE ENGINEERING BODY OF KNOWLEDGE SWEBOK), which is adapted to the model by competences which is applied in the ITM's Pedagogical Model as a result of the HICAP-IOI research project at ITM.

Key Words

Teaching Methodology, learning, software development methodology, software engineering, theory of activity, theory of chaos.

1. INTRODUCCIÓN

La educación ha evolucionado al mismo ritmo que lo ha hecho el hombre. Durante el último siglo, especialmente desde los años ochenta, han surgido cambios en la educación debido a las variaciones sociales, políticas y tecnológicas. Uno de los cambios más significativos ha sido el tecnológico informático, el cual nos reta constantemente a involucrar, o por lo menos a tener en cuenta, sus avances. Entonces, nos encontramos con una variedad de elementos pasados y presentes, tales como: las corrientes pedagógicas, el avance en el software y el hardware, y las metodologías que existen para el manejo de cada uno de estos elementos, que por su sinergia llaman a la unión de todos ellos para sacar el mejor provecho en cuanto a las nuevas metodologías de enseñanza aprendizaje.

Hoy tomamos temas como la ingeniería del software y cómo enseñarla, la propuesta pedagógica de Ausubel (En POZO, 1989)¹, y otros autores (COLON, 2002), (ANAYA y TRUJILLO, 2006), (VYGOTSKI, 1996) y las TIC² para lograr un aporte integral a la enseñanza de la ingeniería del software. Es decir, integrar estos elementos de tal manera que se logre un proceso de aprendizaje de la ingeniería del software más efectiva, donde los alumnos mejoren y proyecten esos logros en su futura labor.

En consecuencia, y teniendo en cuenta el modelo pedagógico del ITM (URREGO, 1999), se lleva a cabo el desarrollo del proyecto Herramienta informática como apoyo pedagógico a la asignatura Ingeniería del Software orientado a objetos -HICAP-IOI (GUE-RRERO, 2006a) que constituye un apoyo pedagógico importante en la asignatura Ingeniería del Software Orientada a Objetos, además facilita, por un lado, la realización de exámenes automáticos que ayudan a reforzar los conocimientos de los estudiantes; y, por el otro, a los docentes para elaborar evaluaciones teniendo en cuenta

¹ Interesa para este artículo solo La teoría del aprendizaje significativo de Ausubel, comentada por Pozo en el libro POZO, J.I. (1989).

² Tecnologías de Información y Comunicaciones.

los indicadores de logro del estudiante. Durante el desarrollo del proyecto (2005 - 2006), se evidenció la necesidad de replantear el diseño curricular de la asignatura Ingeniería del Software ISI54, en cuanto a las competencias, los indicadores de logro y la red de concepto, dando como resultado el nuevo diseño curricular para el 2007³ (GUERRERO: 2006) y la propuesta de nuevas estrategias didácticas y plan de actividades para ser implementadas en ISI54, las cuales se presentan en este artículo.

Por lo anterior, se replantea una propuesta más global y profunda del diseño curricular del curso básico de la enseñanza de la ingeniería del software, tanto en el contenido como en la metodología de enseñanza, que pretende tomar los elementos de la metodología de SWEBOK, un proyecto de la IEEE⁴ (IEEE: 2004), integrados en la propuesta realizada por Raquel Anaya y John Trujillo (ANAYA, Raquel y TRUJILLO, John: 2006) así como de la experiencia de más de 12 años en docencia del autor de este artículo. La didáctica planteada surge desde la perspectiva de la práctica del desarrollo de software, atendiendo a la situación actual de las empresas de software colombianas (ANAYA: 2005) adaptado al Modelo pedagógico del ITM.

Inicialmente, se presentan los postulados pedagógicos en los cuales se basa la propuesta del diseño curricular (diseño instruccional); luego se presentan dos mapas conceptuales, uno trata sobre los elementos que componen el modelo SWEBOK y el otro, define las estrategias didácticas aplicadas a la categoría "Productos del Software" del modelo SWEBOK mediadas por las TIC, con el fin de ubicar al lector con respecto a los componentes didácticos mediados por TIC, que se deben manejar en cada una de las actividades del diseño curricular; continúa con la explicación de la didáctica

³ Es imprescindible aclarar que ISI54 es la primera asignatura, o curso inicial, que trabaja el tema de la ingeniería del software en el programa académico Ingeniería de sistemas de información del ITM del 2007.

⁴ COMPUTER SOCIETY PROFESSIONAL PRACTICES COMITÉ, Disponible en www.swebok.org

general para ser manejada en las actividades de las sesiones de clase; luego, sigue con las estrategias de aprendizaje para ser utilizadas en las actividades. Como compendio de lo anterior, se presenta el plan de actividades del diseño instruccional del Curso inicial del proceso de enseñanza de la Ingeniería del Software.

Por último, se presentan las conclusiones y la propuesta para futuros desarrollos con el fin de mejorar los cursos similares y desarrollar nuevas estrategias didácticas, utilizando elementos interactivos y virtuales.

2. POSTULADOS PEDAGÓGICOS

A continuación se presentan, de manera sintetizada y como postulados, los principios pedagógicos que sustentan en gran parte, el trabajo realizado.

POSTULADO 1. La mediación efectuada por las TIC genera múltiples factores estructurales que rodean e influyen en la relación didáctica que se da entre el docente y el estudiante, y entre, éstos y el saber. La tecnología se constituye como una instancia para apoyar situaciones de gestión de clausuras explícitas o implícitas, que evolucionan en el mantenimiento constante del contrato de deberes y derechos entre quien enseña y quien aprende, acciones que se espera sirvan de ayuda para lograr una construcción más autónoma por parte del alumno (TRUJILLO y JARAMILLO: 2006, 98).

En el modelo pedagógico del ITM, el tiempo independiente es considerado fundamental para el trabajo de los estudiantes fuera de clase, por tal razón, utilizar las TIC es de vital importancia, puesto que facilitan y potencializan la aprehensión del conocimiento, a la vez que sirven como medios de apoyos para la comunicación entre los alumnos y el docente, siempre que permitan el descubrimiento individual de cada estudiante y su acercamiento al saber. Esta situación se advierte en la parte titulada “Descripción de las actividades sin acompañamiento”, del diseño curricular propuesto.

POSTULADO 2. En una siguiente construcción de correlaciones se reconocen⁵ las diferentes intervenciones que los estudiantes hacen con la tecnología para interactuar con el docente, sus compañeros y los contenidos, o hacer despliegue de sus propuestas ante los otros. En estos escenarios se marcan las diferencias entre la unidireccionalidad y la bidireccionalidad, la sincronía y la asincronía, la conectividad y la no conectividad a la red. Todas estas situaciones potencian procesos de apropiación, proyección, extrañamiento y reconocimiento alrededor del logro de expectativas individuales o colectivas, dentro o fuera de la intencionalidad curricular. (TRUJILLO y JARAMILLO: 2006, pp. 101-102)

Como el proyecto es desarrollado por un grupo de estudiantes, bajo la dirección del docente, se propone, entonces, exponerlo entre ellos mismos con el fin de compartir experiencias que permitan la apropiación y revisión del conocimiento adquirido por éstos, en el desarrollo del mismo, así como el adquirido por los compañeros. Esto lo pueden realizar en foros virtuales y/o presenciales con la coordinación del docente encargado.

POSTULADO 3. “Es el hombre el que debe enfrentarse a la realidad vacía; un hombre que es capaz de vivir sin ayudas, porque ya no las necesita. Un hombre que ante nada será el encargado de crear el mundo” (COLÓN: 2002, 87)⁶. Es el aprendiz el que debe construir sus propios conocimientos y el que es responsable de los mismos.

En la propuesta del diseño curricular o instruccional y de las estrategias de enseñanza y aprendizaje, se explicita la necesidad de permitirle al estudiante generar su propio conocimiento mediante la exploración del material bibliográfico, de las tecnologías informáticas disponibles y de las asesorías programadas por el docente. También se aclara la importancia de hacerle consciente al estudiante sobre la responsabilidad que tiene con su propio aprendizaje y con la utilización adecuada del tiempo independiente

⁵ En la figura 1, véase el Triángulo del aprendiz, contenidos y tecnología.

⁶ Sólo se analiza la Teoría del caos- Superhombre de Nietzsche.

para la construcción del conocimiento, asimismo, hacerle consciente que la adquisición del conocimiento es individual aunque las actividades se realicen en grupo.

POSTULADO 4. El contexto natural de ubicación de la Teoría del Caos es la Complejidad, de tal manera que sin situaciones complejas no se daría caos; por lo tanto, hablar de teoría del caos es hablar de la COMPLEJIDAD; por lo tanto, supone aceptar el desorden, la innovación y el movimiento como aspectos inherentes a cualquier situación caótica, por lo que la teoría del caos, indudablemente, se nos presenta como otro fundamento para la Teoría Social y por ende para la Teoría Educativa [...] (COLÓN: 2002, 102).

El estudiante debe enfrentar el desarrollo de un proyecto real, que dé solución a un problema existente. Esta situación le permite explorar nuevos conocimientos y buscar cada vez información que le ayude a solucionar el proyecto planteado en la asignatura (Ver diseño instruccional propuesto); también le implica trabajar en equipo y enfrentarse a diferentes situaciones de discusión, nuevas hipótesis, nuevas formas de trabajo, nuevos retos que superar, etc., que le permitirán reafirmar sus conocimientos y sus apreciaciones de los fenómenos observados.

POSTULADO 5. “La Teoría del Aprendizaje Significativo elaborada desde una posición Organicista que se centra en el aprendizaje producido por un contexto educativo, en una situación de interiorización o asimilación a través de la instrucción”. (POZO: 1989, 209). De lo anterior, se extrae que el aprendizaje importante que realiza un individuo, se adquiere por la relación de los nuevos conocimientos con los conocimientos previos.

La aplicación de la teoría del aprendizaje significativo, permite al estudiante desarrollar la capacidad de relacionar los conceptos y conocimientos adquiridos con los nuevos y, poder generar con estas relaciones nuevos conocimientos. El diseño instruccional propuesto presenta una correlación en la red de conceptos que facilita al estudiante generar los conectores entre los conocimientos adquiridos para lograr un aprendizaje significativo. Igualmente, invita a que

el docente en su didáctica les ayude a usar los conectores de una mejor manera (ver tipos de asesorías en el diseño instruccional).

POSTULADO 6. La razón como producto de interacción social⁷. Resaltar la coordinación de comunicaciones como valor fundamental para co-construir conocimiento. La comunicabilidad y la comprensión de eso que se comunica. La interacción social ejerce sobre los sujetos relaciones de presión y de colaboración. A través de la confrontación con otros, el sujeto analiza su propio pensamiento y toma conciencia de su modo de organización, de tal manera que enriquece el mecanismo de la razón (VYGOTSKI: 1996, y LEONTIEV: 1981).

En el diseño instruccional se propone la realización de un proyecto en grupos de estudiantes (Ver diseño instruccional), estos proyectos se deben presentar a los otros grupos de estudiantes a través de foros participativos (virtuales o no) que permitan la confrontación de las experiencias y el conocimiento adquirido; esto permite la creación de criterios sustentables de lo realizado, como también el replanteamiento de conceptos adquiridos, lográndose una afirmación del saber en el estudiante.

POSTULADO 7. Las discusiones sobre didáctica en los contextos educativos contemporáneos exigen acercamientos interpretativos a estrategias de comunicación que llevan a propuestas de diseño instruccional más flexibles. Estos, a la vez, hacen evidente una arquitectura de los aprendizajes apoyada en los contenidos que se transmiten en soportes tecnológicos móviles. (TRUJILLO y JARAMILLO: 2006)

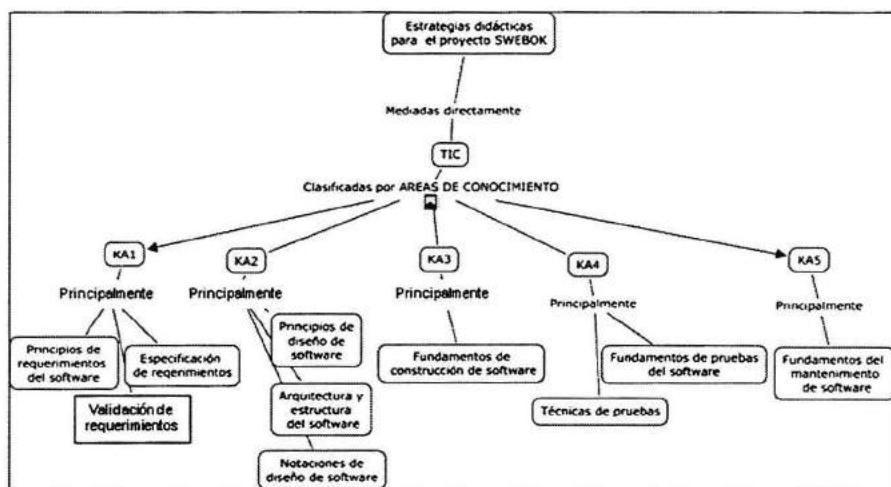
Se propone la utilización de las nuevas tecnologías como mediadoras del aprendizaje, tales como: salas de informática, videos sobre los contenidos de la asignatura, videos sobre algunas de las sesiones de clase, celulares y otros móviles que permitan ver los videos y/o comunicarse con la red inalámbrica para las sesiones virtuales, etc.

⁷ Tomado de ANAYA y TRUJILLO: 2006. Principios Pedagógicos-Didácticos. En: Un modelo para la enseñanza en contexto de la Ingeniería de Software.

3. ESTRATEGIAS DIDÁCTICAS

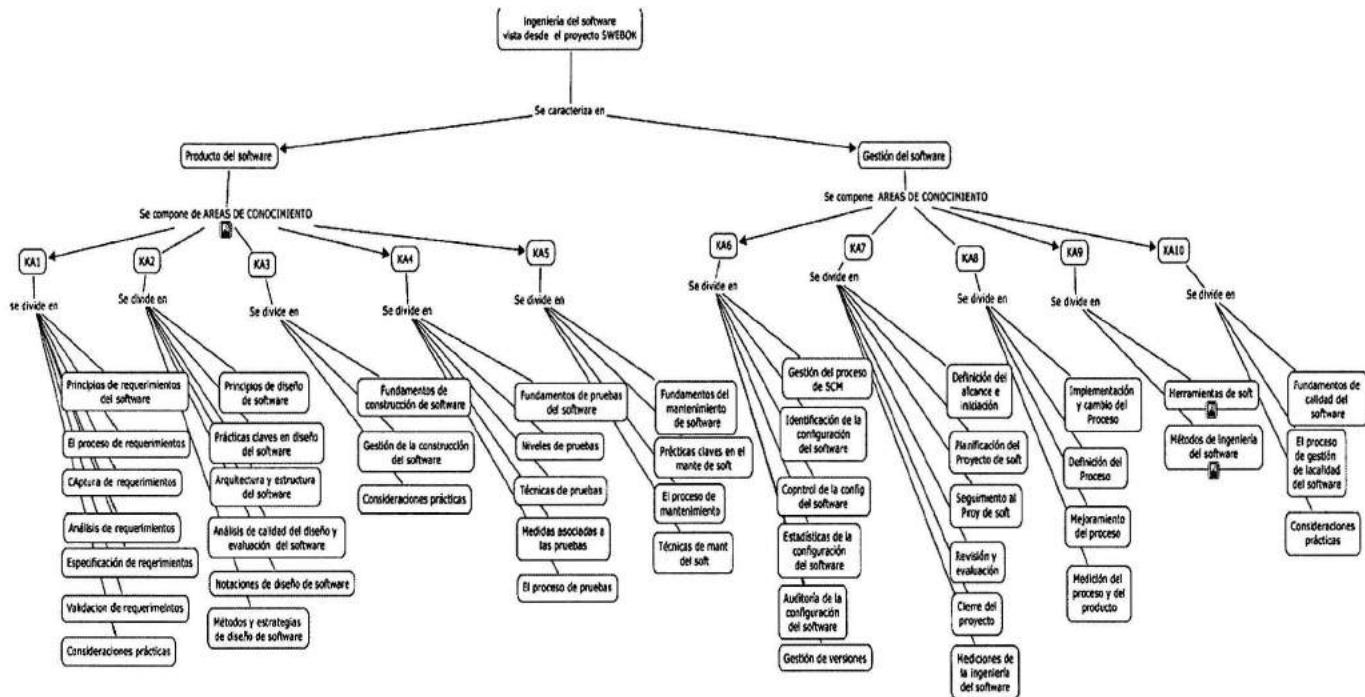
Partiendo del proyecto SWEBOK, de las diferentes posturas dirigidas a los aprendizajes analizados y condensados en los Postulados Pedagógicos referenciados y de las apreciaciones de algunos colegas y expertos⁸, se definen las siguientes estrategias didácticas aplicadas a la categoría “Productos del Software”, a través de un mapa conceptual, con el propósito de establecer cuáles estrategias son más directamente mediadas por las TIC, para tener en cuenta en el momento de elaborar el diseño instruccional del Curso inicial del proceso de enseñanza de la Ingeniería del Software, así como para futuros cursos (ver Figuras 1 y 2):

FIGURA 1. MAPA CONCEPTUAL PROPUESTO



⁸ Se agradece a Raquel Anaya, John Trujillo, Idilia Urrego y Alejandro Villa.

FIGURA 2. MAPA CONCEPTUAL DE SWEBOK.



4. DIDÁCTICA DE LAS ACTIVIDADES

Las actividades de la clase pueden fluir en espiral (Figura 3) pasando por cada uno de los estadios correspondientes. Se empieza por un proceso de documentación, en el cual el alumno obtiene por parte del docente y de sí mismo la documentación necesaria para continuar con el proceso de aprendizaje. Luego se va a un estadio de inducción donde se hace una introducción formal al tema, se aclara el tema como tal y se motiva al estudiante a seguir y desarrollar actividades definidas por el docente. En este punto el alumno hace deducciones según el caso, es decir, elabora su proyecto (se explicará más adelante) utilizando las técnicas y modelos que considere más adecuados. Se continúa con el proceso de abducción en donde plantea discusiones y nuevas soluciones al docente y a sus compañeros, para pasar nuevamente por la documentación y continuar el recorrido por el espiral. A medida que se sale de la espiral, se presenta mayor conocimiento y mayor grado de complejidad.

FIGURA 3. PROCESO EN ESPIRAL



5. ACTIVIDADES Y ESTRATEGIAS DE APRENDIZAJE

5.1. Indicación metodológica

Retomando el modelo de SWEBOK, se desarrollarán las actividades y estrategias de aprendizaje y enseñanza de la ingeniería del software en el contexto de la carrera Ingeniería de Sistemas, aplicando el análisis propuesto en el presente trabajo.

Al iniciar la asignatura, el docente explica el desarrollo del curso, el programa académico, la metodología de trabajo (didáctica de las actividades), fechas de evaluación y entregas de trabajos⁹.

5.1.1. Medios a utilizar

- Exposición magistral
- Foros, talleres y correos virtuales mediante el aplicativo informático de apoyo (HICAP IOI)
- Talleres de aplicación inmediata entre el tema expuesto magistralmente y su posible utilización en el entorno socio - económico empresarial y en la actividad profesional en general
- Talleres de observación - aplicación en la empresa y la universidad, cotejando la relación teórica práctica entre la conceptualización y la aplicación
- Talleres de investigación para que el alumno indague, busque y obtenga respuesta a cuestionamientos previamente planteados
- Discusiones guiadas
- Horas definidas extra clase para asesorías grupales e individuales
- Foros presenciales
- Exposiciones de los estudiantes, individuales y en grupos, sobre temas inherentes al proceso de desarrollo de los sistemas de información en la etapa de análisis de requisitos

⁹ Se explican en el tiempo independiente.

- Software educativo instalado en las salas de informática (HICAP IOI)

Discos compactos-CD- con:

- Aplicativos para el desarrollo de sistemas de información (software libre o comercial-académico)
- Ejemplos de sistemas de información desarrollados
- Documentos principales: Diseño Curricular y Guías de Trabajo independiente
- Libros técnico digitales
- Formatos para el desarrollo de proyectos utilizando varias metodologías
- Bibliografía básica
- Página WEB (www.itm.edu.co) donde se encuentra toda la información de la asignatura, el aplicativo informático (HICAP-IOI) y las herramientas virtuales

Discos compactos-CD- con :

- Software con la herramienta de apoyo informático en el desarrollo de sistemas de información, con talleres, foros etc.
- Videos sobre los conceptos a manejar en la asignatura, realizados por los docentes

5.1.2. Sesiones de clase

El proceso de enseñanza y aprendizaje se cumple en este curso mediante la articulación, presentación, explicación y desarrollo del tema con la exposición magistral, para dar un referente temático y conceptual de todos y cada uno de los temas, pero este método se complementa y amplía con la permanente confrontación práctica a través de talleres, lecturas, trabajos prácticos en clase, revisión y análisis de videos e investigación de sistemas informáticos ya desarrollados en empresas, que lleven al alumno a la racionalización del concepto con la práctica aplicada, que le genere cuestionamientos, preguntas e inquietudes sobre cada uno de los temas que se desarrolle en clase, que se investiguen

u observen en el exterior para hacer el cotejamiento entre el entorno y aquello que el proceso de enseñanza y aprendizaje aporta.

A partir de la presentación magistral de los temas por parte del profesor, de elaboración de talleres y realización de exposiciones por parte de los estudiantes se realiza la confrontación de lo aplicado por éstos, según el nivel de conocimiento que se esté adquiriendo. El docente brinda un espacio de asesoría para dar explicaciones individuales y a los grupos de trabajo, con el fin de inducirlos inicialmente en el área de conocimiento que requieren dominar.

El docente dejará en la página web de la asignatura el video de la clase, si lo considera necesario. Éste podrá ser visto desde dispositivos móviles o en las salas de informática o en la biblioteca; al igual que desde cualquier sitio con conexión a Internet.

Antes de cada clase los alumnos deben de estudiar la documentación definida y programada por el docente, realizar los talleres y trabajos expuestos en los CD, o en la página web, y detallados en las Guías de Trabajo independiente¹⁰.

5.1.3. Tipos de clases

El Profesor utilizará, básicamente, tres tipos de clases:

Clase presencial magistral

El profesor induce a la aprehensión de los conceptos teóricos y los fundamentos básicos de los respectivos temas, con el propósito de aclarar dudas del estudiante, para luego presentar talleres teórico-prácticos que le permiten al estudiante la confrontación de lo enseñado con lo aprendido.

Para ello, el profesor se apoya en varios ejemplos de sistemas de información desarrollados que se entregan a los estudiantes en medios magnéticos y en la página WEB, con el fin de presentar los conceptos de manera más práctica.

¹⁰ Ver las guías de trabajo independiente de la asignatura Ingeniería del Software Orientado a Objetos 2007 (GUERRERO: 2006b) éstas son similares a las que se deben desarrollar como trabajos futuros para complementar el desarrollo de esta propuesta.

En este tipo de clase también se realizan las evaluaciones a los estudiantes.

Clase presencial práctica

El profesor asesora a los estudiantes en el desarrollo de un sistema de información¹¹, tanto individual como grupal, así mismo, dará asesoría sobre los conceptos trabajados.

Clase no presencial

Son clases donde el docente interactúa con los estudiantes a través de medios computacionales (foros virtuales, video conferencia, audio conferencia, chats y video chats entre otros) durante los horarios establecidos para la asignatura; de esta manera los estudiantes pueden solicitar asesoría directa sobre los problemas que los aquejan en el desarrollo de su sistema de información, viendo de manera directa las dificultades que se les presenta con el aplicativo que se está desarrollando (problemas con el lenguaje de programación, manejador de bases de datos, CASE etc.).

5.1.4. Asesorías

El docente tiene tres tipos de asesorías, la primera forma se realiza en la segunda mitad de cada clase presencial práctica, en donde brinda ayuda a sus alumnos de forma grupal e individual; la segunda forma es extra clase (tiempo independiente), es decir en los horarios definidos por el docente para asesorías, las cuales pueden por correo electrónico. La tercera forma es por medio de una herramienta informática (HICAP-IOI a futuro) la cual permite una interacción con el docente y con los compañeros mediante foros, consultas, chats, video chats, etc.

La herramienta informática se debe utilizar en todos los cursos; sin embargo, habrá algunos que requieren de mayor presencialidad por parte del docente (Figura 1) donde se exponen los cursos que

¹¹ Se explican en el tiempo independiente.

pueden ser dictados con un acompañamiento en mayor uso de virtualidad.

Tanto los foros virtuales como los presenciales permiten a los estudiantes discutir sobre los proyectos de sus compañeros. Estos foros son dirigidos por el docente.

5.1.5. Tiempo independiente

Durante el tiempo independiente¹² del curso (128 horas), cuatro horas semanales, durante 16 semanas, el estudiante debe realizar: investigación y consulta de los conceptos de la asignatura, participar en las asesorías presenciales y virtuales y realizar un proyecto de desarrollo de un sistema de información en sus etapas de “Análisis y Diseño” las cuales culminará con la programación del sistema de información.

Antes de cada clase los alumnos deben estudiar la documentación definida y programada por el docente, realizar los talleres y trabajos expuestos en los CD, o en la página WEB, y detallados en las Guías de Trabajo Independiente (GUERRERO: 2006b).

El proyecto de desarrollo del sistema de información es realizado por grupos compuestos de tres a cinco estudiantes, quienes deben investigar en el medio el tipo de empresa a la cual le van a desarrollar el software, éste debe cumplir con los requisitos expuestos por el profesor (proceso de inducción) y por la metodología de desarrollo seleccionada. El proyecto que se realizará va incrementando su complejidad a medida que se adquiere mayor conocimiento y se desarrollan las competencias pertinentes. Continúa con el proceso de deducción que puede ser individual o grupal, es decir, el alumno compara sus deducciones con las de sus compañeros de trabajo del proyecto para reafirmar

¹² El tiempo independiente aparece establecido en el Modelo Pedagógico del ITM como el tiempo que el estudiante debe dedicar, fuera de clase, para adquirir las competencias de la asignatura, éste aparece en el diseño instruccional y depende de cada asignatura, aunque en general por cada hora de clase el estudiante debe dedicar dos horas de tiempo independiente.

o corregir los nuevos conceptos adquiridos y continuar aplicándolos al proyecto en desarrollo. Esto lo puede hacer en reuniones en el aula de clase o en reuniones virtuales mediante videoconferencia, audioconferencia, chats o video chats.

Durante el desarrollo del proyecto deben entregar tres trabajos, explicados en las Guías de Trabajo Independiente¹³. Este proyecto debe ser ingresado al aplicativo HICAP IOI para ser revisado por el docente (GUERRERO: 2006b). Durante las revisiones parciales del proyecto (asesorías) por parte del docente, se realiza un proceso de abducción donde el estudiante sustenta sus posturas críticas con respecto a los métodos y herramientas utilizadas en el proyecto.

5.1.6. Sistema de evaluación

La evaluación se divide en dos porcentajes: un 50% se evalúa a través de exámenes, participación en clase, formación profesional y trabajo en equipo; el otro 50% corresponde a la evaluación de un proyecto de desarrollo de un sistema informático, donde se evalúa el logro de la competencia a través de la creación o aplicación de soluciones prácticas.

Este último se evalúa con la entrega de tres informes, así:

- 1º Informe del “preanálisis” y parte inicial del “análisis”
- 2º Informe de la entrega del resto del “análisis” y parte inicial del “diseño”
- 3º Informe de la entrega de la parte final del “diseño” y el aplicativo funcionando

El sistema de evaluación se extiende a un permanente seguimiento, se parcela la materia por sesiones de clase, se establecen los talleres, exámenes rápidos (quices), talleres de ejecución en

¹³ Ver las guías de trabajo independiente de la asignatura Ingeniería del Software orientado a objetos 2007 (GUERRERO: 2006b) éstas son similares a las que se deben de desarrollar como trabajos futuros, para complementar el desarrollo de esta propuesta.

clase, talleres de seguimiento realizados en casa, investigación de sistemas de información en empresas y participación dinámica en clase.

Cada semana se dictan dos clases de dos horas cada una, durante 16 semanas, por ello se establece el siguiente cuadro de evaluación:

INDICADOR DE LOGRO	PORCENTAJE	FECHA
Primer informe (preanálisis - análisis)	10%	Tercera semana
Segundo informe (análisis - diseño)	20%	Octava semana
Examen parcial 1	20%	Novena semana
Tercer informe (Diseño – implementación)	20%	Decimocuarta semana
Seguimiento (Exámenes cortos, participación, entrega de tareas)	10%	Hasta la decimo- quinta semana
Examen parcial 2	20%	Decimoquinta semana

Para lograr la competencia (ver capítulo 6.1. Desarrollo de diseño curricular) el estudiante debe alcanzar todos los indicadores de logro; para ello el docente debe ponderarlos y correlacionarlos.

La evaluación de los informes depende de la calificación obtenido en el informe anterior, ya que el proyecto se compone de los 3 informes. Un estudiante no puede ganar el informe 2 sin haber ganado (alcanzar el indicador de logro) del informe 1, esto se aplica para el informe 3.

5.1.7. Aplicación de la evaluación

El docente verifica mediante exámenes y evaluación de los proyectos el avance de los alumnos y realiza las correcciones pertinentes. Para lo anterior, la herramienta informática (HICAP-IOI)

le facilitará la creación de la base de datos de preguntas, la creación de cuestionarios de preguntas aleatorias por red de conceptos, por conceptos o en general por todos los conceptos de la asignatura (GUERRERO: 2006a), además, le permitirá la evaluación de cada uno de sus estudiantes revisando los indicadores de logro y las competencias en las cuales tiene falencias¹⁴, el docente puede realizar recuperaciones creando exámenes según los logros no alcanzados, es decir, puede crear distintos tipos de exámenes para los estudiantes según las deficiencias de cada uno.

La validación de los proyectos y de las competencias debe ser realizada por un grupo de docentes, cada uno evalúa su especialidad, a través de una exposición y sustentación realizada por los alumnos.

Los estudiantes tienen un tiempo límite de entrega de los trabajos y de realización de las evaluaciones, el alumno debe solicitar la realización de sus evaluaciones antes de las fechas estipuladas, para poder pasar al siguiente nivel.

El proceso de aprendizaje es en espiral, es decir, el proceso se repite a través de las sesiones de clase, en donde se incrementa el conocimiento, el grado de complejidad y la robustez de los proyectos.

El siguiente capítulo expone los elementos pedagógicos para trabajar a través del plan de actividades del diseño curricular (instruccional) de un curso básico de Ingeniería del Software, incluye sólo los principales ítems que componen una asignatura¹⁵.

¹⁴ En el aplicativo HICAP IOI encuentra el manejo de cada uno de los ítems mencionados. Este aplicativo se encuentra en funcionamiento desde el 2006-1 en las salas de informática del ITM. Se espera una segunda versión como página web para enero del 2008.

¹⁵ El plan de actividades propuesto se basa en el curso “Ingeniería de Software IS154” del programa académico Ingeniería de Sistemas de Información del ITM. (Programa Académico Ingeniería de Sistemas de Información del ITM 2007, 2007).

6. PLAN DE ACTIVIDADES DEL DISEÑO CURRICULAR

El presente plan de actividades del diseño curricular pretende presentar una propuesta sobre cómo se puede enseñar “la Ingeniería del Software”, utilizando el modelo SWEBOK, de una forma práctica y sencilla para un nivel educativo básico profesional, equivalente a un cuarto nivel de la carrera de Ingeniería de Sistemas; comenzando por el curso inicial. Aquí encontrarán los principales elementos para llevarlo a cabo.

6.1. Desarrollo del diseño curricular

COMPETENCIAS	RED DE CONCEPTOS	INDICADORES DE LOGRO
Desarrollar e implementar sistemas informáticos utilizando herramientas informáticas que manejen diferentes metodologías generales de análisis y diseño de última generación	<p>Conceptualización básica de los sistemas de información.</p> <ul style="list-style-type: none"> • Sistema de información y sus componentes • Definición de Sistema • Definición de Sistema de Información <p>Introducción a la Ingeniería del Software:</p> <ul style="list-style-type: none"> • Definición de Ingeniería de Software • Historia • Características • Crisis del software • El software como producto. • El software como proceso. 	<ul style="list-style-type: none"> • En el contexto de un sistema de información de una organización, propone los modelos y metodologías adecuados para el desarrollo de un software que dé solución a un problema específico. • Formula y evalúa la factibilidad de un proyecto específico de desarrollo de software. • Realiza el desarrollo completo de un aplicativo software en la solución de un problema específico de un proceso productivo.

COMPETENCIAS	RED DE CONCEPTOS	INDICADORES DE LOGRO
<ul style="list-style-type: none"> • Modelos y metodologías de desarrollo de software <p>Formulación y evaluación básicas de proyectos de desarrollo de software.</p> <ul style="list-style-type: none"> • Ejecución de un proyecto de desarrollo de software • Etapa de análisis de requerimientos: <p>Principios y conceptos de análisis.</p> <ul style="list-style-type: none"> • Construcción de modelos de información • Principios de la especificación de los requisitos del software. • Etapa de Diseño de software: - Principios y Conceptos del Diseño - El lenguaje de Modelado Unificado (UML) para el desarrollo de la etapa de diseño • Diseño de la interfaz de usuario. - Diseño de pantallas - Diseño de reportes - Diseño de bases de datos 	<ul style="list-style-type: none"> • Realiza la etapa de análisis de requerimientos en el desarrollo en un aplicativo específico, utilizando una herramienta informática de desarrollo de software de última generación. • Realiza el desarrollo completo de un aplicativo software en la solución de un problema específico de un proceso productivo. • Realiza la etapa de diseño de software en el desarrollo en un aplicativo específico, utilizando una herramienta informática de desarrollo de software de última generación. • Realiza el desarrollo completo de un aplicativo software en la solución de un problema específico de un proceso productivo. 	

6.2. Diseño instruccional detallado

SITUACIONES DE APRENDIZAJE CONCEPTOS NUEVOS A TRABAJAR EN LA SESIÓN	COMPETENCIAS	OBJETIVO	ACTIVIDADES DURANTE LA CLASE	DESCRIPCIÓN DE LAS ACTIVIDADES ACADÉMICAS SIN ACOMPAÑAMIENTO	INDICADORES DE LOGRO	CONTENIDOS, MATERIALES Y RECURSOS (DISCRIMINADOS EN DETALLE) RECURSOS	No. SESIÓN (HORAS TP/TI) TP TIEMPO PRESENCIAL TI TIEMPO INDEPENDIENTE
Conceptualización básica de los sistemas de información. <ul style="list-style-type: none"> • Sistema de Información y sus componentes • Definición de Sistema • Definición de Sistema de Información Introducción a la Ingeniería del Software: <ul style="list-style-type: none"> • Definición de Ingeniería de Software • Historia • Características • Crisis del software • El software como producto. • El software como proceso. • Modelos y metodologías de Desarrollo de software Formulación y evaluación básicas de proyectos de desarrollo de software	Conocer y saber manejar los principios básicos del modelado orientado a objetos.	Conocer todos los aspectos que rodean el desarrollo del software. Conocer cómo se gestiona un proyecto de desarrollo de software	Exponer sobre los modelos de desarrollo de software por grupos de máximo tres estudiantes. Realizar asesorías sobre el preanálisis del proyecto planteado por los estudiantes (en grupo de 3 a 5 estudiantes). Realizar actividades lúdicas con el fin de que los estudiantes reconozcan el software como producto y como proceso. Asesoría virtual (Chat) sobre la realización de la formulación y evaluación del proyecto de software a ser desarrollado. Realización de quiz, utilizando HICAP IOI Clase presencial para evaluación	Consultar sobre el papel de los sistemas de información en las organizaciones. Leer los conceptos de la arquitectura cliente servidor. Leer primeros 2 capítulos del Libro: "Elementos básicos de Ingeniería del Software" (GUERRERO, 2007). Lectura de los capítulos 1 y 2 en el libro de "Ingeniería del Software" (PRESSMAN, 2002). Leer "Guía de Trabajo independiente No 1". Lectura del documento "Programación de Scripts".	<ul style="list-style-type: none"> • En el contexto de un sistema de información de una organización, propone los modelos y metodologías adecuados para el desarrollo de un software que dé solución a un problema específico • Formula y evalúa la factibilidad de un proyecto específico de desarrollo de software • Quiz de verificación de los conceptos(parte del 10% de seguimiento) • Evaluación del anteproyecto (parte del primer informe) 	Video Beam y computador CD's del curso Videos de los conceptos del curso, los ejemplos de sistemas informáticos y libros digitales. Página WEB de la asignatura. Montaje del mapa conceptual con los conceptos principales seleccionados del texto leído. http://www.educar.org/articulos/usodemapas.asp	1 - 4 (16/32)

SITUACIONES DE APRENDIZAJE CONCEPTOS NUEVOS A TRABAJAR EN LA SESIÓN	COMPETENCIAS	OBJETIVO	ACTIVIDADES DURANTE LA CLASE	DESCRIPCIÓN DE LAS ACTIVIDADES ACADÉMICAS SIN ACOMPAÑAMIENTO	INDICADORES DE LOGRO	CONTENIDOS, MATERIALES Y RECURSOS (DISCRIMINADOS EN DETALLE) RECURSOS	No. SESIÓN (HORAS TP/TI) TP TIEMPO PRESENCIAL TI TIEMPO INDEPENDIENTE
Ejecución de un proyecto de desarrollo de software: <ul style="list-style-type: none"> Etapa de Análisis de requerimientos: Principios y Conceptos de análisis Construcción de modelos de información Principios de la especificación de los requisitos del software El lenguaje de Modelado Unificado (UML) para el desarrollo de la etapa de análisis Herramientas informáticas de última generación para el desarrollo de software 	Conocer y saber manejar el análisis orientado a objetos y la forma como sus modelos permiten comprender las clases, sus relaciones y comportamientos. Utilización de estándares en el desarrollo de sistemas de información	Conocer los aspectos que rodean un análisis de requerimientos	Realización de asesorías sobre la primera parte del Análisis del sistema del proyecto seleccionado, reconociendo la aplicación de los conceptos técnicos, métodos y mediciones aplicables al análisis	Realizar lectura previa de cada uno de los temas relacionados con los aspectos que rodean el proceso de análisis de requisitos del software. Lectura de los capítulos 10 y 11 en el libro de "Ingeniería del software" de Roger Pressman (PRESSMAN, 2002)	Realizar lectura previa de cada uno de los temas relacionados con los aspectos que rodean el proceso de análisis de requisitos del software. Lectura de los capítulos 10 y 11 en el libro de "Ingeniería del software" de Roger Presuman (PRESSMAN, 2002)	<ul style="list-style-type: none"> Realiza la etapa de Análisis de Requerimientos en el desarrollo en un aplicativo específico, utilizando una herramienta informática de desarrollo de software de última generación Quiz de verificación de los conceptos (parte del de seguimiento) 	CD s del curso Videos de los conceptos del curso Página WEB De la asignatura Aplicativo HICAP IOI 5 - 6 (8/16)

SITUACIONES DE APRENDIZAJE CONCEPTOS NUEVOS A TRABAJAR EN LA SESIÓN	COMPETENCIAS	OBJETIVO	ACTIVIDADES DURANTE LA CLASE	DESCRIPCIÓN DE LAS ACTIVIDADES ACADÉMICAS SIN ACOMPAÑAMIENTO	INDICADORES DE LOGRO	CONTENIDOS, MATERIALES Y RECURSOS (DISCRIMINADOS EN DETALLE) RECURSOS	No. SESIÓN (HORAS TP/TI) TP TIEMPO PRESENCIAL TI TIEMPO INDEPENDIENTE
Ejecución de un proyecto de desarrollo de software: <ul style="list-style-type: none"> El lenguaje de Modelado Unificado (UML) para el desarrollo de la etapa de análisis Herramientas informáticas de última generación para el desarrollo de software 	Conocer y saber manejar los elementos de un modelo de análisis orientado a objetos, con respecto a los diagramas de casos de uso. Utilización de estándares en el desarrollo de sistemas de información.	Reconocer el análisis del dominio, los componentes genéricos del modelo de AOO, y el Proceso de AOO	Llevar a cabo actividades grupales para que los estudiantes identifiquen las diferentes metodologías de AOO, pero sobre todo, reconocer como expresa el análisis mediante UML con respecto a los diagramas de casos de uso. Exponer el aplicativo "Sistemas de préstamos bibliotecario". Utilizar los trabajos desarrollados por los otros compañeros para ratificar conceptos. Clase presencial para evaluación. Asesoría al desarrollo del informe 2 (Análisis del sistema y Diseño Parte inicial).	Exploración del libro: UML y patrones introducción al análisis y diseño orientado a objetos (LARMAN, 1999). Realizar las preguntas que aparecen al final de los capítulos. Leer "Guía de Trabajo independiente N° 2" Comenzar a realizar el Informe del Análisis del sistema bajo la metodología. Análisis orientado a objetos del proyecto a desarrollar en grupos de estudiantes. Realización de foro virtual para revisión de los conceptos aprendidos.	<ul style="list-style-type: none"> Realiza la etapa de Análisis de Requerimientos en el desarrollo en un aplicativo específico, utilizando una herramienta informática de desarrollo de software de última generación Evaluación del 1º informe (Preanálisis y análisis) 	Aula de clase Biblioteca. CD's del curso. Videos de los conceptos del curso, los ejemplos de sistemas informáticos y libros digitales. Página WEB de la asignatura. Página: www.software-engin.com Aplicativo HICAP IOI	7 -9 (6/12)

SITUACIONES DE APRENDIZAJE CONCEPTOS NUEVOS A TRABAJAR EN LA SESIÓN	COMPETENCIAS	OBJETIVO	ACTIVIDADES DURANTE LA CLASE	DESCRIPCIÓN DE LAS ACTIVIDADES ACADÉMICAS SIN ACOMPAÑAMIENTO	INDICADORES DE LOGRO	CONTENIDOS, MATERIALES Y RECURSOS (DISCRIMINADOS EN DETALLE) RECURSOS	No. SESIÓN (HORAS TP/TI) TP TIEMPO PRESENCIAL TI TIEMPO INDEPENDIENTE
Ejecución de un proyecto de desarrollo de software: <ul style="list-style-type: none"> El lenguaje de Modelado Unificado (UML) para el desarrollo de la etapa de análisis Herramientas informáticas de última generación para el desarrollo de software 	Conocer y saber manejar los elementos de un modelo de análisis orientado a objetos, con respecto a los diagramas objeto-objetos y objeto-comportamiento.	Reconocer el análisis del dominio, los modelos: objeto-relación y objeto-comportamiento.	Realizar un foro participativo virtual con el fin que los estudiantes distingan los objetos y sus relaciones en el modelado AOO, pero sobre todo, reconocer como lo expresa el análisis mediante UML. Clase presencial para evaluación utilizando HICAP IOI.	Exploración del libro: UML y patrones introducción al análisis y diseño orientado a objetos (LARMAN, 1999), con respecto a los conceptos tratados, profundizando en el tema de casos de uso. Realizar lecturas de los capítulos 20 y 21 del libro Ingeniería del software: un enfoque práctico (PRESSMAN, 2002). Utilizar HICAP IOI para lectura y revisión de los conceptos a manejar en clase.	<ul style="list-style-type: none"> Realiza la etapa de Análisis de Requerimientos en el desarrollo en un aplicativo específico, utilizando una herramienta informática de desarrollo de software de última generación Evaluación informe 2 (Análisis del sistema y Diseño parte inicial) Examen sobre la etapa de Análisis 	Aula de clase. Internet Biblioteca Sala de informática con acceso a INTERNET. Página WEB de la asignatura CDs del curso. Videos de los conceptos del curso, los ejemplos de sistemas informáticos y libros digitales. Explorar la página www.software-engin.com	10 - 12 (6/12)

SITUACIONES DE APRENDIZAJE CONCEPTOS NUEVOS A TRABAJAR EN LA SESIÓN	COMPETENCIAS	OBJETIVO	ACTIVIDADES DURANTE LA CLASE	DESCRIPCIÓN DE LAS ACTIVIDADES ACADÉMICAS SIN ACOMPAÑAMIENTO	INDICADORES DE LOGRO	CONTENIDOS, MATERIALES Y RECURSOS (DISCRIMINADOS EN DETALLE) RECURSOS	NO. SESIÓN (HORAS TP/TI) TP TIEMPO PRESENCIAL TI TIEMPO INDEPENDIENTE
<ul style="list-style-type: none"> Principios y conceptos del diseño El lenguaje de Modelado Unificado (UML) para la etapa de Diseño 	Conocer y saber manejar los elementos de un modelo de diseño orientado a objetos, desde su concepción. Implementar los diseños a un aplicativo de software.	Reconocer el proceso del diseño en general y del diseño orientado a objetos en particular	<p>Retomar los ejemplos de la bibliografía con el fin de que los estudiantes distingan los elementos del diseño orientado a objetos, mediante UML.</p> <p>Asesoría al informe 2, sobre los temas de diseño.</p> <p>Evaluación del informe 2. (Análisis del sistema y Diseño Parte inicial).</p>	<p>Utilizando el AOO realizar la narrativa, clases, atributos y métodos, los casos de uso, curso normal de los eventos utilizando una herramienta informática de desarrollo de software de última generación</p> <p>2.</p> <p>Análisis del sistema 20% (En grupos de 3 a 5 estudiantes).- Asistir a asesoría con el docente en los grupos de trabajo.</p> <p>Ingresar el informe 1 y lo que se lleva del segundo al aplicativo HICAP IOI.</p>	<ul style="list-style-type: none"> Realiza la etapa de Diseño de Software en el desarrollo en un aplicativo específico, utilizando una herramienta informática de desarrollo de software de última generación Realiza el desarrollo completo de un aplicativo software en la solución de un problema específico de un proceso productivo. 	<p>Aula de clase Internet Biblioteca CD s del curso. Videos de los conceptos del curso, los ejemplos de sistemas informáticos y libros digitales.</p> <p>Explorar la página www.software-engineering.com Aplicativo HICAP IOI</p>	13 - 14 (4/8)

SITUACIONES DE APRENDIZAJE CONCEPTOS NUEVOS A TRABAJAR EN LA SESIÓN	COMPETENCIAS	OBJETIVO	ACTIVIDADES DURANTE LA CLASE	DESCRIPCIÓN DE LAS ACTIVIDADES ACADÉMICAS SIN ACOMPAÑAMIENTO	INDICADORES DE LOGRO	CONTENIDOS, MATERIALES Y RECURSOS (DISCRIMINADOS EN DETALLE) RECURSOS	No. SESIÓN (HORAS TP/TI) TP TIEMPO PRESENCIAL TI TIEMPO INDEPENDIENTE
<p>Definición de los siguientes diseños, e implementación de los mismos</p> <ul style="list-style-type: none"> • Diseño de clases • Diseño de Contratos • Diseño de la interfaz de usuario • Diseño de pantallas • Diseño de reportes 	<p>Conocer y saber manejar los elementos de un modelo de diseño orientado a objetos, con respecto al diseño arquitectónico y al diseño de contratos.</p>	<p>Elaborar el diseño arquitectónico de un software.</p> <p>Elaborar el diseño detallado de un sistema de información.</p>	<p>Explicación de los temas y Asesoría del informe 3 en una herramienta informática como Visio, UML <i>Parading</i> etc.</p> <p>Comenzara a realizar el informe 3, en una herramienta informática como Visio, UML <i>Parading</i> etc.</p> <p>Asesoría y evaluación del informe 3 (elementos finales del diseño del sistema e implementación) y el aplicativo funcionando</p> <p>Clase presencial para evaluación final utilizando HICAP IOI</p>	<p>Taller sobre modelos objeto-relación y objeto comportamiento.</p> <p>Realización de los modelos objeto-relación y objeto comportamiento</p> <p>el informe 2 en una herramienta informática como Visio, UML <i>Parading</i> etc.</p> <p>Exploración del documento web "Object-Oriented Design" en las direcciones:</p> <p>http://mini.net/cetus/oo.uml.html</p>	<p>Realiza el desarrollo completo de un aplicativo software en la solución de un problema específico de un proceso productivo.</p> <p>Evaluación del informe 3 (elementos finales del Diseño del sistema e implementación) y el aplicativo funcionando.</p>	<p>Biblioteca Páginas: http://www.iscmolina.com/Herramientas CASE.html</p> <p>http://gyc.escet.urjc.es/simple_com/phd-theses/node/11.html</p>	15 - 16 (4/8)

SITUACIONES DE APRENDIZAJE CONCEPTOS NUEVOS A TRABAJAR EN LA SESIÓN	COMPETENCIAS	OBJETIVO	ACTIVIDADES DURANTE LA CLASE	DESCRIPCIÓN DE LAS ACTIVIDADES ACADÉMICAS SIN ACOMPAÑAMIENTO	INDICADORES DE LOGRO	CONTENIDOS, MATERIALES Y RECURSOS (DISCRIMINADOS EN DETALLE) RECURSOS	No. SESIÓN (HORAS TP/TI) TP TIEMPO PRESENCIAL TI TIEMPO INDEPENDIENTE
	Implementar los diseños a un aplicativo de software.	Entrega del aplicativo funcionando.		<p>Utilizar HICAP IOI para lectura y revisión de los conceptos a manejar en clase.</p> <p>Realizar lecturas del capítulos 22, 25 y 26 del libro Ingeniería del software: un enfoque práctico (PRESSMAN, 2002).</p> <p>Realizar lecturas del capítulos 22, 25 y 26 del libro Ingeniería del software: un enfoque práctico (PRESSMAN, 2002).</p> <p>Exploración del libro: UML y patrones introducción al análisis y diseño orientado a objetos (LARMAN, 1999), con respecto a los conceptos tratados, enfatizando en contratos.</p> <p>Asistir a asesoría con el docente en los grupos de trabajo.</p> <p>Ingresar el informe 3 al aplicativo HICAP IOI.</p> <p>Desarrollar los ejercicios al final del capítulo 22 del libro de PRESSMAN, Roger S. Ingeniería del Software: un enfoque práctico. 5. ed. Madrid McGraw-Hill, 2002. 601 p.</p>	<p>Examen sobre Diseño 20%.</p> <p>Terminación del seguimiento 10%</p>	<p>Sala de informática con acceso a INTERNET.</p>	

6.3. Bibliografía del curso

- BOOCH, Grady. (1998). Análisis y diseño orientado a objetos con aplicaciones. 2^a ed. México : s.n. 638 p.
- BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. (1999). El lenguaje unificado de modelado. Madrid: Addison Wesley. 432 p. + CD-ROM.
- BROPHY, J. (1999) Teaching En UNESCO: Educational Practices Series, nº 1, 35.
- GUERRERO, Diego. (2007). Elementos básicos de ingeniería del software. Medellín: ITM, próximo a editar.
- LARMAN, Craig. (1999). UML y patrones introducción al análisis y diseño orientado a objetos. México: Prentice-Hall. 507 p. 2^a edición.
- MEYER, Bertrand. Construcción de software orientado a objetos. 2. ed. Madrid, Prentice-Hall, 1999. 1198 p. + CD-ROM.
- PRESSMAN, Roger S. (2002). Ingeniería del software: un enfoque práctico. 5. ed. Madrid: McGraw-Hill. 601p.
- TOPPING, K. (2000) Tutoring En UNESCO: Educational Practices Series nº 5, 35p.
<http://www.ibe.unesco.org/International/Publications/EducationalPractices/prachome.htm>. Tomado de la red el 25 de septiembre de 2002.

7. CONCLUSIONES Y TRABAJO FUTUROS

Este artículo ha presentado una propuesta para la didáctica en la enseñanza de la Ingeniería de Software dirigida hacia la formación de competencias que dinamicen la práctica del desarrollo en las empresas de software. El modelo de enseñanza en el cual se basa se sustenta en teorías pedagógicas que perciben el proceso de aprendizaje en ciclos consecutivos a manera de espiral, fundamentado en la Teoría de la Actividad.

El presente trabajo permitió una conceptualización más próxima de las teorías del aprendizaje y cómo se pueden aplicar en un contexto real de enseñanza y aprendizaje de la Ingeniería del Software.

La experiencia docente no es suficiente por sí misma, se requiere de un recorrido histórico para comprender aún mejor las distintas

formas como el hombre aprende y las distintas formas como el hombre enseña. Es necesario comprender los diferentes estadios del hombre, del conocimiento y de la tecnología para poder desarrollar estrategias de enseñanza y aprendizaje cada vez más certeras.

No hay duda que las TIC permiten un mayor acercamiento a la información, asimismo, es de gran ayuda tanto para los docentes como para los estudiantes en su proceso de aprendizaje, puesto que en el aula de clase todos aprenden.

Esta propuesta de estrategias didácticas y diseño curricular para el curso básico de la enseñanza de la Ingeniería del Software a partir del proyecto SWEBOK, es el resultado de acciones concretas que se vienen desarrollando en la decanatura de Sistemas de Información, del INSTITUTO TECNOLÓGICO METROPOLITANO – ITM-, en conjunto con la Escuela de Pedagogía de la misma Institución, para fortalecer el perfil del ingeniero de Sistemas de Información, como desarrollador de software.

En la nueva estructura del currículo (2007-1) el área de Ingeniería de Software cuenta con dos asignaturas en los semestres V y VIII dentro del ciclo de formación básica. Los estudiantes que optan por enfatizar en el tema tendrán la oportunidad de seleccionar una electiva que sirve de profundización en algunos temas (ingeniería de requisitos, análisis y diseño orientado a objetos, métricas de software, calidad de software aún no definida oficialmente). Este trabajo permitió una reflexión profunda acerca de la práctica del desarrollo del software, además que abrió un espacio para el análisis de la didáctica que se estaba llevando a cabo en las asignaturas analizadas.

El diseño del currículo fundamentado en las competencias y en las actividades de conocimiento antes que en contenidos temáticos, favorece una tendencia de aprendizaje orientada al quehacer de la práctica del desarrollo en las empresas de software. Esta postura exige que la universidad establezca nexos coherentes con las industrias, conozca sus problemas y proponga mecanismos academia-empresa que favorezcan la participación de los estudiantes en proyectos encaminados a solucionar problemas reales de las

empresas, partiendo de propuestas y políticas establecidas desde el Centro de Investigación.

Este trabajo continúa con una definición detallada del diseño curricular de la asignatura Ingeniería del Software, donde se detallarán las estrategias de aprendizaje dando prioridad a la didáctica expuesta en este artículo, y donde se debe especificar las ayudas informáticas y virtuales que apoyan de una manera más dinámica este curso. Además es un resultado del proyecto de investigación HICAP-IOI (GUERRERO, 2006a), siendo ésta la base para todas las demás ayudas de informática-educativa que acompañarán al curso en un futuro.

Se requieren plantear las Guías de Trabajo independiente, de manera que incluyan elementos de informática educativa, así como de virtualidad, con el fin de facilitar el aprendizaje en los alumnos.

Continuar desarrollando el Proyecto HICAP-IOI con las nuevas propuestas de ayudas virtuales, interactivas y computacionales y extenderlas a otras asignaturas.

REFERENCIAS

- ANAYA, Raquel (2005). Una visión de la enseñanza de la ingeniería de software como apoyo a las empresas de software. Medellín: Revista Universidad EAFIT. Próximo a publicar.
- ANAYA, Raquel y TRUJILLO, John. (2006)."Un modelo para la enseñanza en contexto de la Ingeniería del Software". Medellín: Departamento de Informática y Sistemas de la Universidad E.A.F.I.T.
- BOOCH, Grady(1998). Análisis y diseño orientado a objetos con aplicaciones. 2^a ed. México: s.n., 638 p.
- BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar (1999). El lenguaje unificado de modelado. Madrid: Addison Wesley. 432 p. + CD-ROM.
- BROPHY, J. (1999).Teaching en UNESCO: Educational practices series, N° 1, 35.
<http://www.ibe.unesco.org/International/Publications/EducationalPractices/prachome.htm> tomado de la red el 25 de septiembre de 2002.

- COLON, A. J. (2002). La (de) construcción del conocimiento pedagógico: Nuevas perspectivas en la Teoría de la educación ". Barcelona: Ediciones Paidos, (Pág. 83-157).
- GUERRERO, Diego. (2006), Diseño del micro currículo "Ingeniería del software ISI54". Medellín: Decanatura Ingeniería de sistemas de información, Instituto Tecnológico Metropolitano. Disponible en www.itm.edu.co
- _____ A. (2006A) Herramienta informática como apoyo pedagógico a la asignatura ingeniería del software orientado a objetos -HICAP-IOI-. Proyecto de investigación. Medellín: Centro de investigación Instituto Tecnológico Metropolitano.
- _____ B. (2006B). Guías de trabajo independiente del núcleo: "ingeniería del software orientado a objetos IOI52". Medellín: Decanatura Ingeniería de sistemas de información, Instituto Tecnológico Metropolitano, Guide to the Software Engineering Body of Knowledge. SWEBOK,2004 Version. A project of the IEEE Computer Society Professional Practices Comité. Disponible en www.swebok.org
- LARMAN, Craig. (1999). UML y patrones introducción al análisis y diseño orientado a objetos. 2^a edición. México: Prentice-Hall. 507p.
- LEONTIEV, A. N. (1981). Problems of the development of the mind. Moscow: Editorial Progress.
- MEYER, Bertrand. (1999) Construcción de software orientado a objetos. 2^a ed. Madrid: Prentice-Hall, 1198 p. + CD-ROM.
- POZO, J.I. (1989) "Teorías cognitivas del aprendizaje". Madrid: Ed. Morata.
- PRESSMAN, Roger S. (2002). Ingeniería del software: un enfoque práctico. 5. ed. Madrid: McGraw-Hill. 601p.
- PROGRAMA ACADÉMICO INGENIERIA DE SISTEMAS DE INFORMACIÓN DEL ITM 2007. (2007). Plan de estudios. Medellín: Decanatura Ingeniería de sistemas de información, Instituto Tecnológico Metropolitano. Disponible en www.itm.edu.co
- TOPPING, K. (2000). Tutoring En UNESCO: Educational Practices Series n° 5, 35.
- TRUJILLO V, John Antonio y JARAMILLO R., Carlos Mario. (2006). "Estrategias didácticas en educación superior con la mediación de la computación móvil", Revista Educación y Pedagogía, Medellín, Universidad de Antioquia, facultad de Educación, vol. XVIII, num. 45, (Mayo-agosto). 93-107p.

- URREGO, María Idilia; CASTAÑO, Luz Elena. (1999).Modelo pedagógico. Medellín: Instituto Tecnológico Metropolitano,
- VYGOTSKI, L. S. (1996). El desarrollo de los procesos psicológicos superiores. Barcelona: Crítica traducción de "Mind in Society. The development of Higher Psychological Processes" Cambridge, Massachusetts: Harvard University Press.,
http://www.ibe.unesco.org/International/Publications/Educational_Practices/prachome.htm tomado de la red el 25 de septiembre de 2002.

AGRADECIMIENTOS

Mis más sinceros agradecimientos a la doctora Idilia Urrego del ITM por su colaboración y desinteresado apoyo en la pertinencia pedagógica del trabajo; igualmente, a la doctora Raquel Anaya y al doctor John Trujillo de la Universidad EAFIT, por sus conocimientos y enseñanzas, y sobre todo por su confianza, que permitieron la elaboración de este artículo.

Agradezco, en gran manera, al ingeniero Fray Osorio, Decano de Ingeniería de Sistemas de Información del ITM por su colaboración y confianza, al docente Alejandro Villa y alumnos de las asignatura "Ingeniería del Software Orientada a Objetos IOI52" 2007 del ITM, que me ayudaron en la revisión y actualización de este trabajo, a los integrantes del proyecto HICAP IOI por su trabajo investigativo.

