

Entorno integral de enseñanza aprendizaje: Diseño y Construcción de Software Dirigido por SWEBOK

Sandra del Pilar Chuquín Badillo, Jorge Enrique Otálora Luna

^a Maestría en Tecnología Informática, Universidad Pedagógica y Tecnológica de Colombia, Av. Central del Norte Edificio Administrativo Piso 4, Tunja., Colombia
spchuquin@yahoo.com

^b Docente Facultad de Ingeniería, Universidad Pedagógica y Tecnológica de Colombia, Av. Central del Norte Edificio Administrativo Piso 4, Tunja, Colombia
jorge.otalora@uptc.edu.co

Resumen. Este artículo presenta los resultados obtenidos en la aplicación de estrategias metodológicas basadas en la guía SWEBOK a un grupo de estudiantes de la Universidad Pedagógica y Tecnológica de Colombia (UPTC) y propuesta de infraestructura en esta área realizada en el desarrollo del trabajo de titulación “Evaluación de estrategias metodológicas y propuesta de infraestructura para enseñanza de diseño y construcción de software”. La investigación es un aporte que coadyuva con el objetivo principal del macroproyecto al que pertenece, denominado “Propuesta metodológica para el desarrollo de contenidos del área de ingeniería de software en la Escuela de Ingeniería de Sistemas y Computación (EISC) de la UPTC, a partir de los lineamientos de la guía SWEBOK”. El macroproyecto pretende mejorar la calidad en los procesos de enseñanza aprendizaje de las áreas de ingeniería de software del programa de ingeniería de sistemas y computación de la UPTC, teniendo en cuenta componentes como: contenidos orientados por la guía SWEBOK, estrategias metodológicas apropiadas de acuerdo los tópicos de cada área del conocimiento, instrumentos de evaluación adecuados para medir resultados, herramientas tecnológicas pertinentes y espacios necesarios para llevar a cabo la apropiación y transmisión de conocimientos por parte de estudiantes y docentes, dando lugar a la conformación de un entorno de enseñanza aprendizaje integral.

Palabras Clave: SWEBOK, enseñanza, competencia, software, entorno.

1 Introducción

La UPTC [1] interesada en mejorar la calidad de enseñanza de las áreas de ingeniería de software de la EISC, formuló el macroproyecto “Propuesta metodológica para el desarrollo de contenidos del área de ingeniería de software en la Escuela de Ingeniería de Sistemas y Computación de la UPTC, a partir de los lineamientos de la guía SWEBOK”. El macroproyecto está estructurado en dos etapas y siete fases. Se desglosa en subproyectos destinados a plantear y validar estrategias de enseñanza aprendizaje que permitan lograr resultados de alta calidad en las áreas de la ingeniería de software. A continuación se describen brevemente las etapas.

Etapa I. Los objetivos planteados y descritos sucintamente son: determinar el nivel de cumplimiento de los contenidos de la línea de ingeniería de software, frente a los propuestos en la línea SWEBOK; identificar las estrategias pedagógicas y metodológicas actualmente utilizadas; evaluar la efectividad de uso de las estrategias pedagógicas y metodológicas identificadas previamente; definir las estrategias adecuadas; diseñar y construir guías de enseñanza aprendizaje a partir de SWEBOK.

Fases: 1. Recopilación de información; 2. Análisis y selección de información; 3. Aplicación de conceptos al diseño de estrategias metodológicas; 4. Aplicación de estrategias en pruebas piloto.

La etapa I, para las áreas de conocimiento de diseño y construcción de software, se llevó a cabo con el subproyecto “Propuesta pedagógica y metodológica basada en SWEBOK para la enseñanza del diseño y construcción de software” [2].

Etapa II. Los objetivos planteados son: evaluar el impacto de la aplicación de las guías de enseñanza aprendizaje; especificar las herramientas metodológicas necesarias para impartir las temáticas; determinar los principales aspectos necesarios para el diseño de infraestructura física (espacios locativos y herramientas tecnológicas) para el desarrollo de los componentes metodológicos propuestos.

Fases: 1. Evaluación de estrategias planteadas; 2. Síntesis de herramientas y estrategias metodológicas; 3. Planteamiento de aspectos de infraestructura.

La etapa II, para las áreas de conocimiento de diseño y construcción de software, se llevó a cabo con el subproyecto “Evaluación de estrategias metodológicas y propuesta de infraestructura para enseñanza de diseño y construcción de software” [3], objeto principal del presente documento.

La investigación [3] es pertinente para los intereses del ente educativo puesto que su plan de desarrollo va alineado con el fortalecimiento de los programas académicos y la armonización con las áreas estratégicas de desarrollo regional.

Se encamina a apoyar el logro de los objetivos del plan nacional de desarrollo 2014-2018 [4] “Todos por un nuevo país”, en el capítulo “Colombia la más educada”, pretende que para el año 2025 Colombia ocupe los primeros lugares en educación, haciendo necesario cambiar los modelos educativos y realizar nuevas propuestas.

Para el futuro ingeniero adquirir competencias y habilidades en las áreas de diseño y construcción de software le permitirá contar con mejores perspectivas laborales y podrá satisfacer sus necesidades y expectativas con más oportunidades.

1.1 Problema

A raíz de la identificación de debilidades en las estrategias de enseñanza aprendizaje del PAE [5] en las áreas de diseño y construcción de software del Programa de Ingeniería de Sistemas y Computación de la UPTC, se lleva a cabo la investigación [3], que pretende mitigar las falencias y realizar un aporte que permita mejorar la calidad de las áreas señaladas dentro del programa académico. Las debilidades identificadas se sustentan en resultados de la etapa I observados en la tabla 1 y en información suministrada por fuentes de la UPTC, según tabla 2.

Los resultados de la tabla 1 se obtuvieron de encuestas realizadas a estudiantes de noveno y décimo semestre, con terminación académica, egresados y docentes, en donde mediante preguntas se invitó a los participantes a responder sobre aspectos como la completitud temática, la modalidad utilizada (teórica, teórico-práctica), la importancia del tema, la estrategia de enseñanza aplicada, entre otras. Las respuestas se tabularon y ponderaron, aplicando el procedimiento de evaluación de programas de la UPTC [6], llegando a los datos presentados a continuación [2].

Tabla 1. Análisis de cumplimiento de asignaturas con respecto a la guía SWEBOK y efectividad de las estrategias pedagógicas utilizadas [2]. Resultados en escala de 1 a 5

Área de conocimiento	Análisis de cumplimiento de asignaturas con respecto a la guía SWEBOK	Análisis de efectividad de estrategias pedagógicas
Diseño de software	3,15	2,97
Construcción de software	3,01	2,95

Se evidencia que es necesario realizar ajustes al programa académico, para mejorar la satisfacción, lo que condujo a la propuesta de guías cubriendo la temática SWEBOK, utilizando estrategias que apliquen la taxonomía de Bloom. La segunda etapa evalúa la efectividad de las estrategias pedagógicas propuestas.

Tabla 2. Informe de egresados según modalidad de grado. Fuentes: CEDEC UPTC, Secretaría EISC UPTC, GOS UPTC y Secretaría de Facultad de Ingeniería UPTC.

Escuela de Ingeniería de Sistemas y Computación. UPTC, Tunja, 2013						
Año de egreso	Software	BD	Redes	Práctica Empresarial	Especialización en BD	Maestría en TI
	Trabajos de grado					
2000	8	2	2	1	0	0
2001	2	0	1	0	0	0
2002	6	2	1	6	0	0
2003	12	4	1	9	0	0
2004	4	8	0	8	0	0
2005	8	6	0	9	0	0
2006	18	2	0	3	0	0
2007	27	11	1	0	0	0
2008	27	5	3	1	0	0
2009	15	7	5	0	0	0

2010	14	7	2	0	10	0
2011	14	7	3	0	30	0
2012	10	5	3	0	17	11
2013	5	6	0	0	14	4
Totales	170	72	22	37	71	15
Total modalidad trabajo de grado = 264. Total egresados 2000-2013 = 374 Trabajos en software: 170 que corresponde al 45% del total.						

1.2 Alcance

El desarrollo de la investigación se enfoca en las áreas de diseño y construcción de software llevando a cabo la aplicación de las guías de enseñanza aprendizaje construidas con base en SWEBOK, con el fin de determinar su validez y pertinencia, mediante la validación estadística de resultados. También se realizó la propuesta de espacios locativos y herramientas tecnológicas, que según el análisis realizado se consideraron más adecuadas, por parte del autor y director del proyecto.

La investigación no cubre otras áreas del conocimiento de la guía SWEBOK como requisitos, pruebas, mantenimiento, gestión de la configuración, gestión de la ingeniería u otros, por corresponder algunos de ellos, a otros subproyectos.

1.3 Aspectos teóricos a tener en cuenta

Para atender los diferentes frentes de un entorno aprendizaje integral para la enseñanza de las áreas de diseño y construcción de software se tuvieron en cuenta componentes como contenidos direccionados por la guía SWEBOK, estrategias de enseñanza apropiadas, modelos de evaluación adecuados, herramientas tecnológicas pertinentes y ambientes escolares propicios.

Guía SWEBOK [7]. Documento que recopila los esfuerzos realizados para definir el cuerpo de conocimiento de la ingeniería de software. No debe confundirse con el cuerpo del conocimiento mismo, que ya existe en la literatura.

Sus objetivos fundamentales son: 1. Promover una visión consistente de la ingeniería de software en todo el mundo; 2. Aclarar el lugar y establecer los límites de la ingeniería de software con respecto a otras disciplinas como la informática, la gestión de proyectos, la ingeniería informática y las matemáticas; 3. Caracterizar el contenido de la disciplina de la ingeniería de software; 4. Proporcionar un acceso temático al cuerpo de conocimiento de la ingeniería de software; 5. Proporcionar fundamentos para el desarrollo curricular y para la certificación individual y material de licencias.

Las áreas de conocimiento (KAs) que conforman la estructura SWEBOK son: Requisitos de software; diseño de software; construcción de software; pruebas de software; mantenimiento de software; gestión de configuración de software; gestión de ingeniería de software; proceso de ingeniería de software; instrumentos y métodos de ingeniería de software; calidad de software.

Diseño de software. Según el glosario IEEE de Términos de Ingeniería de Software [8] se define como “el proceso para definir la arquitectura, los componentes, las interfaces, y otras características de un sistema o componente”, y “como el resultado de este proceso”. Se descompone en los siguientes subprocesos [7]:

Diseño de la arquitectura o de alto nivel, en el cual se describe cómo descomponer el sistema y organizarlo en los diferentes componentes.

Diseño detallado, en el que se describe el comportamiento específico de cada uno de los componentes de software identificados.

Los tópicos contemplados corresponden a: Fundamentos de diseño; aspectos clave; estructura y arquitectura de software; análisis y evaluación de la calidad; notaciones de diseño; estrategias y métodos de diseño de software.

Construcción de software. Según la IEEE [8] codificar es el proceso de expresar un programa de computadora en un lenguaje de programación. Construir, según SWEBOK [7], consiste en crear software mediante una combinación de codificación, verificación, pruebas unitarias y depuración.

Los tópicos contemplados corresponden a: Fundamentos de construcción; gestión de la construcción; consideraciones prácticas.

Taxonomía de Bloom [9]. Es una clasificación muy conocida y ampliamente utilizada en la definición de los objetivos educativos cognitivos. Para el caso de la ingeniería de software se refiere al conocimiento *generalmente aceptado*. Para las áreas de diseño y construcción de software se cuenta con la clasificación correspondiente para cada tópico, según lo establecido en el apéndice D de la guía SWEBOK [7]. Contiene 6 niveles, que se describen brevemente.

Conocimiento (K): Recuerda datos.

Comprensión (C): Entendimiento del significado, traslación, interpolación e interpretación de instrucciones y problemas, explica un problema con sus palabras.

Aplicación (AP). Usa un concepto en una nueva situación o usa una abstracción espontánea; aplica lo aprendido en el salón de clase a situaciones nuevas.

Análisis (AN). Separa material o conceptos en partes de componentes de modo que su organización estructural puede ser entendida; distingue entre hechos e inferencias.

Síntesis (S). Construye una estructura o patrón de diversos elementos; coloca partes juntas dentro de un todo, con énfasis en crear un nuevo significado o estructura.

Evaluación (E). Hace juicios sobre el valor de las ideas o materiales.

Estrategias de enseñanza aprendizaje. Según Sandoval [2], se analizaron estrategias con base en experiencias de la UPTC e internacionales [10], [11], [12]. La siguiente

figura presenta un compendio de algunas de las identificadas para los niveles que aplican para las áreas de diseño y construcción de software, según SWEBOK.

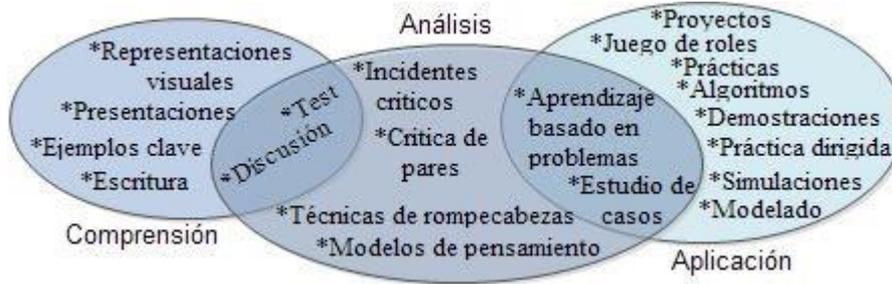


Fig. 1. Estrategias pedagógicas de comprensión, aplicación y análisis. Fuente: [2], [10], [11], [12]

Una vez analizadas, fueron seleccionadas las consideradas más apropiadas para el tópico a tratar.

Tabla 3. Relación entre tópicos SWEBOK, estrategias enseñanza aprendizaje y taxonomía de Bloom (TX) para diseño y construcción de software. Fuentes: Tópicos y nivel taxonómico (TX): Anexo D [7]; Estrategia [2]

Diseño de software		
Tópicos	Estrategia	TX
Fundamentos de diseño de software	Discusiones y debates	C, AN
Aspectos clave de diseño de software	Aprendizaje inductivo	AP
Estructura del software y arquitectura	Proyectos	AP, C
Análisis y evaluación de la calidad	Presentaciones	C, AN
Notaciones de diseño de software	Representaciones visuales	AP
Estrategias y métodos de diseño	Test	AN, AP, C
Construcción de software		
Tópicos	Estrategia	TX
Fundamentos de construcción de software	Técnica de rompecabezas	AN, AP
Gestión de la construcción	Algoritmos	C, AP
Consideraciones prácticas	Juego de roles	AN, AP

Proyecto Académico Educativo-PAE. Documento que tiene como propósito orientar y definir políticas de desarrollo académico del programa de Ingeniería de Sistemas y Computación. La malla curricular está conformada por las siguientes áreas.

Área general, se entiende como la integración de saberes y prácticas que complementan la formación integral, axiológica y cultura.

Área interdisciplinar, se entiende como los saberes, las competencias y las prácticas afines y próximas que comparten varios programas académicos.

Área disciplinar y de profundización, se entiende como los saberes, competencias y prácticas que determinan el perfil estricto y específico de un programa académico que define una profesión.

Las áreas del conocimiento de ingeniería de software se ubican en el área disciplinar y profundización, la figura 2 muestra la relación con SWEBOK.

Modelo de evaluación de aprendizajes y competencias [13]. La educación basada en competencias comenzó a realizarse en los años treinta del siglo XX en Estados Unidos, pero el interés en esta técnica aumentó recientemente ya que se identificó que podría ser apropiada para dar respuesta a la necesidad de adecuar la educación a las exigencias de la industria [14].

El término competencia tiene una amplia gama de definiciones pero la que más se ha generalizado es: “Saber hacer en un contexto” [14].

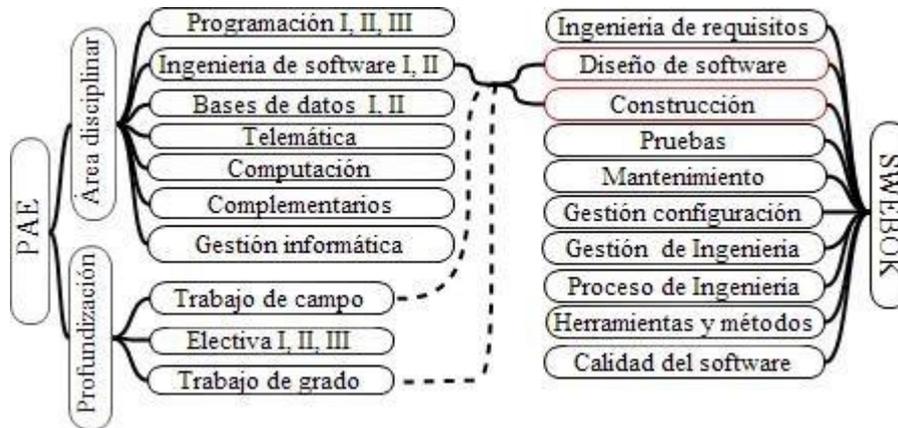


Fig. 2. PAE vs. SWEBOK. Fuente: Autores

Dentro de las competencias básicas asumidas por el ICFES [15] para evaluar a los estudiantes próximos a graduarse y que son requeridas por el Ministerio de Educación para la formación y el desempeño profesional como estándares mínimos de calidad son [16]: competencias interpretativa, argumentativa y propositiva.

Para construir una competencia se deben tener en cuenta los siguientes pasos [9]:

Tomar el *verbo u operación*, el cual implica señalar la cualidad de la acción, para cuyo fin se utiliza la taxonomía de objetivos.

Indicar el *objeto*, es decir, precisar los contenidos curriculares conformados por el aspecto cognitivo, procedimental y/o actitudinal.

Complementar con la *condición o situación de contexto*, que se refiere a la aplicación personal, social, académica y/o laboral.

Ejemplo: Manifiesta – perseverancia e interés – en el logro de los objetivos.

Entorno de aprendizaje [17]. Un entorno o ambiente de aprendizaje se puede definir como un espacio en el que los estudiantes interactúan, bajo condiciones y circunstancias físicas, humanas, sociales y culturales propicias, para generar experiencias de aprendizaje significativo y con sentido.

En el marco del desarrollo de competencias, se encamina a la construcción y apropiación de un saber que pueda ser aplicado en las diferentes situaciones que se le presenten al individuo en la vida y las acciones que éste puede realizar en la sociedad.

Debe fomentar el aprendizaje autónomo, dando lugar a que los sujetos asuman la responsabilidad de su propio proceso de aprendizaje y generar espacios de interacción entre los estudiantes en los cuales el aprendizaje se construya conjuntamente.

Marco Conceptual. Se tomaron como referencia conceptos de ingeniería de software, diseño y construcción de software [8], [18], [19], [20], [21] y la Norma Técnica Colombiana NTC 4595 [22], cuyo objeto es establecer los requisitos para el planeamiento y diseño físico-espacial de nuevas instalaciones escolares, orientado a mejorar la calidad del servicio educativo en armonía con las condiciones locales, regionales y nacionales.

1.4 Antecedentes del currículo en la ingeniería de software

El término software fue acuñado por el estadístico John Turkey en 1958 y el término ingeniería de software fue utilizado por primera vez en una conferencia de la OTAN llevada a cabo en Alemania en 1968 [7].

En los comienzos de la ingeniería de software, cuando ésta aún no se conocía con este nombre, la principal preocupación era ejecutar instrucciones para automatizar procesos en grandes mainframes [23], [24], [25]. Las universidades se vieron en la necesidad de incluir áreas relacionadas con la programación de algoritmos inicialmente con un enfoque matemático.

En la medida en que se evolucionó hacia tecnologías cada vez más complejas, fue necesario proveer un orden y darle estructura al desarrollo de los sistemas de información. Organismos a nivel mundial como la IEEE [26], el SEI [27] y las universidades comenzaron a desarrollar metodologías que permitieran responder con mayor calidad a los requisitos a nivel de software. En 1998, la IEEE publicó la norma internacional ISO/IEC12207 [28] para el ciclo de vida del software.

Sólo hasta el año 2000 se observa unificación de criterios entre los diferentes organismos encargados de estructurar el currículo de la ingeniería de software. En 2001 la IEEE publicó la primera versión de la Guía del Cuerpo de Conocimiento de la Ingeniería de Software SWEBOK [7], diseñado para caracterizar la disciplina de la ingeniería de software y proporcionar una guía describiendo el conocimiento

generalmente aceptado. La guía fue actualizada a la versión 3.0 [29] en el año 2014 para adaptarse a los nuevos cambios tecnológicos.

En la literatura consultada desde comienzos de 2000 se observa la necesidad de incorporar en el currículo académico de la ingeniería de software, nuevas tendencias, como las tecnologías para dispositivos móviles tanto dentro de los contenidos académicos como herramienta para la enseñanza [30], también se ha popularizado el uso de tecnologías en la nube [31], la computación orientada a servicios [32] y lenguajes de desarrollo como Ruby y Python [33].

Para ampliar la información sobre este apartado se sugiere al lector consultar el artículo “Ingeniería de software: evolución de la enseñanza” [34], producto de la investigación [3].

1.5 Metodología

La investigación [3] se llevó a cabo mediante el desarrollo de los objetivos específicos (o.e) y actividades descritos en la tabla 4.

Tabla 4. Metodología [3]

Fase 5. Evaluación de estrategias planteadas	
o.e. 1. Definir el modelo de medición para evaluar los resultados de la aplicación de las estrategias metodológicas propuestas en las guías de enseñanza aprendizaje.	Actividades: Determinación del modelo de evaluación a utilizar, construcción de los instrumentos de evaluación y definición de la evaluación.
o.e. 2. Implementar las estrategias metodológicas propuestas en las guías de enseñanza aprendizaje con estudiantes de las asignaturas de diseño y construcción de software de la EISC.	Actividades: Preparación para la implementación de estrategias metodológicas, implementación de las estrategias metodológicas y aplicación de la evaluación.
Fase 6: Síntesis de herramientas y estrategias metodológicas	
o.e. 3. Analizar el impacto de las guías de enseñanza aprendizaje implementadas con estudiantes, aplicando el modelo de evaluación, para determinar las estrategias metodológicas válidas.	Actividades: Obtención de resultados, análisis y síntesis de resultados obtenidos de la evaluación.
Fase 7: Planteamiento de aspectos de infraestructura y laboratorios	
o.e. 4. Determinar los principales aspectos necesarios para el diseño de la infraestructura física (espacios locativos y herramientas tecnológicas) para el desarrollo de los componentes metodológicos propuestos para el mejoramiento del proceso enseñanza aprendizaje de los contenidos de diseño y construcción de software postulados en la guía SWEBOOK.	Actividades: Determinar herramientas tecnológicas para diseño y construcción de software y aspectos de infraestructura física.

2 Desarrollo de la Investigación

A continuación se presenta la ficha técnica definida para el desarrollo del proyecto “Evaluación de estrategias metodológicas y propuesta de infraestructura para enseñanza de diseño y construcción de software”.

Tabla 5. Ficha técnica [3]

Ficha técnica	
Tipo de investigación	Investigación mixta en una combinación investigación documental e investigación de campo [35].
Ámbito Geográfico	UPTC, Boyacá, Colombia.
Universo	Estudiantes de la EISC de sexto semestre del año 2014.
Metodología de selección de la muestra	Muestreo por conveniencia [36].
Tamaño muestral	Un curso de estudiantes de la EISC de los tres impartidos en el segundo semestre del año 2014, dos en la ciudad de Tunja y uno en la ciudad de Sogamoso.
Margen de confianza	95%
Método de investigación	Método analítico-sintético

2.1 Resultados

Para llevar a cabo el desarrollo del proyecto se ejecutaron las actividades descritas en la metodología, obteniéndose los siguientes resultados.

En la fase 5, se analizaron modelos de evaluación como: modelo de evaluación por objetivos [37], modelo de evaluación U.T.O de Cronbach [38], modelo de evaluación CIPP de Stufflebeam [38], modelo de evaluación de metas de Scriven [38] y modelo de evaluación de aprendizajes y competencias [9]. Se realizó un comparativo teniendo en cuenta aspectos como: objetivo de la evaluación, concepto evaluativo, fases principales de la aplicación, metodología evaluativa, tipo de información a recoger, tipo de informe de resultados, utilidad de los hallazgos, pertinencia y actualidad.

Se consideró que el modelo de evaluación de aprendizajes y competencias es el más adecuado debido a que contempla, además de la adquisición de conocimientos, aspectos tales como comportamientos, actitudes y valores. Adicionalmente se complementa con la taxonomía de Bloom y va alineado con el contexto actual, ya que permite preparar al estudiante para desenvolverse satisfactoriamente en el Examen de Estado de Calidad de la Educación Superior [39], que en uno de sus objetivos busca comprobar el grado de desarrollo de las competencias de los estudiantes próximos a culminar los programas académicos de pregrado de las instituciones educativas.

Una vez seleccionado el modelo de evaluación se definió la escala de evaluación a aplicar, tomando como referencia la sugerida por el Ministerio de Educación Nacional

en el Decreto 1290 de 2009 [40] para las competencias interpretativas, argumentativas y propositivas, utilizando una escala de 1.0 a 5.0 y la escala de Likert [41] para las competencias comportamentales.

Posteriormente se procedió a construir los instrumentos de evaluación, autoevaluación y la evaluación a aplicar por cada uno de los tópicos. Se seleccionó el estadístico T de Student como modelo para análisis de datos. Los elementos mencionados constituyen el modelo de medición utilizado.

Se procedió a aplicar las guías de enseñanza aprendizaje, recopilar la autoevaluación y realizar la ponderación de los resultados de la evaluación [3].

En la fase 6 se tabularon los datos obtenidos a través de los instrumentos de evaluación y autoevaluación, se aplicó el estadístico T de Student y se diagramaron los resultados.

A manera de ejemplo se presenta el resultado del tópico “Fundamentos de diseño de software” [3], en donde se definieron las siguientes hipótesis.

Hipótesis nula. Con la aplicación de las estrategias basadas en la guía SWEBOK se obtienen calificaciones iguales a 4.0.

Hipótesis alternativa. Con la aplicación de estrategias basadas en los contenidos de la guía SWEBOK se obtienen resultados superiores a 4.0.

El resultado de la aplicación del modelo estadístico se presenta en la tabla 6 y figura 3.

Con los resultados obtenidos en esta fase se logró demostrar la hipótesis alternativa, que permite concluir que con la aplicación de estrategias de enseñanza aprendizaje basadas en contenidos de la guía SWEBOK se obtienen resultados superiores a 4.0 en cuanto al componente cuantitativo, utilizando un modelo de evaluación de aprendizajes y competencias [3].

En cuanto al componente cualitativo se evaluaron las competencias comportamentales y actitudinales, observándose que es necesario reforzar en los estudiantes principalmente la competencia de “Comunicación asertiva en el entorno social”, debido a que en las diferentes pruebas realizadas fue en la que se identificó mayor debilidad.

La fórmula del estadístico T de Student es la siguiente:

$$t = \frac{\bar{x} - \mu_0}{\hat{s} / \sqrt{n-1}} \quad (1)$$

Tabla 6. Resultados obtenidos para el tópico fundamentos de diseño de software [3]

T de Student			
Variables	Valores	Variables	Valores
\bar{x} (valor de la media)	4,3	μ_0 (hipótesis a evaluar)	4,0
\hat{s} (desviación típica)	0,2222876	n (tamaño muestra)	18
gl (n-1) (grados libertad)	17	α (nivel significancia)	0,05

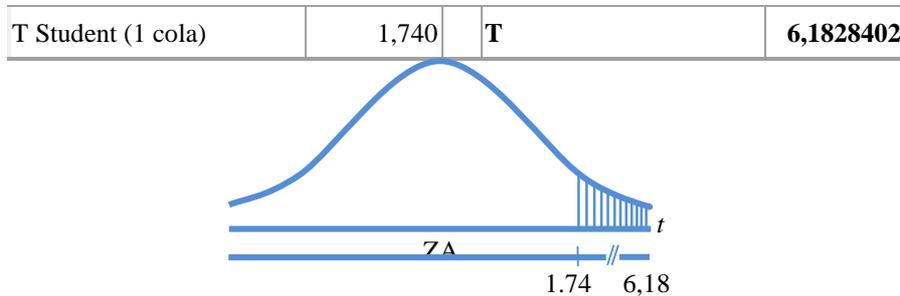


Fig. 3. Resultado T Student para el t3pico Fundamentos de dise1o de software [3]

En el desarrollo de la fase 7, analizadas las herramientas de dise1o de software se determin3 que las herramientas open source, aunque pueden resultar aceptables en un entorno acad3mico, son b3sicas y carecen de mayores prestaciones. Las herramientas comerciales son m3s elaboradas y existen diferencias entre ellas, encontr3ndose que la m3s completa de acuerdo a las caracter3sticas analizadas, es Visual Paradigm, puesto que cumple con todos los diagramas UML 2.0, es muy amigable, f3cil de usar y en la comparaci3n de precios con respecto a otros, teniendo en cuenta sus prestaciones, tambi3n resulta favorable, aunque tambi3n se podr3a optar por las otras opciones, como Enterprise Architect o StarUML.

En el an3lisis realizado sobre herramientas de construcci3n de software se determin3 que las cinco m3s adecuadas para el entorno acad3mico en el 3rea de ingenier3a de software de la UPTC son: Java, Python, C, PHP y C#.

Finalmente, como resultado de la consulta de normas y recomendaciones para la construcci3n de espacios de infraestructura f3sica se determin3 que los apropiados para la UPTC son aulas de clase, laboratorios, salas de reuniones de diferentes modelos, 3reas de estudio y espacios de esparcimiento con dise1os apropiados.

2.2 Discusi3n

La UPTC interesada en el mejoramiento continuo de las 3reas relacionadas con la ingenier3a de software propone el macroproyecto “Propuesta metodol3gica para el desarrollo de contenidos del 3rea de Ingenier3a de Software en la escuela de Ingenier3a de Sistemas y Computaci3n de la UPTC, a partir de la gu3a SWEBOK”.

SWEBOK puede considerarse como un documento de gran valor para la construcci3n de un curr3culo de ingenier3a de software, porque provee los lineamientos en los temas claves que debe contener.

Para lograr excelentes resultados, adem3s de las estrategias de ense1anza aprendizaje basadas en SWEBOK, es necesario contar elementos adicionales que permitan construir entornos integrales, tales como los instrumentos para medir los resultados y los espacios de infraestructura y laboratorios que faciliten la apropiaci3n y transmisi3n del conocimiento. El desarrollo del proyecto “Evaluaci3n de estrategias metodol3gicas y propuesta de infraestructura para ense1anza de dise1o y construcci3n

de software” [3], permite coadyuvar en el logro de los fines principales del macroproyecto.

En la figura 4 se observa la interacción de los diferentes elementos constitutivos del entorno integral propuesto por los autores.

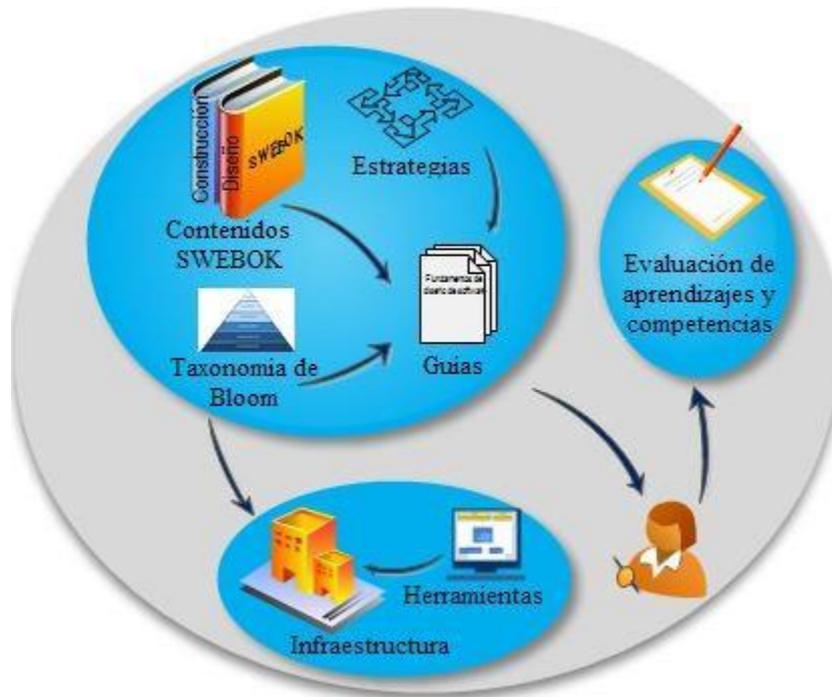


Fig. 4. Entorno integral de aprendizaje. Fuente: Autores

3 Conclusiones

En la ejecución de la segunda etapa del macroproyecto, se llevó a cabo el desarrollo del proyecto “Evaluación de estrategias metodológicas y propuesta de infraestructura para enseñanza de diseño y construcción de software” [3], aplicando las estrategias metodológicas resultado de la primera etapa, a un grupo de estudiantes de la UPTC, utilizando como modelo de evaluación el de aprendizajes y competencias.

Como producto de esta actividad se obtuvieron resultados satisfactorios evidenciando que las estrategias metodológicas aplicadas permiten obtener calificaciones superiores a 4.0 en una escala de 0.0 a 5.0, logrando no solamente resultados cuantitativos sino también cualitativos, que son de considerable importancia para el desenvolvimiento en el entorno laboral del futuro egresado.

La articulación de un conjunto de elementos que constituyan un entorno de enseñanza aprendizaje integral que incluyan estrategias basadas en contenidos de la guía SWEBOOK, modelos de evaluación adecuados, herramientas de software

apropiadas e infraestructura física basada en normas técnicas, permitirá a la UPTC lograr una educación de alta calidad y subsanar las falencias que presenta la transmisión y apropiación del conocimiento del programa actual.

Para complementar las estrategias metodológicas propuestas se recomienda la implementación de laboratorios, utilizando técnicas de coaching que permitan a los estudiantes aplicar sus conocimientos, acompañados por estudiantes de niveles superiores, para que desarrollen sus habilidades y profundicen sus conocimientos en un ambiente que simule el entorno laboral actual.

La guía SWEBOK es un documento de gran valor que debería ser tenido en cuenta en la construcción curricular de los programas de ingeniería de software de los entes educativos, por contener el conocimiento generalmente aceptado para estas áreas del conocimiento.

El actual PAE (2009) de la EISC de la UPTC contempla las asignaturas de ingeniería de software I y II, que sólo permiten cubrir algunas de las áreas de conocimiento de la guía SWEBOK, como son requisitos, diseño, construcción y pruebas de software. Luego del estudio realizado se recomienda al Comité de Currículo de la EISC de la UPTC, única entidad autorizada para proponer cambios en el currículo del programa, incluir una electiva que cubra las otras áreas del conocimiento bajo una denominación como “Gestión de proyectos de ingeniería de software” de forma que se dé la posibilidad al estudiante de profundizar en estos temas, si son de su interés dentro de su formación profesional; adicionalmente actualizar los contenidos temáticos de las asignaturas según los parámetros consignados en el informe de la presente investigación.

Los resultados de la investigación realizada pueden ser tomados como referencia por otros entes de educación con programas iguales o similares, ya que constituyen un aporte que permite mejorar los procesos de enseñanza aprendizaje de las áreas de diseño y construcción de software.

Agradecimientos

Este trabajo fue inscrito en la DIN (Dirección de Investigaciones de la Universidad Pedagógica y Tecnológica de Colombia) por parte de GIS (Grupo de Investigación en Software).

Los autores desean expresar su agradecimiento a la Universidad Pedagógica y Tecnológica de Colombia, a la Escuela de Ingeniería de Sistemas y Computación y al Grupo GIS, por la colaboración prestada a lo largo del desarrollo del trabajo de titulación y sus productos generados, dentro de los que se encuentra el presente documento.

Referencias

1. Universidad Pedagógica y Tecnológica de Colombia, <http://www.uptc.edu.co/>
2. Sandoval, J: Trabajo de grado Propuesta pedagógica y metodológica basada en SWEBOOK para la enseñanza de diseño y construcción de software. Tunja (2013)
3. Chuquín, S: Trabajo de grado Evaluación de estrategias metodológicas y propuesta de infraestructura para enseñanza de diseño y construcción de software. Tunja (2015)
4. Departamento Nacional de Planeación: Bases del Plan Nacional de Desarrollo 2014-2018
5. Universidad Pedagógica y Tecnológica de Colombia: Proyecto Académico Educativo PAE (2009)
6. Universidad Pedagógica y Tecnológica de Colombia: Procedimiento de autoevaluación de programas: Instructivo taller de gradación de cumplimiento y emisión de juicios. Guía 3, http://virtual.uptc.edu.co/acreditacion/MODELO/ANEXOS/GUIASYFORMATOS/GUIA_S/GUIA_03_GRADACION_JUICIOS.pdf
7. Abran, A. et al.: Guía del cuerpo de conocimiento de la Ingeniería de Software–SWEBOOK. IEEE, Los Alamitos, CA (2004)
8. Institute of Electrical and Electronics Engineers-IEEE: 610.12-1990-IEEE Glosario Estándar de Terminología de Ingeniería de Software (en inglés) (1990)
9. Santivañez, V: Diseño curricular a partir de competencias. Ediciones de la U, Bogotá, Colombia (2013)
10. National Employee Development Center: Instrucción Proceso de Diseño: Estrategias sugeridas de instrucción para el uso con cada nivel de Bloom (en inglés), <ftp://ftpfc.sc.egov.usda.gov/NEDC/isd/taxonomy.pdf>
11. University of Northern Iowa-College of Education: Taxonomía de Bloom-Revisada, palabras clave, modelo de preguntas y estrategias de instrucción (en inglés), <http://www.uni.edu/coe/sites/default/files/wysiwyg/BloomRevisedTaxonomy.pdf>
12. Harris, P., Johnson, R.: Enseñanza no tradicional y estrategias de aprendizaje (en inglés), <http://www.montana.edu/teachlearn/Papers/activelearn2.pdf>
13. Castillo, S, Cabrerizo, J: Evaluación educativa de aprendizajes y competencias. Pearson Educación, Madrid, España (2010)
14. Álvarez, R: Formación superior basada en competencias, interdisciplinaria y trabajo autónomo del estudiante. Revista Iberoamericana de Educación, 8
15. Instituto Colombiano para el Fomento de la Educación Superior, <http://www.icfes.gov.co/>.

16. Ministerio de Educación Nacional de Colombia. Decreto 3963 de 14 de octubre de 2014 17. Ministerio de Educación Nacional de Colombia. ¿Qué es un ambiente de aprendizaje?, <http://www.colombiaaprende.edu.co/html/productos/1685/w3-article-288989.html>
18. Pressman, R: Ingeniería de software: un enfoque práctico. McGraw Hill, México (2010)
19. Sommerville, I: Ingeniería de software. Pearson Educación, Madrid, España (2005)
20. Sánchez, S., Sicilia, M., Rodríguez, D: Ingeniería de Software: Un enfoque desde la guía SWEBOK. Alfaomega, México (2012)
21. Instituto Colombiano de Normas Técnicas y Certificación. Ingeniería de Software y Sistemas: Procesos del ciclo de vida del software NTC 4243. ICONTEC, Bogotá, D.C, Colombia (2009)
22. Instituto Colombiano de Normas Técnicas y Certificación. Ingeniería Civil y Arquitectura. Planeamiento y Diseño de Instalaciones y Ambientes Escolares NTC 4595. ICONTEC, Bogotá, D.C., Colombia (2006)
23. Berger, R. W.: Software para simulación de decisiones financieras (en inglés). ACM SIGSIM Simulation Digest, 4, 3, 31 a 40 (1973)
24. Banner, D. K.: Lo que un gerente financiero debería saber sobre cobol y lenguaje ensamblador (en inglés). Management and computer systems, 293 (1970)
25. Shooman. M. L.: Modelos estructurados para la predicción de la fiabilidad del software (en inglés). En: IEEE Computer Society ICSE'76 Proceeding of the 2nd international conference on Software engineering, pp. 268 a 280 (1976)
26. Instituto de Ingenieros Eléctricos y Electrónicos, <http://www.ieee.org>
27. Software Engineering Institute SEI, <http://www.sei.cmu.edu>
28. IEEE/EIA: Implementación de la industria de la norma internacional ISO/IEC 12207 (en inglés), United States of America. (1998)
29. Bourque, P., Farley, R E: SWEBOK V3.0. IEEE (2014)
30. Maceiras, R, et al: Adaptación de un campo virtual para dispositivos móviles de aprendizaje (en inglés). En: Global Engineering Education Conference, pp. 165 a 167. IEEE (2011) 31. Yang, G., Zhu, Z: La aplicación de computación en la nube basada en SaaS en la investigación universitaria y plataformas de enseñanza (en inglés). En: International Conference on Intelligence Science and Information Engineering, pp. 201 a 213. IEEE (2011)
32. Chen, Y: Orientación al servicio en el currículo de la informática (en inglés). En: 6th International Symposium on Service Oriented System Engineering, pp. 122 a 133, IEEE (2011)
33. Harley, Z., Harley, E. R: Un asistente para la programación informática en E-Learning (en inglés). IEEE, 95 a 98 (2012)
34. Chuquín, S., Otálora, J: Ingeniería de Software: evolución de la enseñanza. En proceso de publicación.
35. Lara, E: Fundamentos de investigación: Un enfoque por competencias. Alfaomega, México (2011)
36. Medina, M: La investigación aplicada a proyectos. Alfaomega, Bogotá, Colombia (2014)
37. Tyler, R.: Principios básicos de currículo e instrucción (en inglés). The University of Chicago Press, Chicago, United States of America (2013)
38. Ruiz, E: Propuesta de un modelo de evaluación para el nivel superior: Una orientación cualitativa. Cuadernos del CESU, México D.F. (1998)
39. Ministerio de Educación Nacional. Conoce los ECAES,

- <http://www.colombiaaprende.edu.co/html/home/1592/article-82458.html> 40.
Ministerio de Educación Nacional: Decreto 1290 de 2009. Colombia (2009)
41. Likert, A: Test psicológicos y evaluación. Pearson Education, México (2003)