

Estado de la publicación: El preprint ha sido publicado como artículo en una revista
DOI del artículo publicado: <https://doi.org/10.37135/chk.002.21.10>

MODELO PARA EL PROCESO DE AUTOFORMACIÓN EN LA ENSEÑANZA APRENDIZAJE DE LA INGENIERÍA DE SOFTWARE

Ivet Espinosa Conde, Raquel Diéguez Batista, Yulkeydis Martínez Espinosa

<https://doi.org/10.1590/SciELOPreprints.6626>

Enviado en: 2023-08-16

Postado en: 2023-08-17 (versión 1)

(AAAA-MM-DD)

MODELO PARA EL PROCESO DE AUTOFORMACIÓN EN LA ENSEÑANZA APRENDIZAJE DE LA INGENIERÍA DE SOFTWARE ***A SELF-TRAINING MODEL FOR THE TEACHING AND LEARNING PROCESS OF SOFTWARE ENGINEERING***

Ivet Espinosa Conde¹, ORCID: <https://orcid.org/0000-0001-5102-2339>

Raquel Diéguez Batista², ORCID: <https://orcid.org/0000-0002-4975-6947>

Yulkeydis Martínez Espinosa³, ORCID: <https://orcid.org/0000-0003-2221-0650>

¹Universidad de Ciego de Ávila Máximo Gómez Báez, Vicerrectoría de Investigación y Posgrado, Grupo de Grados Científicos, Ciego de Ávila, Cuba, email: ivet.conde@gmail.com

²Universidad de Ciego de Ávila Máximo Gómez Báez, Vicerrectoría de Investigación y Posgrado, Grupo de Grados Científicos, Ciego de Ávila, Cuba, email: raquel@unica.cu

³Universidad de Ciego de Ávila Máximo Gómez Báez, Vicerrectoría Docente, Centro de Estudios Educativos, Ciego de Ávila, Cuba, email: yulkeydis@unica.cu

RESUMEN

El presente trabajo se realiza teniendo en cuenta los resultados de la aplicación de instrumentos diagnósticos: observación al proceso de enseñanza aprendizaje de la Ingeniería de Software en la carrera Ingeniería Informática de la Universidad de Ciego de Ávila, cuestionario a estudiantes y entrevista a profesores; donde se revelan limitaciones para la solución de tareas que requieren del enriquecimiento autónomo del conocimiento. La aplicación de métodos científicos como análisis, síntesis, inductivo, deductivo y de sistematización de información, permitió constatar carencias teóricas para la transformación del proceso. Por tanto, el objetivo se direcciona a elaborar un modelo de la dinámica del proceso de autoformación del estudiante en la enseñanza aprendizaje de la Ingeniería de Software, teniendo en cuenta la relación entre la auto-apropiación de contenidos y la auto-concientización. En general, se desarrolló una investigación con un enfoque metodológico exploratorio, descriptivo y explicativo (mixto). El modelo de autoformación obtenido se concreta en la interactividad teórico práctica, desde la lógica integradora que se produce entre la auto-apropiación de contenidos y la auto-concientización que logra el estudiante en la sistematización formativa. El impacto de la propuesta se expresa en su aplicación en la construcción de aportes prácticos para su instrumentación en la práctica docente.

PALABRAS CLAVE: Autoformación, proceso de enseñanza aprendizaje, ingeniería de software.

ABSTRACT

The present work is carried out taking into account the results of the application of diagnostic instruments: the observation of the teaching and learning process of Software Engineering in the Computer Engineering career in the University of Ciego de Ávila, a questionnaire to students and an interview with professors; where limitations revealed the solution of tasks that require the autonomous enrichment of knowledge. The application of scientific methods such as analysis, synthesis, inductive, deductive and information systematization, allowed to verify theoretical deficiencies for the transformation of this process. Therefore, the objective is aimed at developing a model about the dynamics of the student's self-training in the teaching and learning process of Software Engineering, by considering the relationship between self-appropriation of content and self-awareness. In general, an investigation was developed with an exploratory, descriptive and explanatory (mixed) methodological approach. The autonomous training model is materialized in theoretical-practical interactivity, from the logical integration that occurs between the self-appropriation of content and the self-awareness that the student achieves in the formative systematization. The impact of

the proposal is expressed in its application in the construction of practical contributions for its implementation in the teaching practice.

KEYWORDS: *Self- training, teaching and learning process, software engineering*

Recibido: (29/04/2023)

Aceptado: (05/08/2023)

INTRODUCCIÓN

La educación superior cubana tiene entre sus objetivos desarrollar un proceso formativo activo, que tenga como centro al estudiante, donde la creatividad e innovación prevalezcan sobre lo informativo. Por tanto, las universidades buscan métodos para lograr una transformación en los estudiantes que les permita analizar con espíritu crítico, avanzar en la búsqueda del conocimiento y consolidar el carácter desarrollador; logrando convertirse en los profesionales que las sociedades actuales demandan. En este sentido, “las universidades tienen ante sí el inmenso reto de transformarse a sí mismas para estar en capacidad de contribuir sensiblemente en la formación integral de sus estudiantes” (Alarcón et al., 2019, p. 10).

En particular, la formación de ingenieros tiene una importancia significativa en un mundo en el que el desarrollo tecnológico se concreta en todos los aspectos de la vida cotidiana. La sociedad actual necesita ingenieros que respondan a los problemas que ocurren en el entorno social donde se desenvuelven, que logren hacer, resolver y afrontar la realidad desde la experiencia para construir conocimiento (Betanco, 2020). Dentro del área de las ingenierías ocupa un lugar importante la ingeniería informática, que en la sociedad actual apuesta por la informatización de sus procesos.

La carrera Ingeniería Informática se crea con el objetivo de formar profesionales integrales, independientes, responsables y creativos capaces de dar solución a los problemas que enfrentará y las restricciones que puedan presentarse en el medio donde laborará; además de fomentar un espíritu de autosuperación en el estudiante que le permita mantenerse actualizado en los avances de la ciencia y la técnica en su campo profesional (MES, 2017). Para alcanzar estos objetivos, la carrera se estructura a partir de disciplinas y asignaturas donde los estudiantes alcanzan los conocimientos y habilidades para asumir tareas relacionadas con el ejercicio de su profesión.

En particular, el programa de la disciplina Ingeniería y Gestión de Software de la carrera Ingeniería Informática en la Universidad de Ciego de Ávila (Facultad de Ingeniería Informática, 2019) establece que el estudiante debe ser capaz de:

- Analizar un proceso en una organización de base productiva o de servicio o en cualquier otro medio para determinar las tareas a automatizar, los medios técnicos necesarios y la planificación del proyecto.
- Aplicar metodologías para organizar y dirigir el desarrollo, implantación y puesta en marcha de los sistemas informáticos acorde con las tecnologías y paradigmas que se utilicen.
- Aplicar, con un alto nivel de profesionalidad, los principios de la programación en la automatización de cualquier aplicación.
- Desarrollar productos informáticos tanto de manera individual como en equipos.
- Diseñar las estructuras de datos adecuadas para la solución informática de cualquier aplicación.

Para lograr estos objetivos se estructuran diferentes asignaturas como la Ingeniería de Software, que comprende la aplicación práctica del conocimiento en el estudio del contexto del sistema, el análisis, diseño, implementación, pruebas y despliegue o mantenimiento de sistemas informáticos, manteniendo una disciplina de trabajo sistemática (Gómez & Moraleda, 2020). En este sentido, el proceso de enseñanza aprendizaje de la Ingeniería de Software aborda los métodos, técnicas y herramientas en el proceso de desarrollo de software mediante “el uso de metodologías y modelos

para lograr sistemas más eficientes y de mayor calidad, con la documentación necesaria en perfecto orden y en el tiempo requerido” (Delgado & Díaz, 2021, p. 11).

A pesar de las exigencias planteadas a los graduados para que asuman los retos actuales con creatividad, autonomía y compromiso, se pudo comprobar desde la aplicación de cuestionarios y entrevistas a estudiantes y profesores de la carrera Ingeniería Informática, la observación a clases, así como la experiencia académica de los autores (teniendo en cuenta como indicadores de análisis el nivel de precisión en la selección de una metodología de desarrollo de software, rigor en el estudio y análisis previo del contexto para desarrollar un sistema informático, grado de independencia para la realización de los flujos de trabajos propios de las fases ingenieriles de desarrollo de software y de precisión en la selección de las tecnologías de desarrollo de software ante un proyecto real, independencia lograda en el diseño de un sistema informático sin la orientación de los métodos y herramientas a emplear) que en el proceso de enseñanza aprendizaje de la Ingeniería de Software, los estudiantes presentan dificultades para:

- Realizar con rigor e independencia el estudio y análisis previo del contexto para desarrollar un sistema informático.
- Seleccionar una metodología de desarrollo de software ante un proyecto real sin la orientación del profesor.
- Ejecutar con independencia los flujos de trabajos propios de las fases ingenieriles de desarrollo de software sin el apoyo del profesor.
- Seleccionar las tecnologías de desarrollo de software más adecuadas ante un proyecto real, según su propio criterio.
- Diseñar un sistema informático sin la orientación de los métodos y herramientas a emplear.

Estas manifestaciones se sintetizan en las limitaciones que presentan los estudiantes de la carrera Ingeniería Informática, en la Ingeniería de Software para la solución de tareas que requieren del enriquecimiento autónomo del conocimiento.

Entre las causas que pudieran conducir al problema planteado, se encuentran las limitaciones en la concepción didáctica del proceso de enseñanza aprendizaje de la Ingeniería de Software en la carrera Ingeniería Informática desde la utilización de métodos que desde la auto-concientización y auto-apropiación del contenido potencien la autoformación.

El proceso de enseñanza aprendizaje de la Ingeniería de Software se ha realizado tradicionalmente con una estructura de clases presenciales y como evaluación final proyectos prácticos que resuelven una problemática dada, real o no. También, se han utilizado métodos expositivos y problematizadores como los estudios de casos o aprendizaje basado en problemas. Sin embargo, en un mundo tecnológico en constante cambio, se necesitan profesionales centrados en aprender de la experiencia que poseen y construir su propio conocimiento.

El aprendizaje es la base de construcción del futuro de las sociedades y por las características propias de la carrera y el acelerado desarrollo tecnológico no es posible que en la actualidad el estudiante pueda aprender todo lo que necesita utilizando métodos tradicionales. En este contexto de rápida evolución se debe evitar una enseñanza basada en la acumulación de conocimientos, es momento de transformar la forma en la que se aprende para transformar a su vez el mundo (Menéndez, 2020). Ante esta situación “el mundo actual exige nuevas estrategias de formación profesional, que mejoren las prácticas de enseñanza aprendizaje, teniendo en cuenta las características de las nuevas generaciones” (Monroy et al., 2022, p. 16).

Desde esta perspectiva se considera importante potenciar la autoformación como un proceso reflexivo de auto-concientización de los estudiantes sobre la necesidad de su formación y su auto-transformación activa y creadora en la auto-apropiación de contenidos en la sistematización formativa; lo que requiere del reconocimiento por parte del profesor que su rol ha cambiado, no se trata de enseñar contenidos, sino de enseñar a aprender.

En este objeto han investigado autores, entre los que destacan Labra et al. (2006), Guitart et al. (2006), Anaya (2012) e Incencio et al. (2022), los cuales estudian el proceso desde concepciones

teóricas basadas en la importancia de la asignatura y la necesidad de incorporar variantes didácticas para su mejor aprovechamiento.

Otros autores como Mariño et al. (2001) y Ciudad & Soto (2006) presentan medios didácticos como apoyo a la asignatura y su visualización por medio de teleclases. A su vez, varios autores han realizado aportes sobre la incorporación de estrategias didácticas para la enseñanza basada en juegos en la Ingeniería de Software, entre los que destacan Taran (2007), Zapata & Duarte (2008), Zapata, González & Manjarrés (2012), Zapata, Villegas & Arango (2012), Montoya-Suárez & Pulgarín-Mejía (2013); estas variantes se limitan a su uso solamente en el aula y tienen como finalidad estimular al estudiante en el aprendizaje lúdico.

Otros como Naya & Oliva (2015), Linares et al. (2019) y Hernández et al. (2019) realizan estudios más profundos en torno a la ingeniería de software y proponen sistemas informáticos evaluadores y modelos didácticos para fases específicas del desarrollo de software.

Se evidencia investigaciones que conducen a la transformación del proceso de enseñanza aprendizaje de la Ingeniería de Software en la carrera Ingeniería Informática, en particular se realizan propuestas que responden a estudios teóricos y de aplicación de variantes didácticas como complementos en las clases, pero en general, constituyen soportes metodológicos para el desarrollo de actividades docentes, su estructura no potencia suficientemente la autoformación.

METODOLOGÍA

El presente artículo responde a una investigación efectuada en un campo de conocimiento específico con un enfoque mixto; pues se realiza un estudio exploratorio, descriptivo y explicativo (Hernández et al., 2010). Se inicia desde la exploración de un tema poco estudiado, para revelar aportes, sus alcances y limitaciones; luego se describe el objeto de estudio con la finalidad de evidenciar sus propiedades más importantes y termina como explicativa al fundamentarse una lógica del proceso de autoformación en la enseñanza aprendizaje de la Ingeniería de Software, desde el establecimiento de nuevas relaciones.

Se ejecuta un estudio con un muestreo intencional no probabilístico en la carrera Ingeniería Informática de la Universidad de Ciego de Ávila, Cuba, por ser donde se desempeña como docente la autora principal de la presente investigación. La población objeto de estudio estuvo formada por 46 estudiantes de la carrera Ingeniería Informática (16 de 4to año del Curso Regular Diurno, 16 y 14 de 5to y 6to año del Curso por Encuentro), no se selecciona muestra porque la totalidad de estudiantes que reciben el contenido de Ingeniería de Software se considera adecuada para la aplicación de los diferentes instrumentos.

Para obtener información adicional se utilizó como fuente la entrevista aplicada a 16 profesores que pertenecen a la disciplina Ingeniería y Gestión de Software. Los implicados en la investigación ofrecieron su consentimiento para participar y su autorización expresa para divulgar la información resultante.

A partir del enfoque asumido en la investigación se realiza la recolección, análisis e interpretación de datos cualitativos y cuantitativos, sobre la base de una combinación de técnicas e instrumentos contruidos a partir de indicadores para determinar la situación actual de la autoformación en el proceso de enseñanza aprendizaje de la Ingeniería de Software:

- El estudio de fuentes documentales: programa de la disciplina, programa de la asignatura y Plan de estudios de la carrera Ingeniería Informática, permitió corroborar las exigencias que desde estos documentos se establecen para la formación del profesional.
- La observación del proceso permitió corroborar el problema de investigación y caracterizar la situación actual de la autoformación en el proceso de enseñanza aprendizaje de la Ingeniería de Software.
- La encuesta a estudiantes de la carrera Ingeniería Informática que cursan las asignaturas de Ingeniería de Software permitió caracterizar el estado actual del nivel de autoformación del estudiante.

- La entrevista a profesores de la disciplina Ingeniería y Gestión de Software permitió corroborar el problema de investigación y caracterizar la situación actual del proceso.

En general, los resultados evidenciaron insuficiencias en el proceso de enseñanza aprendizaje de la Ingeniería de Software para contribuir al enriquecimiento autónomo del conocimiento del estudiante, que le permita la solución de tareas de manera independiente.

La aplicación de métodos teóricos permitió el análisis de fuentes bibliográficas sobre el proceso de enseñanza aprendizaje de la Ingeniería de Software desde la autoformación. En este sentido, para llegar a elaborar el modelo de autoformación fue necesario la utilización de métodos como análisis, síntesis, inductivo, deductivo; además del método de sistematización, tanto desde la sistematización de información o de datos, como desde la sistematización de experiencias, porque estimula la capacidad creativa de pensar.

Se asumieron las consideraciones de Ramos & Vidal (2016) de sistematización como proceso permanente y acumulativo de creación de conocimientos a partir de las experiencias de intervención de los investigadores en la realidad social del proceso de enseñanza aprendizaje de la Ingeniería de Software.

En el orden epistemológico y metodológico, para la construcción del modelo del proceso de autoformación de los estudiantes en la enseñanza aprendizaje de la Ingeniería de Software como resultado científico, se retoma la concepción científica holístico-configuracional de Fuentes (2009), que a través de su método holístico-dialéctico permite el establecimiento de los eslabones esenciales del proceso modelado, partiendo del reconocimiento de sus configuraciones y las relaciones que se establecen entre estas, que conducen a la transformación del estudiante en su contexto formativo.

Lo holístico en el proceso estudiado por su naturaleza como un todo y a la vez contradictoria de las relaciones, condicionan su movimiento y autodesarrollo; siendo un proceso totalizador donde desde sus eslabones se interrelacionan procesos en diversas direcciones, sin que ello signifique la existencia del todo por encima de las categorías.

Se tuvo en cuenta para la modelación del proceso las ideas de Vigotsky (1987), que plantea que en el marco de un aprendizaje desarrollador se debe tener en cuenta no solo lo que ya ha sido aprendido por el estudiante, lo que ya conoce y puede enfrentar, sino aquello que aún no es capaz de enfrentar por sí.

Para lograr este propósito se retomaron los métodos activos centrados en el estudiante, que problematizan y estimulan la conciencia crítica y transformadora, a la vez que conducen a participar en la organización, conducción y desarrollo de su propio proceso de aprendizaje (Jiménez et al., 2020).

Se tomaron en cuenta las ideas del ilustre pedagogo De la Luz y Caballero (1862, como se citó en Barajas & Ochoa, 2021), donde se reconoce la necesidad de aprender a aprender; lo cual se traduce en la autoformación que logra el estudiante en la sistematización formativa en su vínculo con las técnicas de individualización mediante la aportación de recursos diversos, y donde el docente adopta funciones tutoriales, de dinamizador, de facilitador.

De la Filosofía Marxista Leninista se asumió como sustento la teoría dialéctica materialista del conocimiento, para interpretar, analizar y transformar la realidad y dar respuesta a las contradicciones que se dan en el proceso de enseñanza aprendizaje de la Ingeniería de Software que conlleven a la autoformación del estudiante y con ello la formación de un profesional que demanda la sociedad actual.

RESULTADOS Y DISCUSIÓN

El proceso de enseñanza aprendizaje se entiende como un todo integrado donde el protagonista es el estudiante. Tiene como eje fundamental el vínculo entre lo educativo e instructivo como principio pedagógico esencial en la formación integral de la personalidad del estudiante; al constituir la vía para adquirir conocimientos, habilidades y valores en la sistematización formativa

(MINED, 2001). Este proceso debe desarrollarse desde un enfoque sistémico con la concatenación de todos sus componentes y debe caracterizarse por la flexibilidad, con énfasis en el estudiante activo en el proceso, donde el profesor deja de ser el protagonista y asume un rol de guía (Trujillo, 2019).

En el proceso de enseñanza aprendizaje de la Ingeniería de Software se establece un sistema de relaciones desde los diferentes escenarios en que se desarrolla. Para lograr que su estructuración potencie y amplíe la Zona de Desarrollo Próximo (Vigotsky, 1987), es imprescindible tener en cuenta no solo lo que ya ha sido aprendido por el estudiante sino lo que desconoce pero tiene las herramientas y estrategias para resolver.

El Ministerio de Educación Superior en Cuba (MES, 2022) establece la necesidad de formar profesionales de perfil amplio, que posean una profunda formación básica y básica-específica de la profesión para resolver con independencia y creatividad los problemas generales y frecuentes que se presentan en el ejercicio de su profesión. En concreto, la carrera Ingeniería Informática tiene entre sus objetivos formar ingenieros independientes, responsables y creativos capaces de dar solución a los problemas que enfrentará, además de fomentar un espíritu de autosuperación que le permita mantenerse actualizado en los avances de la ciencia y la técnica en su campo profesional. En general, se establece la necesidad de que el estudiante sea capaz de lograr autonomía en su formación académica para enfrentarse a un entorno donde la tecnología evoluciona constantemente.

Para lograr tales fines las universidades buscan desarrollar un proceso de enseñanza aprendizaje utilizando métodos activos caracterizados por otorgar el rol de mediador del aprendizaje al docente, desarrollar en el estudiante habilidades investigativas, promover la adquisición de conocimientos de manera autónoma, estimular las relaciones sociales y las habilidades de comunicación, al promover el análisis, la reflexión y la argumentación sobre lo que se hace y cómo se hace.

De esta forma, el proceso de enseñanza aprendizaje se desarrolla a través de “actividades y procedimientos basados en el enfrentamiento a situaciones que requieren de la construcción de conocimientos y desarrollo de habilidades que le permitan implementar estrategias y tomar decisiones; de esta forma se convierte en un participante activo del proceso” (Guamán & Espinoza, 2022, p. 9).

“El aprendizaje activo tiende a convertirse en aprendizaje significativo siempre y cuando parta de una experiencia previa del alumno y que haya sido de su interés” (Enríquez, 2021, p. 5). Para enfrentar alguna situación en concreto se requiere que el estudiante lleve a cabo diferentes actividades para establecer relaciones entre lo nuevo y lo que ya sabe, es decir, reformular, diferenciar, descubrir, ordenar, clasificar, jerarquizar, relacionar, integrar, comprender y resolver problemas.

En este proceso se hace necesario que el estudiante tenga una participación activa y lo conduzca a aprender durante toda la vida; a que sea partícipe de su propio conocimiento y lo enriquezca en el tiempo mediante la auto-concientización sobre la necesidad de su formación y su auto-transformación activa y creadora en la sistematización formativa, como vía epistemológica y metodológica mediante la profundización del contenido que le permita lograr un aprendizaje desarrollador y transformador.

La sistematización formativa es el proceso que desarrolla el carácter de continuidad y consecutividad, a niveles superiores en la construcción científica del contenido socio cultural por el sujeto y en el que a partir de la apropiación de la cultura se significan factores y criterios que propician la reestructuración epistemológica de ese contenido y con ello su sistematización. (Fuentes, 2009, p. 132)

En relación con la idea anterior, durante la ejecución del proceso de enseñanza aprendizaje de la Ingeniería de Software, la sistematización formativa constituye el evento a través del cual transcurre el proceso, donde los estudiantes y profesores, en sus relaciones configuran los conocimientos, habilidades, valores y valoraciones, potenciando sus capacidades transformadoras profesionales y sociales. El profesor solo debe guiar la actividad y el estudiante convertirse en el centro como ente

activo participativo. El profesor enseña a aprender, al sistematizar las habilidades generales en cada tarea planteada, y el estudiante al ejercitarlas las adquiere, como resultado de la auto-concientización y la auto-apropiación de los contenidos.

Los procesos de auto-concientización y auto-apropiación de contenidos permiten que el estudiante sea activo en su proceso cognitivo, porque está consciente de la necesidad de su formación y utiliza estrategias propias para la apropiación de lo estudiado; por tanto, el proceso de enseñanza aprendizaje adquiere una orientación didáctica activo-transformadora. De esta forma, el estudiante una vez que haga conciencia de estos aspectos y comprenda el proceso que está realizando, logra trazarse estrategias de regulación del aprendizaje que propicien de manera eficaz su propio desarrollo y que a su vez las incorpore a su estilo de vida (Veloso & Veloso, 2022).

Coincidiendo con Morales et al. (2018) en el proceso de auto-apropiación del contenido el estudiante hace suyo los conocimientos, habilidades y valores, así como los mecanismos a través de los cuales lo logra, incorporándolos a su crecimiento personal. En este proceso el estudiante construye, elabora, enriquece y transforma lo que va aprendiendo contribuyendo a la creatividad y su propia formación profesional.

Un proceso de enseñanza aprendizaje donde el estudiante tome conciencia de la necesidad de su transformación estimula a su vez la auto-apropiación de los contenidos, como base para su sistematización y generalización. En este sentido, durante la auto-apropiación de contenidos en la medida que el estudiante concientiza y comprende lo que va aprendiendo, el aprendizaje se hace duradero; por tanto, se considera una condición necesaria, aspiración principal, instrumentación y criterio de calidad en el proceso de enseñanza aprendizaje. En consecuencia, todo aprendizaje donde exista una comprensión consciente y consecuente en la auto-apropiación de contenidos determina en la retroalimentación continua un aprendizaje en menor tiempo y más duradero.

En el proceso de enseñanza aprendizaje de la Ingeniería de Software en la carrera Ingeniería Informática se considera necesario tener en cuenta los criterios anteriores como proceso formativo desarrollador, donde el estudiante bajo la conducción del profesor, desde la auto-concientización y auto-apropiación del contenido transita por diferentes niveles de autoformación desde la sistematización formativa, para dar solución a problemas profesionales que requieran del enriquecimiento autónomo del conocimiento, en el transcurso de la carrera y en su desempeño profesional.

Estas propuestas, aunque contribuyen al desarrollo didáctico de la asignatura, responden a estudios teóricos y de aplicación de variantes didácticas como complementos en las clases. Evidenciándose su evolución desde la utilización de métodos tradicionales dirigidos a la transmisión del conocimiento por parte del profesor, hasta la incentivación del papel protagónico del estudiante, pero aún con sesgos en la necesidad de su autoformación.

Por tanto, se demuestra carencia en el desarrollo de variantes con un sustento metodológico dirigidas a la autoformación desde las diferentes formas de enseñanza de la Ingeniería de Software, toda vez que constituye un proceso complejo, donde el estudiante en el ejercicio de su profesión se enfrenta a un mundo tecnológico en constante cambio. Además, en el que se requiere del establecimiento de principios y métodos para lograr desarrollar software con la calidad requerida, elemento fundamental en su desarrollo profesional.

En relación con el diagnóstico realizado, se identificaron insuficiencias provocadas por la contradicción entre las exigencias que se precisan en los documentos rectores de la carrera Ingeniería Informática para la solución de tareas de Ingeniería de Software con autonomía y rigor académico y la auto-preparación de los estudiantes de la carrera Ingeniería Informática para el desarrollo de software con calidad.

Las encuestas realizadas a los estudiantes y entrevistas a profesores que pertenecen a la disciplina de Ingeniería y Gestión de Software revelaron los siguientes resultados:

- El 96 % de los estudiantes manifiesta estar en desacuerdo con la independencia lograda para la realización de los flujos de trabajos propios de las fases ingenieriles de desarrollo de software, lo que coincide con lo expresado por el 100 % de los profesores.
- En cuanto al rigor con que realizan el estudio del contexto y el proceso donde se desarrolla el sistema más del 50 % está en desacuerdo. Los estudiantes realizan el estudio del contexto, pero

no con el rigor necesario, lo cual se confirma en la entrevista a los profesores, que respondieron el 100 % como no adecuado.

- El 100 % de los estudiantes está en desacuerdo con la preparación que tienen para seleccionar con exactitud una metodología de desarrollo de software. En tal sentido, el 100 % de los profesores lo evalúa de poco adecuado y fundamentan que en la preparación del estudiante realizar esta actividad resulta imprescindible para desarrollar un software con calidad.
- Más del 50 % de los estudiantes manifiesta su desacuerdo sobre la precisión que alcanzan para la selección de las tecnologías de desarrollo de software; lo que se corresponde con lo manifestado por el 100 % de profesores.
- En cuanto al grado de independencia que logran los estudiantes en el diseño e implementación de un sistema informático sin la orientación de los métodos y herramientas a emplear, el 91 % manifiesta estar en desacuerdo. Los profesores coinciden en el 100 % de evaluar este aspecto de poco adecuado.
- Más del 50 % de los estudiantes reconoce que en ocasiones se les dificulta el uso de métodos que los preparen para la solución de problemas de manera independiente. Esta situación es reconocida por el 100 % de profesores.
- El 100 % de los profesores reconoce que orienta con un nivel de bastante adecuado a los estudiantes, tareas que requieren de la búsqueda de conocimientos no tratados en clases. Orientan la realización de manera independiente de los flujos de trabajos propios de las fases ingenieriles de desarrollo de software, la selección de la metodología y tecnologías de desarrollo y el diseño de un sistema informático, así como la solución de problemas que requieren del enriquecimiento autónomo.
- El 100 % de los profesores expresa que aún existen limitaciones en la autoformación de los estudiantes en el proceso de enseñanza aprendizaje de la Ingeniería de Software.

Se realiza entonces la observación al proceso de enseñanza aprendizaje de la Ingeniería de Software en la carrera de Ingeniería Informática de la Universidad de Ciego de Ávila. Se obtienen como resultados:

- Bajo nivel de independencia para la realización de los flujos de trabajos propios de las fases ingenieriles de desarrollo de software, teniendo en cuenta que más del 50 % de los estudiantes requiere de ayuda del profesor o de otros estudiantes.
- Irregularidades en la selección de las tecnologías de desarrollo de software en más del 50 % de los estudiantes.
- En el diseño e implementación del sistema informático, el 100 % de los estudiantes requiere de la orientación de los métodos y herramientas a emplear.
- Más del 50 % de los estudiantes muestra imprecisiones en la selección de una metodología en el desarrollo del proyecto asignado.
- Carencias y falta de rigor en el estudio y análisis del contexto para desarrollar un sistema informático.
- Prevalencia de métodos expositivos, pero con limitaciones en la problematización del proceso formativo para potenciar la autoformación.
- No siempre se desarrollan acciones para la sistematización formativa desde de la auto-concientización y auto-apropiación del contenido por el estudiante para su desarrollo cognitivo de manera independiente.

En general, los resultados de las encuestas, la entrevista, así como la observación, evidencian insuficiencias en el proceso de enseñanza aprendizaje de la Ingeniería de Software para contribuir al enriquecimiento autónomo del conocimiento del estudiante, que le permita la solución de tareas de manera independiente

En este contexto se hace necesario establecer un modelo de la dinámica del proceso de autoformación del estudiante en el proceso de enseñanza aprendizaje de la Ingeniería de Software en la carrera Ingeniería Informática que contribuya a la autonomía del estudiante en la auto-apropiación del conocimiento, una vez que concientiza lo que aprende en la sistematización formativa.

MODELO DE LA DINÁMICA DEL PROCESO DE AUTOFORMACIÓN EN LA ENSEÑANZA APRENDIZAJE DE LA INGENIERÍA DE SOFTWARE EN LA CARRERA INGENIERÍA INFORMÁTICA

Teniendo en cuenta las ideas anteriores y desde la teoría de Fuentes (2009) que considera el modelo como una representación simplificada del objeto o proceso que se analiza y que refleja solo algunas características esenciales en el fenómeno estudiado y las consideraciones de Tejeda (2015) al establecer que un modelo puede asumir un sentido didáctico, educativo o formativo en su forma de concebir desde la teoría, la dinámica de la práctica expresada desde una estructura que tiene niveles jerárquicos y relaciones internas entre; se realiza el modelo de la dinámica del proceso de autoformación del estudiante la enseñanza aprendizaje de la Ingeniería de Software en la carrera Ingeniería Informática (Figura 1).

En la modelación de esta dinámica, se reparó en estas especificidades, desde la relación establecida entre la auto-apropiación de contenidos y la auto-concientización en la sistematización formativa, lo que posibilita su concepción desde tres eslabones, que no se suceden de forma lineal, sino que se interrelacionan en la práctica del proceso de enseñanza aprendizaje de la asignatura Ingeniería de Software para lograr en el estudiante su autoformación.

1. Eslabón de construcción interactiva del contexto a informatizar desde la construcción empírica de las bases para el desarrollo de software.
2. Eslabón de análisis y diseño de métodos y estrategias para el desarrollo de software desde la concientización transformadora.
3. Eslabón de implementación y pruebas desde la comprensión crítica del desarrollo de software.

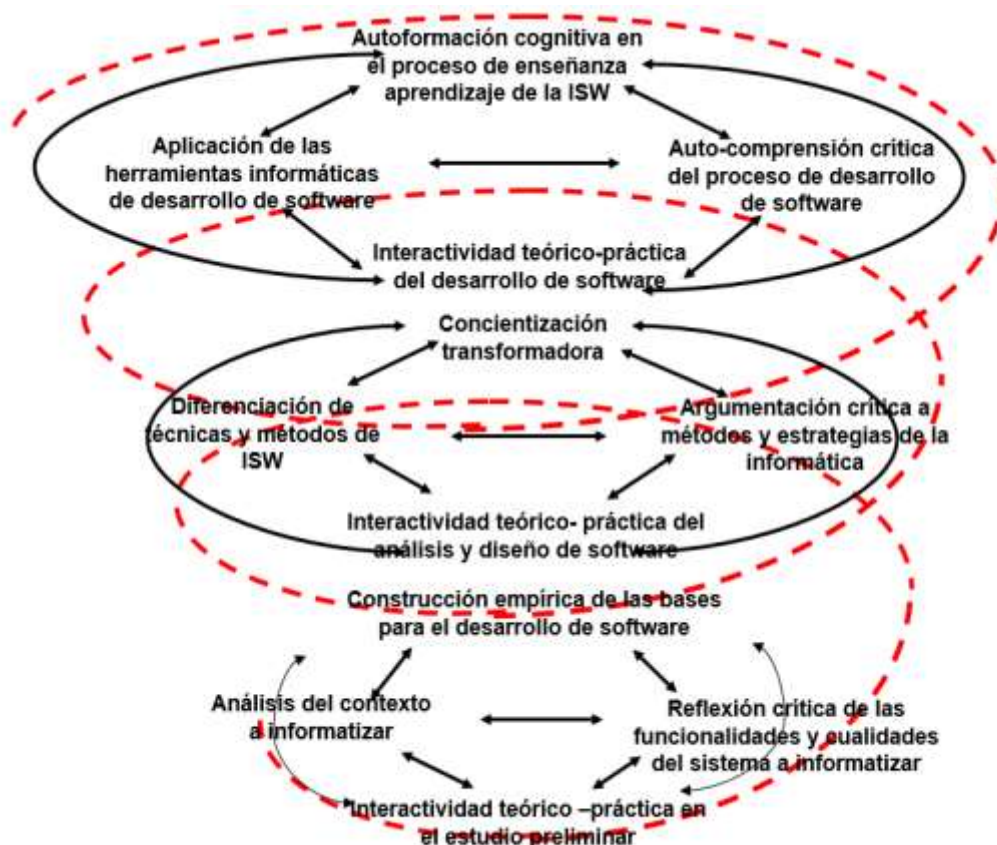


Figura 1. Modelo de la dinámica del proceso de autoformación del estudiante en el proceso de enseñanza aprendizaje de la Ingeniería de Software

ESLABÓN DE CONSTRUCCIÓN INTERACTIVA DEL CONTEXTO A INFORMATIZAR DESDE LA CONSTRUCCIÓN EMPÍRICA DE LAS BASES PARA EL DESARROLLO DE SOFTWARE

En este eslabón se establece la relación entre el análisis del contexto a informatizar - la reflexión crítica de las funcionalidades y cualidades del sistema informático - interactividad teórico práctica en el estudio preliminar - construcción empírica de las bases para el desarrollo de software que define la dimensión analítico-reflexiva de la dinámica del proceso de autoformación en la enseñanza aprendizaje de la Ingeniería de Software mediante métodos novedosos; que expresa la necesidad de establecer bases científico-cognitivas, donde el estudiante reescribe su propia acción en el estudio del contexto del sistema a informatizar, como vía de reflexión y análisis en la construcción de las bases para el desarrollo de software.

El análisis del contexto a informatizar es el proceso mediante el cual el estudiante identifica los objetos que se relacionan en el dominio del proceso que va a ser informatizado, desde la observación de los rasgos, particularidades y características que lo tipifican, que conducen a una primera aproximación de las funcionales y cualidades del sistema a desarrollar. Este proceso requiere de un análisis profundo de las características del sistema que se desarrollará en el contexto estudiado como vía de profundización del conocimiento.

Desde este punto de vista, el análisis de las características del contexto a informatizar conlleva a una reflexión crítica de las funcionalidades y cualidades del sistema a desarrollar, como proceso que posibilita al estudiante comprender los factores determinantes para tomar decisiones sobre el camino a seguir en la construcción del conocimiento práctico hacia el desarrollo de software. Esta reflexión a su vez, conduce al análisis del contexto con el objetivo de definir las funcionalidades que tendrá el sistema informático y bajo qué condiciones se desarrollará. En este proceso el estudiante reflexiona sobre lo analizado, reescribe su propia acción y toma conciencia de lo que se va a construir en el camino hacia su desarrollo autónomo.

En la dinámica del proceso de autoformación en la enseñanza aprendizaje de la Ingeniería de Software, estos procesos se contraponen y presuponen en la interactividad teórico práctica del estudio preliminar, como espacio donde el estudiante combina los métodos de aprendizaje en la práctica con la aplicación del conocimiento, habilidades y valores que logra en la sistematización formativa, con la finalidad de lograr la construcción empírica de las bases para el desarrollo de software.

ESLABÓN DE ANÁLISIS Y DISEÑO DE MÉTODOS Y ESTRATEGIAS PARA EL DESARROLLO DE SOFTWARE DESDE LA CONCIENTIZACIÓN TRANSFORMADORA

En este eslabón se establece la relación entre la diferenciación de técnicas y métodos de ingeniería de software - argumentación crítica de métodos y estrategias de la informática - interactividad teórico práctica del análisis y diseño de software – concientización transformadora, la cual define la dimensión metodológica argumentativa del proceso modelado, que expresa la necesidad de una concientización transformadora en el estudiante sobre el comportamiento que tendrá el sistema informático y cómo será implementado en términos de componentes, objetos y sus interacciones; así como las herramientas a utilizar; para obtener un software con la calidad requerida.

La diferenciación de técnicas y métodos de ingeniería de software es el proceso mediante el cual se realiza un análisis del sistema a desarrollar desde una abstracción que establece funciones y comportamiento de los objetos que intervienen para su posterior diseño. El sistema a implementar se descompone y organiza en término de objetos y sus relaciones con el objetivo de establecer una estructura interna del software. En este proceso se tiene en cuenta la independencia cognitiva que logra el estudiante durante al construir y aplicar el conocimiento en la selección de técnicas y métodos.

La diferenciación de técnicas y métodos de Ingeniería de Software requiere entonces de una argumentación crítica a métodos y estrategias de la informática donde el estudiante asume una posición clara, precisa y profunda para expresar su opinión y los argumentos razonables que la sustentan y lo lleven a conclusiones sobre la solución encontrada.

A su vez, esta argumentación lleva a la diferenciación de métodos y técnicas que se manifiestan en la selección acertada. Al lograr una selección y argumentación sólida el estudiante en su proceso de apropiación del contenido logra niveles de concientización de lo ejecutado.

En este segundo eslabón se desarrolla el proceso de enseñanza aprendizaje desde la interactividad teórico práctica del análisis y diseño de software, que requiere la construcción del conocimiento sobre los objetos, sus relaciones y comportamiento; bajo la orientación del profesor con una intencionalidad pedagógica, aplicando métodos activos. Este proceso, se construye y desarrolla en torno al análisis y diseño de software; por tanto, tiene una intención instruccional; donde se establecen características y secuenciación de habilidades y del contenido precedente.

Estos procesos, desde su relación como unidad dialéctica en el proceso de enseñanza aprendizaje de la Ingeniería de Software, conllevan a logros superiores que apuntan hacia la concientización transformadora del estudiante, entendida como el proceso mediante el cual este avanza y toma una conciencia profunda de lo aprendido al involucrarse, transformar el conocimiento y actuar en correspondencia durante el análisis y diseño del sistema informático.

Por tanto, en el proceso de autoformación del estudiante la argumentación crítica de métodos y estrategias de la informática en relación bidireccional con la diferenciación de técnicas y métodos de ingeniería de software; en la interacción teórico práctica del análisis y diseño de software se enriquece con métodos novedosos en la sistematización formativa para dinamizar la apropiación y profundización del conocimiento. Este proceso enriquece la construcción de las bases para el desarrollo de software y forma en el estudiante una conciencia transformadora para promover acciones que conducen al desarrollo de software elevando los niveles de independencia.

ESLABÓN DE IMPLEMENTACIÓN Y PRUEBAS DESDE LA COMPRESIÓN CRÍTICA DEL DESARROLLO DE SOFTWARE

En este eslabón se establece la relación entre la aplicación de las herramientas informáticas de desarrollo de software - autocomprensión crítica del proceso de desarrollo de software - interactividad práctica del desarrollo de software - autoformación cognitiva en el proceso de enseñanza aprendizaje de la Ingeniería de Software, que define la dimensión práctica autoformativa del proceso modelado, expresada en el constante auto-desarrollo del estudiante, que implica la existencia de técnicas novedosas para su autonomía.

En el proceso hacia la autoformación, el estudiante no solo depende de los conocimientos adquiridos, se requiere de la auto-concientización de lo que aprende una vez que es capaz de actuar por sí solo en la solución de tareas ingenieriles. En este contexto se potencia la relación entre los métodos utilizados y el desarrollo independiente del estudiante que lo conduzca a lograr su autoformación en el proceso de enseñanza aprendizaje de la Ingeniería de Software.

La aplicación de las herramientas informáticas de desarrollo de software se produce en la dinámica del proceso de enseñanza aprendizaje cuando el estudiante utiliza programas o aplicaciones que le permiten implementar y probar el sistema informático. En este proceso se traducen los modelos del diseño realizado en etapas anteriores en código, utilizando un lenguaje de programación adecuado para el sistema a desarrollar.

Una vez programado el sistema informático se evalúa y verifica que el sistema responda a las funcionalidades deseadas de las clases, componentes o módulos y mantenga el comportamiento esperado previendo errores; revelándose la pertinencia del estudio realizado para seleccionar los métodos y estrategias más adecuados para el desarrollo de software.

Por tanto, este proceso conduce a la auto-comprensión crítica del proceso de desarrollo de software en el cual se realiza una retrospectiva de las actividades y acciones realizadas para verificar y

comprobar su estado. Cuando el estudiante es capaz de responder qué hizo, para qué lo hizo y cómo lo hizo el aprendizaje se considera significativo, relacionando en cada etapa los nuevos conocimientos con los saberes previos y a la vez profundizando en el conocimiento. De igual forma, una adecuada auto-comprensión crítica requiere de la aplicación de las herramientas informáticas donde el estudiante demuestra los conocimientos y habilidades unido a las valoraciones que realiza; en ese punto alcanza su mayor grado de auto-concientización.

En este proceso de interactividad práctica del desarrollo de software se produce la apropiación de contenidos a la vez que el estudiante concientiza lo que aprendió en la dinámica del proceso de enseñanza aprendizaje, organizando y asimilando los nuevos conocimientos con la lógica y sentido que le encuentra en su aplicación práctica y con las habilidades requeridas en el desarrollo de software que lo conducen a la autoformación cognitiva en el proceso de enseñanza aprendizaje de la Ingeniería de Software.

Por tanto, la autoformación cognitiva en este proceso se logra cuando el estudiante en la constante renovación de los métodos y estrategias utilizados; valora, establece juicios propios y resalta el significado de lo aprendido. En la apropiación de contenidos, el estudiante desarrolla un aprendizaje más duradero y profundo al valorar y tomar conciencia de sus propias estrategias, destrezas y habilidades.

Se produce entonces, en el proceso de tránsito desde la construcción de las bases para el desarrollo de software hacia la concientización transformadora que va logrando el estudiante de lo aprendido en la interactividad del proceso, una autoformación cognitiva y el estudiante logra aprender a aprender.

Por todo lo anterior, se considera importante la autoformación del estudiante desde el propio proceso de enseñanza aprendizaje de la Ingeniería de Software, incentivándose la apropiación de contenidos en el desarrollo de software y la auto-concientización que deriva hacia la auto-comprensión del proceso.

Teniendo en cuenta el análisis realizado, de cómo transcurre la dinámica del proceso de autoformación del estudiante para el desarrollo de software y las cualidades esenciales reveladas, se definen tres relaciones fundamentales que se ponen de manifiesto en el proceso de enseñanza aprendizaje de la Ingeniería de Software:

- Relación analítica-reflexiva para la construcción de las bases del desarrollo del software.
- Relación de sistematización analítica-argumentativa de la selección de técnicas métodos de ingeniería de software.
- Relación de autoformación cognitiva en el proceso de enseñanza aprendizaje de la Ingeniería de Software.

CONCLUSIONES

La modelación de la dinámica del proceso de autoformación del estudiante en la enseñanza aprendizaje de la Ingeniería de Software en la carrera Ingeniería Informática parte de la sistematización teórica general y el diagnóstico del estado actual de este proceso formativo, lo cual posibilita develar la necesidad de su elaboración.

La aplicación del método holístico-dialéctico posibilita identificar configuraciones, como expresiones del proceso modelado como un todo, establecer relaciones entre estas, definir dimensiones y eslabones o estadios en este proceso formativo, desde la lógica integradora que se establece entre la auto-apropiación de contenidos y la auto-concientización en la sistematización formativa.

El modelo aporta relaciones fundamentales y la regularidad esencial, que se concreta en el carácter analítico-reflexivo de la dinámica de autoformación y auto-desarrollo para la concientización transformadora en la interactividad teórico práctica en el proceso de enseñanza aprendizaje de la Ingeniería de Software, desde la lógica integradora que se produce entre la auto-apropiación de contenidos y la auto-concientización en la sistematización formativa, sustento teórico que, desde

su concreción en instrumentos prácticos, puede contribuir a atenuar las limitaciones que presentan los estudiantes en la solución de tareas que requieren del enriquecimiento autónomo del conocimiento en la Ingeniería de Software.

DECLARACIÓN DE CONFLICTOS DE INTERESES: Los autores declaran no tener conflictos de interés.

DECLARACIÓN DE CONTRIBUCIÓN DE LOS AUTORES Y AGRADECIMIENTOS: A continuación, se menciona la contribución de cada autor, en correspondencia con su participación, utilizando la Taxonomía Crédit:

- Ivet Espinosa Conde: Autor principal, Conceptualización, Análisis formal, Investigación, Metodología, Redacción, Redacción - revisión y edición.
- Raquel Diéguez Batista: Conceptualización, Análisis formal, Metodología, Redacción-revisión y edición.
- Yulkeidys Martínez Espinosa: Investigación, Análisis formal, Metodología, Redacción-revisión

DECLARACIÓN DE APROBACIÓN DEL COMITÉ DE ÉTICA: Los autores declaran que la investigación fue aprobada por el Comité de Ética de la institución responsable, en tanto la misma implicó a seres humanos.

REFERENCIAS BIBLIOGRÁFICAS

- Alarcón, R. A., Guzmán, Y., & García, M. (2019). Formación integral en la Educación Superior: una visión cubana. *Revista Estudios del Desarrollo Social: Cuba y América Latina*, 7(3), 73-82. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2308-01322019000300010
- Anaya, R. (2012). Una visión de la enseñanza de la Ingeniería de Software como apoyo al mejoramiento de las empresas de software. *Universidad Eafit*, 42(141), 60-76. <https://www.redalyc.org/pdf/215/21514105.pdf>
- Barajas, J. I., & Ochoa, E. (2021). Autoformación online para la búsqueda y recuperación de información en una biblioteca virtual. *Campus virtuales*, 10(2), 27-49. <https://dialnet.unirioja.es/servlet/articulo?codigo=8053693>
- Betanco, J. A. (2020). Una buena práctica de empresa, es clave para el desarrollo profesional de estudiantes de Energía Renovable. *Revista Multi-Ensayos*, 6(11), 67-76. <https://doi.org/10.5377/multiensayos.v6i11.9291>
- Ciudad, F., & Soto, N. (2006). *La enseñanza de la Ingeniería de Software (ISW) en la Universidad de las Ciencias Informáticas (UCI) utilizando emisiones televisivas didácticas*. UCIENCIA. II Conferencia Científica de la Universidad de Ciencias Informáticas, La Habana, Cuba.
- Delgado, L. C., & Díaz, L. M. (2021). Modelos de desarrollo de software. *Revista Cubana de Ciencias Informáticas*, 15(1), 3-14. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=s2227-18992021000100037#aff2
- Enríquez, R. I. (2021). La Efectividad del aprendizaje activo en la práctica docente. *EduSol*, 21(74), 102-110. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1729-80912021000100102
- Facultad de Ingeniería Informática. (2019). *Programa de la Disciplina Ingeniería y Gestión de Software*. UNICA.
- Fuentes, H. (2009). *Pedagogía y didáctica de la Educación Superior*. Universidad de Oriente.
- Guamán, V. J., & Espinoza, E. E. (2022). Aprendizaje basado en problemas para el proceso de enseñanza aprendizaje. *Revista Universidad y Sociedad*, 14(2), 124-133. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2218-36202022000200124
- Gómez, R., & Moraleda, A. (2020). *Aproximación a la Ingeniería de Software*. Editorial Universitaria Ramón Areces.

- Guitart, I., Rodríguez, M. E., Cabot, J., & Serra, M. (2006). *Elección del modelo de evaluación: caso práctico para asignaturas de ingeniería del software*. Universitat, XII Jornadas Enseñanza Universitaria de la Informática, España. http://bioinfo.uib.es/~joemi/procJenui/Jen2006/prDef0035_96a3be3cf2.pdf
- Hernández, M. L., Olgún, P. A., & Hernández, Y. (2019). Transformando la enseñanza de Ingeniería de Software a nivel posgrado. En *Tendencias y desafíos en la innovación educativa: un debate abierto* (pp. 1159-1167). Laboratorio Nacional de Informática Avanzada. <https://www.repo-ciie.dfie.ipn.mx/pdf/480.pdf>
- Hernández, R., Fernández, C., & Baptista, P. (2010). *Metodología de la investigación* (5° Ed.). McGraw Hill Interamericana.
- Incencio, G. S., Guerra, L. M., & Lissabet, J. L. (2022). Caracterización epistemológica del proceso de enseñanza-aprendizaje de la asignatura Ingeniería de Software I. *Edusol*, 22(80), 1-14. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1729-80912022000300176
- Jiménez, D., González, J. J., & Tornel, M. (2020). Metodologías activas en la Universidad y su relación con los enfoques de enseñanza. *Profesorado. Revista de currículum y formación del profesorado*, 24(1), 76-94. <https://doi.org/10.30827/profesorado.v24i1.8173>
- Labra, E., Fernández, D., Calvo, J., & Cernuda del Río, A. (2006). *Una experiencia de aprendizaje basado en proyectos utilizando herramientas colaborativas de desarrollo de software libre*. Universitat, XII Jornadas de Enseñanza Universitaria de la Informática, España. <http://rua.ua.es/dspace/handle/10045/127576?mode=full>
- Linares, M., Aleas, M., Mena, J. A., Cruz, D., & Rosales, D. (2019). Comportamiento histórico de la enseñanza del diseño de software para la carrera Sistemas de Información en Salud. *Revista Cubana de Informática Médica*, 11(2), 1684-1859. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1684-18592019000200158
- Mariño, S. I., López, M. V., & Golobisky, M. F. (2001). *Un software interactivo orientado a la enseñanza del Método de Programación por Camino Crítico*. CACIC, VII Congreso Argentino de Ciencias de la Computación, Argentina. <http://sedici.unlp.edu.ar/handle/10915/23209>
- Menéndez, S. (16 de junio de 2020). El sentido de aprender. *Enfoque Educación*. <https://blogs.iadb.org/educacion/es/elsentidodeaprender/>
- Ministerio de Educación República de Cuba [MINED]. (15 al 18 de mayo de 2001). *Contenido de la educación y estrategias de aprendizaje para el siglo XXI*. Seminario Subregional sobre el Desarrollo Curricular para Aprender a Vivir Juntos. MINED. http://www.ibe.unesco.org/fileadmin/user_upload/archive/curriculum/Caribbean/CaribbeanPdf/cuba.pdf
- Ministerio de Educación Superior [MES]. (2017). *Plan de estudio E carrera Ingeniería Informática*. MES.
- Ministerio de Educación Superior [MES]. (2022). *Reglamento organizativo del proceso docente y de dirección del trabajo docente y metodológico para las carreras universitarias*. MES.
- Monroy, M. E., Chanchí, G. E., & Ospina, M. A. (2022). Desarrollo de habilidades técnicas en ingeniería de software aplicando ingeniería inversa. *Revista Boletín Redipe*, 11(1), 34-50. <https://revista.redipe.org/index.php/1/article/view/1661>
- Montoya-Suárez, L. M., & Pulgarín-Mejía, E. (2013). Enseñanza en la Ingeniería de Software: aproximación a un estado del arte. *Lámpakos*, 9(10), 76-91. <https://revistas.ucatolicaluisamigo.edu.co/index.php/lampakos/article/download/1338/1216>
- Morales, Y., Aguilar, V., & Rodríguez, C. (2018). Los medios de enseñanza para la apropiación de contenidos profesionales. *Mendive. Revista de Educación*, 16(1), 65-78. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1815-76962018000100065
- Naya, I., & Oliva, K. (2015). *Sistema evaluador de habilidades y de recomendación para el apoyo al proceso de enseñanza-aprendizaje de la Ingeniería de Software* [Tesis de pregrado,

- Universidad de las Ciencias Informáticas]. Repositorio de la Universidad de las Ciencias Informáticas. <https://repositorio.uci.cu/jspui/handle/123456789/7055>
- Ramos, J. M., & Vidal, R. (2016). ¿Cómo realizar la sistematización de la práctica educativa? *Docencia e Investigación*, 41(26), 53-76. <https://revista.uclm.es/index.php/rdi/issue/view/26>
- Taran, G. (2007). *Using games in software engineering education to teach risk management*. Cseet'07. 20th Conference Software Engineering Education & Training. <https://ieeexplore.ieee.org/document/4271608>
- Tejeda, R. (2015). El aporte teórico en investigaciones asociadas a las ciencias pedagógicas. *Didáctica y Educación*, 6(6), 103-120. <https://revistas.ult.edu.cu/index.php/didascalía/article/view/438>
- Trujillo, T. (2019). Reflexiones sobre el proceso enseñanza- aprendizaje. *Correo Científico Médico*, 23(4), 2187-2191. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1560-43812019000401460
- Veloso, A., & Veloso, E. (2022). La educación de la personalidad del Licenciado en Psicología: cómo estimularla desde la formación profesional. *Revista Integración Académica en Psicología*, 4(10), 85-96. <https://www.alfepsi.org/revista-integracion-academica-en-psicologia-vol-4-n10/>
- Vigotsky, L. (1987). *Teoría del desarrollo cultural de las funciones psíquicas*. Pueblo y Educación.
- Zapata, C. M., & Duarte, M. I. (2008). El juego de la consistencia: una estrategia didáctica para la Ingeniería de Software. *Revista Técnica Ingeniería de la Universidad del Zulia*, 31(1), 1–10. http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S0254-07702008000100002
- Zapata, C. M., González, G., & Manjarrés, R. A. (2012). *Un juego para la enseñanza de métodos de desarrollo de software* [Tesis de pregrado, Universidad de Medellín]. Repositorio de la Universidad de Medellín. <https://repository.udem.edu.co/handle/11407/2323?locale-attribute=en>
- Zapata, C. M., Villegas, S. M., & Arango, F. (2012). Reglas de consistencia entre modelos de requisitos de un método. *Universidad Eafit*, 42(141), 40-59. <https://www.redalyc.org/pdf/215/21514104.pdf>

Este preprint fue presentado bajo las siguientes condiciones:

- Los autores declaran que son conscientes de que son los únicos responsables del contenido del preprint y que el depósito en SciELO Preprints no significa ningún compromiso por parte de SciELO, excepto su preservación y difusión.
- Los autores declaran que se obtuvieron los términos necesarios del consentimiento libre e informado de los participantes o pacientes en la investigación y se describen en el manuscrito, cuando corresponde.
- Los autores declaran que la preparación del manuscrito siguió las normas éticas de comunicación científica.
- Los autores declaran que los datos, las aplicaciones y otros contenidos subyacentes al manuscrito están referenciados.
- El manuscrito depositado está en formato PDF.
- Los autores declaran que la investigación que dio origen al manuscrito siguió buenas prácticas éticas y que las aprobaciones necesarias de los comités de ética de investigación, cuando corresponda, se describen en el manuscrito.
- Los autores declaran que una vez que un manuscrito es postado en el servidor SciELO Preprints, sólo puede ser retirado mediante solicitud a la Secretaría Editorial deSciELO Preprints, que publicará un aviso de retracción en su lugar.
- Los autores aceptan que el manuscrito aprobado esté disponible bajo licencia [Creative Commons CC-BY](#).
- El autor que presenta el manuscrito declara que las contribuciones de todos los autores y la declaración de conflicto de intereses se incluyen explícitamente y en secciones específicas del manuscrito.
- Los autores declaran que el manuscrito no fue depositado y/o previamente puesto a disposición en otro servidor de preprints o publicado en una revista.
- Si el manuscrito está siendo evaluado o siendo preparando para su publicación pero aún no ha sido publicado por una revista, los autores declaran que han recibido autorización de la revista para hacer este depósito.
- El autor que envía el manuscrito declara que todos los autores del mismo están de acuerdo con el envío a SciELO Preprints.