

# Tendencias históricas de la autoformación en el proceso de enseñanza aprendizaje de la ingeniería de software en la carrera ingeniería informática

## historical trends of self-training in the teaching-learning process of software engineering in the computer engineering career

Ivet Espinosa-Conde

✉ [ivetc@unica.cu](mailto:ivetc@unica.cu)

 <https://orcid.org/0000-0001-5102-2339>

Yulkeidi Martínez-Espinosa

✉ [ymtnez@unica.cu](mailto:ymtnez@unica.cu),

 <https://orcid.org/0000-0003-2221-0650>

Raquel Diéguez-Batista

✉ [raquel@unica.cu](mailto:raquel@unica.cu)

 <https://orcid.org/0000-0002-4975-6947>

Universidad de Ciego de Ávila Máximo Gómez Báez, Cuba.

### Resumen

El estudio realizado tuvo como objetivo determinar las tendencias históricas de la autoformación en el proceso de enseñanza aprendizaje de la asignatura Ingeniería de Software en la carrera Ingeniería Informática a partir de las limitaciones que presentan los estudiantes de la carrera para la solución de tareas en la asignatura Ingeniería de Software que requieren del enriquecimiento autónomo del conocimiento. Para realizar el análisis se determinaron tres etapas según indicadores definidos, mediante el empleo de métodos del nivel teórico y empírico como analítico-sintético, abstracción-concreción, inducción-deducción; lo que permitió revelar las características y transformaciones más importantes en cada período. Además, se evidenció la necesidad de utilización de métodos activos y mediadores didácticos digitales para potenciar la autoformación.

**Palabras clave:** autoformación, Ingeniería de Software, proceso de enseñanza-aprendizaje

## Abstract

The study carried out aimed to determine the historical trends of self-training in the teaching-learning process of the Software Engineering subject in the Computer Engineering career based on the limitations that the students of the career present for the solution of tasks in the subject. Software Engineering that require the autonomous enrichment of knowledge. To carry out the analysis, three stages were determined according to defined indicators, through the use of theoretical and empirical methods such as analytical-synthetic, abstraction-concretion, induction-deduction; which allowed revealing the most important characteristics and transformations in each period, which evidenced the need to use active methods and digital didactic mediators to enhance self-training.

**Keywords:** self training, Software Engineering, teaching-learning process

## Introducción

La Ingeniería de Software es conocida como la columna vertebral del ingeniero informático, por la importancia que reviste en la formación del futuro profesional. Su encargo es la aplicación práctica del conocimiento en el análisis, diseño, implementación, pruebas y despliegue de sistemas informáticos, mediante un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software (Gómez & Moraleda, 2020). Para lograr sus objetivos hace uso de metodologías y modelos que guían el proceso para lograr sistemas más eficientes y de mayor calidad, con la documentación necesaria en perfecto orden y en el tiempo requerido, apoyándose en herramientas informáticas (Olivera y Alonso, 2021).

El proceso de enseñanza-aprendizaje en la asignatura Ingeniería de Software en Cuba, en su devenir histórico ha sufrido transformaciones en los diferentes planes de estudio por los que ha transitado la carrera de Ingeniería Informática. Al realizar un análisis de los hechos que marcaron la evolución del proceso se delimitaron hitos como la flexibilidad curricular con mayor énfasis en las horas de estudio independiente y la utilización con mayor medida de los mediadores didácticos digitales como apoyo al proceso. Lo que posibilitó dividir el estudio en etapas y determinar sus características. El estudio parte de considerar las transformaciones más sobresalientes desde que se menciona por primera vez el termino Ingeniería de Software en el año 1968.

Los primeros planes de estudio estuvieron enfocados a la formación de un profesional que enfrentara la creciente incorporación de las Tecnologías de la Información y las Comunicaciones a diversos sectores del país; sin embargo, con el decursar de los años estos fueron perfeccionándose respondiendo a los objetivos y exigencias de la sociedad actual. En el Reglamento organizativo del proceso docente y de dirección del trabajo docente y metodológico para las carreras universitarias en la Educación Superior en Cuba, se 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 (Ministerio de Educación Superior, 2022).

En específico la carrera Ingeniería Informática tiene entre sus objetivos formar ingenieros independientes, responsables y creativos capaces de dar solución a los problemas que enfrentará, considerando el amplio espectro de equipos multidisciplinarios que integrará 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.

En correspondencia con las exigencias planteadas se diseñan los programas de disciplina y asignatura en los diferentes Centros de Educación Superior. En particular, en 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) se 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

En general, en estos objetivos se establece la necesidad de que el estudiante sea capaz de lograr autonomía en su formación académica, que le permita enfrentarse a un entorno donde la tecnología evoluciona constantemente; situación que presenta limitaciones en la Universidad de Ciego de Ávila. 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 sistematización formativa, lo que requiere del reconocimiento por parte del docente que su rol ha cambiado, no se trata de enseñar contenidos, sino de enseñar a aprender.

Teniendo en cuenta los elementos anteriores, se hace necesario profundizar en el estudio de las tendencias históricas de la autoformación en el proceso de enseñanza aprendizaje de la Ingeniería de Software, objetivo del presente trabajo que demuestren la existencia de insuficiencias y revelen la necesidad de contribuir a su perfeccionamiento.

El empleo del método histórico-lógico, análisis-síntesis e inducción-deducción permitieron profundizar en las transformaciones ocurridas en el proceso de enseñanza aprendizaje de la Ingeniería de Software en cada una de las etapas definidas. El uso de los métodos mencionados permitió comprender, explicar e interpretar la necesidad de perfeccionar el proceso utilizando mediadores didácticos digitales como alternativa didáctica fundamental para potenciar la autoformación en el proceso de enseñanza aprendizaje de la Ingeniería de Software; teniendo en cuenta la necesidad de enriquecimiento autónomo del conocimiento como consecuencia del vertiginoso desarrollo de la ciencia y la técnica.

## Desarrollo

El estudio de las tendencias históricas del proceso de enseñanza aprendizaje de la asignatura Ingeniería de Software de la carrera Ingeniería Informática parte de la definición de los indicadores:

- Métodos que se utilizan en el proceso de enseñanza aprendizaje para potenciar la autoformación.

- Grado de integración de los mediadores didácticos digitales al proceso de enseñanza aprendizaje de la Ingeniería de Software para la autoformación.

A partir de los indicadores establecidos se determinan las etapas desde que se menciona por primera vez el término Ingeniería de Software e inicia el proceso de enseñanza aprendizaje de la asignatura en la carrera Ingeniería Informática perfeccionándose continuamente hasta la actualidad que se encuentra vigente el Plan de estudio E. El estudio realizado se basa en los acontecimientos relevantes que determinaron el comportamiento de los indicadores en el proceso. Teniendo en cuenta lo mencionado, se identifican en este proceso tres etapas:

- Primera etapa: Utilización de mediadores didácticos tradicionales en el proceso de enseñanza aprendizaje de la Ingeniería de Software. (1968-1991).
- Segunda etapa: Introducción de mediadores didácticos digitales como nuevo escenario en el proceso de enseñanza aprendizaje de la Ingeniería de Software (1991-2004).
- Tercera etapa: El uso de mediadores didácticos digitales para el desarrollo independiente del estudiante (2004-2015).
- Cuarta etapa: El uso de ambientes virtuales de aprendizaje para la optimización del proceso de enseñanza aprendizaje de la Ingeniería de Software (2015-Actualidad).

*Primera etapa: Utilización de mediadores didácticos tradicionales en el proceso de enseñanza aprendizaje de la Ingeniería de Software (1968-1991)*

La enseñanza aprendizaje de la Ingeniería de Software se remonta al año 1968 cuando Fritz Bauer enunció la primera definición de “Ingeniería de Software”, en el marco de la 1ra Conferencia de la OTAN para el debate relativo a la crisis del software. A partir de ese momento, su objeto de estudio lo constituyó el proceso de desarrollo de software. Las instituciones educativas que ofrecían formación en las especialidades relacionadas con la industria de software, comenzaron a incorporar los contenidos a sus planes de estudio; primero como temas aislados en diferentes disciplinas, luego como una disciplina única y finalmente, en varios países, como una carrera universitaria (Ciudad y Ruiz, 2012).

Esta primera etapa estuvo caracterizada por el aprendizaje autónomo de la Ingeniería de Software de forma empírica, lo que marcó un hito en el crecimiento de la industria (Linares

et al., 2019). En Cuba, tuvo sus antecedentes en el año 1976 cuando se crea la especialidad de Ingeniero en Sistemas Automatizados de Dirección Técnico Económico (SAD-TE). Al crearse esta carrera se regula el proceso de formación del profesional con el Plan de estudio A, diseñado de forma tal que el especialista se dedicara a la automatización de los procesos en empresas y dentro de esta se inclinaba hacia los procesos industriales. En este plan de estudio existían muy pocas asignaturas propias de la especialidad, la práctica era muy escasa y la bibliografía de la especialidad era prácticamente inexistente.

El Plan B significó un avance respecto al A, perfilándose mejor los ciclos de asignaturas, obteniéndose un mejor completamiento bibliográfico e incluyéndose prácticas en máquinas. Sin embargo el tiempo dedicado a la formación específica y de la especialidad era solo ligeramente mayor y el trabajo independiente del estudiante era aún bajo; lo que conllevó a su posterior actualización.

El Plan C comenzó a aplicarse en el curso 1990–1991 con mayor cantidad de horas dedicadas al trabajo independiente y al trabajo con computadoras personales; aunque aún era bajo en el sentido de lograr autonomía en el estudiante. Las indicaciones metodológicas ubican en esta etapa al proceso de enseñanza-aprendizaje en función del entorno sociocultural comunitario. Se orienta la forma de tratar el contenido, realizar las demostraciones y resolver los problemas propuestos en la bibliografía básica de cada año; pero no se ofrecen sugerencias concretas para efectuar el tratamiento de metodologías para el ciclo de vida de desarrollo de software. Además el estudiante soluciona problemas relacionados con el desarrollo de sistemas informáticos sin una suficiente comprensión de la labor que realiza, lo que limita el cumplimiento de los objetivos del año (Piñeiro, 2022).

En esta etapa se comienzan a utilizar un conjunto de estrategias didácticas para ingeniería en general. En la enseñanza incluyen las clases magistrales y pequeños proyectos prácticos para la enseñanza de aspectos básicos de la asignatura utilizando métodos expositivos. Se utilizan recursos para apoyar el proceso en formato impreso, a través de libros y documentos digitales que apoyan al profesor en el aula en su trabajo de transmitir los conocimientos.

Los indicadores de análisis en esta etapa revelan carencia de orientación didáctica hacia el uso de mediadores didácticos digitales. Los mediadores didácticos que se utilizan son libros

impresos, revistas, folletos para acompañar el proceso en el aula y contribuir a la apropiación de contenidos pero con limitaciones.

*Segunda etapa: Introducción de mediadores didácticos digitales como nuevo escenario en el proceso de enseñanza aprendizaje de la Ingeniería de Software (1991-2004)*

La década de los '90 significó un salto cualitativo en el uso y difusión las tecnologías informáticas gracias a la aparición del Internet y la *World Wide Web*. En este sentido, se aplican tecnologías y herramientas informáticas que sirvieron de apoyo a los procesos que se desarrollaban en las universidades. Herramientas como el correo electrónico, los mensajes digitales a través de teléfonos celulares y el uso en mayor medida de la computadora en los procesos; son solo algunos de los avances tecnológicos de esta etapa.

En Cuba, a tenor de este avance y luego de seis cursos de aplicación del Plan C, se obtuvieron resultados satisfactorios en la calidad de la preparación del graduado, pero a la vez se constató la necesidad de introducir nuevas modificaciones en dicho plan de estudio que lo adaptaran a los nuevos requerimientos. De esta forma y producto a los cambios tecnológicos, la apertura de la carrera en varias universidades del país incluyendo la Universidad de Ciego de Ávila, el surgimiento de la universidad de las Ciencias Informáticas (UCI) y la necesidad de mejorar algunas disciplinas, se realizó un profundo trabajo metodológico para ajustar el Plan de Estudio "C" perfeccionado de Ingeniería Informática, dando respuesta a los diferentes tipos de cursos.

La estrategia para ajustar del Plan de Estudio, aprobada en mayo del 2002 por la Comisión Nacional de Carrera de Ingeniería Informática (Ministerio de Educación Superior, 2018), tuvo en cuenta los elementos siguientes:

- Disminuir el componente académico en no más de un 10 % del total de horas del plan originalmente aprobado.
- Incrementar el componente laboral e investigativo en no más de un 10 % del total de horas del plan, para mejorar la formación de las habilidades profesionales.
- Incrementar el estudio independiente, asistido por las Tecnologías de la Información y las Comunicaciones (TIC).

- Incrementar la flexibilidad para cursar el plan de estudio.

En este sentido se dan los primeros pasos para lograr mayor desarrollo independiente del estudiante en correspondencia con la necesidad de que pueda enfrentarse a su entorno profesional y ayudar a construir un aprendizaje profundo; pero aun con limitaciones. Con el incremento del componente investigativo los estudiantes se preparan para utilizar los métodos, procedimientos, medios de investigación e instrumentos específicos; desarrollando su capacidad creadora y educándose en valores propios (Muñoz, 2020).

La disciplina Ingeniería y Gestión de Software aparece con ese nombre a partir de este plan de estudio, C perfeccionado (C'), aunque desde el primer plan se impartieron sus elementos básicos. En esta disciplina se introdujeron las asignaturas Ingeniería de Software I, II y III y Gestión de Software que abordaron el proceso de desarrollo profundizando en las etapas de análisis y diseño e incorporaron otras (Facultad de Ingeniería Informática, 2019).

En esta etapa se utiliza la computadora como mediador didáctico digital de apoyo al profesor y estudiante en el proceso de enseñanza aprendizaje. Se continúa utilizando el proyecto de curso como evaluación y se utilizan variantes didácticas con el apoyo de la tecnología para la enseñanza de la Ingeniería de Software como es el caso del aprendizaje por proyectos. Se realizan propuestas encaminadas a la elaboración de software mediante pasos o fases y se comienza a utilizar los juegos como parte del uso de métodos activos del estudiante en el aula.

Armarego (2009), considera que en esta etapa no existe una tendencia en la concepción del PEA de la Ingeniería de Software a sostenerse en principios didácticos y a relacionar de forma efectiva sus características con enfoques adecuados de integración con la industria de software. Gary (2009) considera que no se ofrece a los estudiantes entornos de trabajo en equipo que sean del contexto laboral-industrial con clientes, por lo que plantea que no se desarrollan las habilidades que necesitarán en su ejercicio profesional.

En la universidad de Ciego de Ávila, la didáctica del proceso se enfoca en la interacción entre profesores y estudiantes para consolidar conocimientos mediante proyectos de cursos sirviendo de soporte a la integración de las habilidades terminales del profesional.

Los indicadores de análisis en esta etapa revelan carencia en el uso de mediadores didácticos digitales como apoyo al proceso de enseñanza aprendizaje. Se utilizan variantes tecnológicas de apoyo al proceso, pero solo en el aula para la transmisión de conocimientos por parte del profesor y en el caso del estudiante para la consulta de bibliografía y la realización de proyectos.

*Tercera etapa: El uso de mediadores didácticos digitales para el desarrollo independiente del estudiante (2004-2015)*

El avance tecnológico unido a la necesidad de los países de desarrollar software para mejorar los procesos en sus empresas, llevaron a que dentro de las carreras orientadas al software como la informática, la computación o los sistemas existieran en los planes de estudio materias específicas dedicadas a la Ingeniería de Software, como Análisis de Sistemas, Diseño de Sistemas, Ingeniería de Software, Calidad del Software y otras (Linares et al., 2019). En consecuencia, en esta etapa se enseña la Ingeniería de Software en todas las Universidades donde existían carreras a fin para la formación de Ingenieros Informáticos (Universidad de las Ciencias Informáticas, Facultad de matemática de la Universidad de la Habana, Villa Clara, Ciego de Ávila entre otras). También se imparte en las Universidades de Ciencias Médicas para la formación en la carrera Licenciatura en Tecnología de la Salud, pero solo como asignatura básica que los prepara para desempeñarse como miembros de un equipo de desarrollo de software.

La diversidad de enfoques locales en cuanto a tecnologías, lenguajes de programación y áreas de aplicación, sumado a aspectos propios como el cambio en los niveles de abstracción al manejar recursos y herramientas informáticas; la creación del Ministerio de la Informática y las Comunicaciones (MIC) y la necesidad de lograr mayor independencia en el estudiante acercaron al Plan D en Cuba. A tenor con la evolución de la carrera y la necesidad de responder a proyectos de investigación y desarrollo en los que se involucraban estudiantes por la vía de ajustes a sus planes de estudio, fue natural en este contexto la introducción de los currículos propios y optativos/electivos que exigía el plan de estudios D. Prácticamente todos los Centros de Estudio Educativos (CES) en los que se impartía la carrera ya habían aplicado determinadas aproximaciones a currículos propios y optativos/electivos según

necesidades territoriales (Ministerio de Educación Superior, 2018).

En el proceso de enseñanza aprendizaje de la asignatura se comienzan a utilizar variantes didácticas para fomentar el trabajo independiente y la autonomía del estudiante. En este sentido se utilizan en mayor medida los juegos didácticos ya empleando con más frecuencia herramientas informáticas con el fin de ejemplificar acciones y etapas en el desarrollo de software así como fortalecer el trabajo en equipo y crear habilidades en el futuro profesional. Se estructuran elementos teórico-metodológicos con el fin de guiar la elaboración de software, y se realizan estudios con el objetivo de mejorar el proceso. Comienzan a utilizarse los métodos activos o metodologías activas, sustentadas en los postulados del aprendizaje cooperativo o colaborativo. Dentro de estos métodos se utiliza el aprendizaje basado en problemas (ABP), el aprendizaje basado en casos (ABC) y el aprendizaje orientado por proyectos (AOP); siendo considerados de forma indistinta como métodos, técnicas, estrategias o procedimientos.

En este sentido, comienzan los avances para que el profesor sea un facilitador del conocimiento y el estudiante construya su propio aprendizaje a través del uso de estos recursos didácticos, pero aún enfocado a la explicación de los contenidos teóricos por parte del profesor y a la realización de proyectos de clase por parte de los estudiantes. En la Universidad de Ciencias Informáticas, se comienza a utilizar la televisión como medio de transmisión masiva del conocimiento en la asignatura y la utilización de un sitio Web para exponer contenidos. Estas variantes aunque tuvieron definido un grupo de principios y buenas prácticas con el objetivo de no resultar monótonas a los estudiantes, se limitaban solo a la transmisión de contenidos.

En la Facultad de Informática de Ciego de Ávila, en esta etapa y a tenor con el avance de los Planes de Estudio se hizo necesario vincular todos los recursos disponibles para desde la didáctica de la asignatura graduar ingenieros de software altamente calificados. De esta forma, se vincula la teoría y la práctica con el uso de una determinada metodología, siendo utilizado el Proceso Unificado de Desarrollo de Software, acompañada del *Unified Modeling Language* (UML). Se mantiene el cumplimiento del principio de que el principal laboratorio que tienen los estudiantes es la práctica profesional para mantener un estrecho vínculo de trabajo con el entorno universitario, para esto se utilizan los proyectos de curso con la finalidad de resolver una situación problemática que requiera de la informatización como cierre de la asignatura.

En estos primeros cursos que se impartieron las asignaturas de Ingeniería de Software, no existía en la universidad, ningún entorno virtual de aprendizaje que lograra la interacción entre los estudiantes y el trabajo colaborativo.

El análisis de esta tercera etapa evidenció se utilizan mediadores didácticos digitales que apoyan el desarrollo independiente del estudiante pero la estructuración didáctica del contenido y habilidades en el desarrollo de software carecen de una concepción que conduzca al estudiante a elevar los niveles de auto-apropiación del contenido y auto-concientización en la sistematización formativa.

Las indicaciones metodológicas y de organización de la disciplina presentan limitaciones para que el estudiante comprenda, concientice e intérprete de la labor que realiza; limitando la fijación de conocimientos que le permitan enfrentar soluciones prácticas que requieran de aprendizaje autónomo.

*Cuarta etapa: El uso de ambientes virtuales de aprendizaje para la optimización del proceso de enseñanza aprendizaje de la Ingeniería de Software (2015-Actualidad)*

En la actualidad, ante la rápida evolución de las tecnologías informáticas se le ha dado una importancia significativa a la Ingeniería de Software. Países como Estonia imparte programación a los adolescentes de primer grado. En Estados Unidos se crearon escuelas secundarias para orientar a los estudiantes en carreras relacionadas con las Ciencias Computacionales o la Ingeniería de Software.

Existe un grupo grande de universidades americanas, que contemplan dentro del diseño curricular para la formación en Ingenierías, Técnicos Universitarios, Maestrías y Especialidades asociadas a la Computación, la Ingeniería de sistemas y de software. En el caso de Argentina (siete universidades), Chile (29 centros de formación en técnico de nivel superior, ingenierías y licenciaturas en análisis de sistema, diseño de sistemas y programación), Uruguay (tres universidades), Brasil (11 universidades), Venezuela (seis universidades) además cuenta con el Grupo de Investigación en Ingeniería del Software (INGESOF) realizando sus procesos en diversas áreas de la Informática, Ecuador (cinco universidades), Perú (ocho universidades), Bolivia (cinco universidades), Colombia (siete universidades). En Centroamérica podemos mencionar a Costa Rica (cuatro universidades), Guatemala (cuatro universidades), México

(cinco Universidades y 1 Instituto politécnico central) y Panamá (tres universidades) (Linares et al., 2019).

En Cuba se encuentra vigente el plan de estudio E, el cual se fundamenta en las transformaciones que han acontecido en el país. El plan hace énfasis marcado en la necesidad de lograr la formación del profesional hacia la solución de problemas más frecuentes del entorno social y productivo. El fortalecimiento de la formación de posgrado como vía para lograr habilidades y conocimientos de carácter específico del graduado en relación con las tareas que realiza y las tendencias internacionales en el desarrollo informático en su relación con otras tecnologías emergentes. Las tendencias a considerar una formación de pregrado con una mayor esencialidad y menor duración y las necesidades de formación de técnicos de nivel universitario. Además la flexibilidad del Plan E en cuanto a la cantidad de asignaturas optativas garantiza, sin dudas, mantener al estudiante actualizado con las tecnologías y herramientas de avanzadas que le permitan enfrentarse a la disímil gama de proyectos que desarrollan.

El conjunto de transformaciones sociales que se dan en esta etapa propiciadas por la innovación tecnológica y por el acelerado incremento y desarrollo de las tecnologías de la información y de la comunicación (TIC), los cambios en las relaciones sociales y una nueva concepción de las relaciones tecnología--sociedad que condiciona la posición de las tecnologías frente a la educación, ha contribuido a que se incrementen las alternativas, que desde un punto de vista didáctico, contribuyan al proceso de enseñanza aprendizaje de la Ingeniería de Software.

En función de esto, se han realizado investigaciones y propuestas que tienen como eje común el uso de las TICs para el desarrollo del proceso de enseñanza aprendizaje en la asignatura; fundamentalmente mediado por la virtualidad que contribuyan a lograr en los estudiantes mayor independencia cognitiva. En esta etapa, aparecen más investigaciones encaminadas al aprendizaje basado en proyectos y la utilización de software educativo que brindan posibilidades de realizar autoevaluaciones. Se realizan en mayor medida cursos en línea, multimedias y sitios web para apoyar el proceso de enseñanza aprendizaje.

En los últimos años, las instituciones educativas implementan plataformas de software como apoyo para estudiantes y profesores teniendo gran impacto en el proceso de enseñanza aprendizaje. Casi todas las instituciones de educación superior tienen algún tipo de entorno

de aprendizaje virtual o sistema de gestión del aprendizaje, ya sea un paquete comercial como Blackboard o Desire2Learn, o de código abierto como Moodle (Zurita et al., 2020).

En la Universidad de Ciego de Ávila se establece la utilización del Entorno Virtual de Aprendizaje (Moodle) para apoyar el proceso enseñanza aprendizaje de las diferentes disciplinas y asignaturas de las carreras. Los profesores de la carrera de Ingeniería Informática teniendo las habilidades necesarias preparan sus cursos en esta modalidad para apoyar el proceso de enseñanza aprendizaje. Este mediador didáctico digital facilita la interacción entre los actores del proceso, si se compara con los recursos utilizados en años anteriores, no obstante, se debe señalar que generalmente la participación de los estudiantes es poca y no se aprovecha la interactividad que ofrece ya que en ocasiones su uso está limitado a la subida y descarga de archivos.

En la UCI además de utilizar el Moodle, se establece un curso virtual que apoya el proceso de enseñanza aprendizaje de la Ingeniería y Gestión de Software. El curso mencionado, fue diseñado a partir de la utilización de un modelo de diseño instruccional. Fue ejecutado por etapas, culminando con la evaluación de la puesta en práctica del mismo. En el curso se obtiene el criterio de los usuarios a partir de la aplicación de cuatro niveles. Además, sobre la marcha, se realiza una evaluación formativa, con el objetivo de dar solución a los problemas que se presentan durante el desarrollo de la asignatura (Granda & Salinas, 2015).

El estudio de la etapa permitió revelar que el uso de ambientes virtuales de aprendizaje como medidores didácticos digitales en el proceso para el ciclo de vida de desarrollo de software carece aún de indicaciones, orientación y secuencias didácticas hacia el desarrollo de capacidades independientes del estudiante que contribuyan a su autoformación para enfrentar las actividades de su futuro profesional.

## Conclusiones

El análisis de los indicadores por etapas permitió revelar que el proceso de enseñanza aprendizaje de la Ingeniería de software ha cursado desde la utilización de métodos tradicionales dirigidos a la transmisión del conocimiento por parte del profesor, hasta la incentivación del papel protagónico del estudiante para su autoformación, pero aún con sesgos en la necesidad de su auto-concientización como elemento fundamental para su crecimiento profesional.

Asimismo, ha transitado desde la utilización de mediadores didácticos de primera y segunda generación hasta la potenciación de la utilización de las TIC como mediadores, que incluyen Ambientes Virtuales de Aprendizaje, pero aún carentes de una orientación metodológica de estos para potenciar como elemento fundamental en la apropiación de los contenidos, la auto-comprensión, desde la utilización de todas las funcionalidades de estos potenciando la interactividad.

## Referencias bibliográficas

- Armarego, J. (2009). Software Engineering: Effective teaching and learning approaches and practices. *Australasian Journal of Engineering Education* 19(1) <https://researchportal.murdoch.edu.au/esploro/>
- Ciudad Ricardo, F. A., y Ruiz Jhones, A. (2012). El proceso de enseñanza – aprendizaje de la disciplina ingeniería y gestión de software desde los proyectos industriales. *Pedagogía Universitaria*, 17(3) <https://www.researchgate.net/publication/317017857>
- Facultad de Ingeniería Informática. (2019). *Programa de la Disciplina Ingeniería y Gestión de Software*. Ciego de Ávila, Cuba.
- Gary Hunter, K. (2009). The Software Enterprise: preparing industry-ready software engineers. *Software Engineering: Effective teaching and learning approaches and practices*, 116.
- Gómez Palomo, S. R., y Moraleda Gil, E. (2020). *Aproximación a la Ingeniería de Software*. Prentice-hall Englewood cliffs: Editorial Universitaria Ramón Areces. [https://www.cerasa.es/libro/aproximacion-a-la-ingenieria-del-software\\_49949/](https://www.cerasa.es/libro/aproximacion-a-la-ingenieria-del-software_49949/)
- Granda Dihigo, A., y Salinas Ibáñez, J. (2015). *Diseño de curso virtual para apoyar el Proceso de Enseñanza Aprendizaje de la disciplina de Ingeniería y Gestión de Software en la Universidad de las Ciencias Informáticas*, Repositorio Digital UCI, Tesis de Maestría. <https://repositorio.uci.cu/jspui/handle/ident/8618>
- Linares Río, M., Aleas Díaz, M., Mena Lorenzo, A., Cruz Marquez, D., y Rosales Quintana, 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-](http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1684-)

18592019000200158

Ministerio de Educación Superior. (2018). *Plan de Estudio E*, Carrera Ingeniería Informática. La Habana, Cuba.

Ministerio de Educación Superior. (2022). *Reglamento organizativo del proceso docente y de dirección del trabajo docente y metodológico para las carreras universitarias*. La Habana, Cuba.

Muñoz Verduga, D. E. (2020). *La formación investigativa de los estudiantes de la carrera de ingeniería en sistemas de la Universidad Laica Eloy Alfaro de Manabí, Ecuador*. (Ed.) Manta - Manabí – Ecuador, Ecuador: Consejo Editorial: Universidad Laica Eloy Alfaro de Manabí ISBN: 978-9942-827-30-2. <https://munayi.uleam.edu.ec/wp-content/uploads/2020/12/la-formacion-investigativa-web.pdf>

Olivera Delgado, L., y Alonso 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](http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=s2227-18992021000100037#aff2)

Piñero Incencio, G. S., Cantero Guerra, L. M., y Rivero Lissabet, J. L. (2022). Caracterización epistemológica del proceso de enseñanza-aprendizaje de la asignatura Ingeniería de Software I. *Edusol*, 22(80). [http://scielo.sld.cu/scielo.php?script=sci\\_arttext&pid=S1729-80912022000300176](http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1729-80912022000300176)

Zurita Cruz, E., Zaldívar Colado, A., Sifuentes Ocegueda, A. T., y Valle Escobedo, R. M. (2020). Análisis crítico de ambientes virtuales de aprendizaje. *Utopía y Praxis Latinoamericana*, 25(11). <https://www.redalyc.org/journal/279/27964922003/html/>