

Aprendizaje creativo de la ingeniería del software en la formación del profesional informático en la Universidad de Matanzas

Creative learning of software engineering in the training of informatics professionals at the University of Matanzas

Daniel Segura Urrutia¹, Briseida Lucía Blanco Alfonso², *Walfredo González Hernández³

^{1,2}Universidad de Matanzas, Carretera a Canímar, Km 3 ½, Matanzas, Cuba. CP.40100.

daniel.segura@nauta.cu, <https://orcid.org/0009-0005-1236-0948>

briseida.blanco@umcc.cu, CP.40100, <https://orcid.org/0000-0002-3717-3058>

³Universidad Central "Marta Abreu" de las Villas. Carretera a Camajuaní Km. 5 ½. Santa Clara. Villa Clara. Cuba.

wghernandez@uclv.cu, <https://orcid.org/0000-0003-4028-4266>.

*Autor de correspondencia

Resumen

La presente investigación se centra en el aprendizaje creativo de la ingeniería del software en la carrera Ingeniería Informática de la Universidad de Matanzas. La ingeniería del software fue seleccionada por las potencialidades que sus contenidos ofrecen para fomentar el aprendizaje creativo durante el proceso de formación del profesional informático. Para su desarrollo se realiza una fundamentación de la creatividad y su relación con el aprendizaje creativo, lo que permite definir el aprendizaje creativo de la ingeniería del software. Se diagnostica el estado actual del aprendizaje creativo en los estudiantes de la carrera objeto de análisis, donde se identifica el insuficiente desarrollo de este. El objetivo es elaborar un sistema de actividades didácticas que contribuye al aprendizaje creativo de la ingeniería del software en el tercer año de la carrera Ingeniería Informática. Por último, se realiza el criterio de especialista que permita valorar la propuesta.

Palabras clave: Aprendizaje creativo; creatividad; ingeniería del software.

Abstract

The present research examines the creative learning of software engineering within the Computer Engineering course at the University of Matanzas. Software engineering was selected for investigation due to its potential to facilitate creative learning during the training of computer professionals. To this end, a foundation of creativity and its relationship with creative learning is established, which allows the definition of creative learning in software engineering. The current state of creative learning in students from the degree course under analysis is then diagnosed, where its insufficient development is identified. The objective is to develop a system of didactic activities that contributes to the creative learning of software engineering in the third year of the Computer Engineering course. Finally, a specialist criterion is used to evaluate the proposal.

Keywords: Creative learning; creativity; software engineering.

Recibido: 24 de julio de 2023

Aceptado: 03 de julio de 2024

Publicado: 09 de octubre de 2024

Cómo citar: Segura Urrutia, D., Blanco Alfonso, B. L., & González Hernández, W. (2024). Aprendizaje creativo de la ingeniería del software en la formación del profesional informático en la Universidad de Matanzas. *Acta Universitaria* 34, e3949. doi: <http://doi.org/10.15174/au.2024.3949>

Introducción

El reto de la educación en las próximas décadas no será, simplemente, el dominio del conocimiento, sino el poder del aprendizaje creativo. Los alumnos del futuro tendrán que aprehender de forma holística, permanente y creativa todo cuanto se conoce y se sabe no es más que una creación de la mente humana, fruto del aprendizaje creativo (Waki de-Oliveira *et al.*, 2023).

El aprendizaje creativo representa un desafío pedagógico para el sistema educativo basado en la transmisión lineal y fragmentada del conocimiento (Ciriello *et al.*, 2024; Korhonen *et al.*, 2024), pues girar y asumir un enfoque donde el centro es el estudiante y el docente se enfoca en orientar, motivar y acompañar el proceso formativo, demanda una asertiva reingeniería curricular de la visión educativa (Zweeris *et al.*, 2023).

En el pasado, el aprendizaje clásico marcó la época, generando gran beneficio y bienestar, pero ahora es necesario un nuevo enfoque porque se vive en una sociedad cambiante que innova y se desarrolla a mayor velocidad, por lo que se requiere una redefinición del modelo de aprendizaje así como de nuevas estrategias (Stephenson, 2023). Sin embargo, la expresión de la creatividad en el aprendizaje creativo se aborda desde la teoría de la subjetividad y promueve la concepción configuracional de la psiquis humana en general y de la creatividad en particular (Torres-Oliveira & Mitjans-Martínez, 2020). Por tanto, el aprendizaje creativo se enmarca en los procesos de aprendizaje de las asignaturas.

Las investigaciones antes mencionadas ofrecen conceptos, acciones e ideas rectoras para el desarrollo del aprendizaje creativo, las cuales constituyen referentes teóricos indispensables para el desarrollo de esta investigación. Según Wang (2022), la ingeniería es una actividad realizada por el hombre con la finalidad de resolver problemas del mundo que le rodea. En consonancia con estos preceptos, la ingeniería informática como una ingeniería en particular tiene la misión de formar profesionales informáticos creativos que resuelvan problemas de naturaleza informática del entorno en que se encuentran insertados, y con ello ofrecer respuestas a las demandas de la sociedad en esta área del conocimiento.

La carrera Ingeniería Informática prepara a los jóvenes universitarios para enfrentarse al futuro que les espera, alimentando la pasión y la imaginación en la resolución de problemas relacionados con los procesos de informatización, de modo que las habilidades en el universo de las tecnologías de la información y las comunicaciones adquiridas durante su formación actúen como una ventaja competitiva en el mundo laboral.

La revisión de documentos metodológicos y la aplicación de otras técnicas descritas en la metodología de la investigación revelan una serie de debilidades en el tratamiento del aprendizaje creativo en la carrera Ingeniería Informática de la Universidad de Matanzas, entre las que se encuentran:

- Predomina el aprendizaje comprensivo (se aprende para saber los contenidos que se les imparte durante la clase, comprenderlo y aplicarlo: pocos se motivan por saber si hay más allá de los referidos contenidos).
- Se aprecia desconocimiento en las concepciones del aprendizaje creativo (generación de ideas, confrontación de lo dado y búsqueda de nuevas soluciones).
- No se aprovechan las potencialidades del trabajo en proyectos y roles que ofrecen los contenidos que se abordan en la carrera (generalmente no se acepta el error como parte del aprender, solo se establece comunicación con los otros en circunstancias puntuales; pocos crean proyectos propios).

- En el diseño curricular de la carrera se identifican las competencias: modelación de proyectos, implementación de proyectos de software, depuración y gestión de proyectos. Los niveles de cada una se catalogan como bajos, según estudios anteriores (González-Hernández, 2016; González-Hernández & Coloma-Carrasco, 2018; González-Hernández, 2019), y se mantienen al ser corroborados en este.

A partir de las debilidades encontradas, se revela una contradicción entre la necesidad del fomento del aprendizaje creativo en la carrera Ingeniería Informática de la Universidad de Matanzas y la realidad actual, reconocida como un aprendizaje comprensivo.

En aras de profundizar en el estudio de la problemática planteada y obtener una propuesta de mejora, se conduce el presente estudio, tomando como referente los contenidos de ingeniería del *software* de la carrera objeto de análisis. Se enuncia como pregunta de investigación: ¿Cómo fomentar el aprendizaje creativo en la carrera Ingeniería Informática de la Universidad de Matanzas?

Para profundizar en el estudio de la problemática planteada y obtener una propuesta de mejora, se selecciona la ingeniería del *software* para la introducción del aprendizaje creativo en la referida carrera. Para responder este problema científico se estructuran tres momentos fundamentales. En un primer momento se define el aprendizaje creativo de la ingeniería del *software*, en un segundo momento se estructura el sistema de actividades y en un tercer momento se valida el sistema propuesto.

Desarrollo

La complejidad de los nuevos sistemas y productos basados en computadora demanda atención cuidadosa a las interacciones de todos los elementos del sistema en los que el diseño se ha vuelto una actividad crucial, y para su diseño se requiere de la ingeniería del *software* como disciplina (Monat *et al.*, 2022). La ingeniería del *software* es una disciplina de la ingeniería que está relacionada con todos los aspectos de la producción de *software* desde las etapas tempranas de especificación del sistema hasta el mantenimiento del sistema después que ha entrado en uso (Williams & Rainer, 2019).

Los contenidos de Ingeniería del Software ofrecen potencialidades para el tratamiento del aprendizaje creativo de forma integral, desde su comprensión en la interrelación del aprendizaje y la creatividad, con el fin de influir desde los planos cognitivos y axiológicos en los modos de actuación de los estudiantes.

Los contenidos de Ingeniería del Software tienen como objetivo preparar al estudiante en las actividades integrales del ciclo de vida de un proyecto de *software*, así como en cada una de las etapas de dicho ciclo de vida y en las distintas actividades de un ingeniero informático. Incluyen además el estudio de las metodologías, con el propósito de introducir métodos y procedimientos que ayuden al éxito de un proyecto *software*; esta disciplina ofrece al estudiante una visión de la necesidad de la Ingeniería del Software, su origen y evolución, sus objetivos, sus funciones, sus motivaciones y del papel del ingeniero de *software* en ella.

Con el estudio de la Ingeniería del Software se pretende acercar al estudiante al mundo profesional y al desarrollo de *software* en las empresas, explicando y proporcionándoles técnicas, métodos, procedimientos y herramientas para las distintas actividades que tendrán que realizar durante su desempeño profesional. También conocerá la situación actual de los proyectos de *software* en las empresas, su evolución y tendencias.

Mediante la Ingeniería del Software el estudiante adquiere la habilidad de desarrollar, mantener y evaluar servicios y sistemas de *software* que satisfagan los requisitos del usuario, que se comporten de forma fiable y eficiente, que sean asequibles de desarrollar y mantener los sistemas de software y que cumplan normas de calidad.

El estudio de las teorías, principios, métodos y prácticas de la Ingeniería del Software permite que el estudiante adquiera la capacidad de identificar y analizar problemas, así como diseñar, desarrollar, implementar, verificar y documentar soluciones de *software* sobre la base de un conocimiento adecuado de las teorías, modelos y técnicas actuales (Méndez-Fernández & Passoth, 2019).

Como resultado del aprendizaje de los contenidos de Ingeniería del Software incluidos en el currículo establecido por la Comisión Nacional de Carrera para el Ingeniero Informático (González *et al.*, 2021), el estudiante debe:

- Dominar las actividades tanto de gestión como integrales del ciclo de vida del software y tener dominio de notaciones, técnicas, métodos y herramientas para la realización de las referidas actividades.
- Dominar las fases técnicas del ciclo de vida del software: análisis, diseño, implementación y pruebas; y tener dominio de notaciones y lenguajes, técnicas, métodos y herramientas para la realización de estas.
- Dominar los aspectos profesionales de la Ingeniería del Software.
- Aplicar la accesibilidad universal, ergonomía, usabilidad, internacionalización, seguridad, adecuación a la legislación, normativa y estándares vigentes de los sistemas, aplicaciones y servicios informáticos, así como la información que proporcionan durante el diseño y desarrollo de ellos.

Los contenidos declarados en los párrafos anteriores coinciden con los planteados en una amplia literatura que aborda la formación de ingenieros informáticos (González-Hernández *et al.*, 2022a; Guo *et al.*, 2022; Opdahl, 2024).

La ingeniería del software es la columna vertebral de la carrera, pues es la disciplina que estudia los procesos de desarrollo de *software* (Alvarez, 2024; Soto-Duran *et al.*, 2010). Para lograr aprender esos contenidos, en el currículo se colocan las asignaturas Ingeniería de Requisitos, Diseño Orientado a Objetos y Patrones de Diseño, que versan sobre los ciclos y metodologías de desarrollo. Estas asignaturas están desde el segundo semestre de segundo año hasta el segundo semestre de tercer año. Posteriormente, en cuarto año se encuentran Gestión de Pruebas para detectar errores en el sistema y corregirlos y Gestión de la Usabilidad para aprender los estándares y técnicas para desarrollar un sistema usable. Todas estas asignaturas pertenecen a la disciplina Ingeniería y Gestión de Software.

Los requerimientos de la tecnología de la información que demandan los individuos, negocios y gobiernos se hacen más complejos con cada año que pasa. En la actualidad, grandes equipos de personas crean programas de cómputo que antes eran elaborados por una sola persona. El *software* actual, que alguna vez se implementó en un ambiente de cómputo predecible y su contenido, se halla incrustado en el interior de todo, desde la electrónica de consumo hasta dispositivos médicos o sistemas de armamento. La Ingeniería del Software representa un elemento crucial en el desarrollo de las aplicaciones de *software* que demanda la sociedad en general (Kruchten *et al.*, 2019; Mellor, 2023).

La ingeniería de *software* busca apoyar el desarrollo de *software* profesional en equipos de desarrollo, en lugar de la programación individual. Incluye técnicas que apoyan la especificación, el diseño y la evolución del programa, las cuales son relevantes para el desarrollo de *software*. La disciplina se estructura con el objetivo de ayudar al estudiante a obtener una amplia visión de los procesos ingenieriles en su campo de acción como ingeniero informático (Uzair-Khan *et al.*, 2019).

A partir de las concepciones referentes a la creatividad, su relación con el aprendizaje creativo y la caracterización de ingeniería del software, antes expuesta, se hace necesario para esta investigación expresar una definición de aprendizaje creativo de la ingeniería del software.

El aprendizaje creativo de la ingeniería del software se define como: la forma de aprender los contenidos relativos a la ingeniería del software, caracterizado por los tres procesos principales: 1) personalización de los contenidos, 2) confrontación con lo transmitido y 3) producción de nuevas ideas relacionadas con el contenido de la ingeniería del software, que permita resolver problemas de naturaleza informática en las organizaciones.

Para caracterizar el aprendizaje creativo de la Ingeniería del Software, se toman como referentes las dimensiones e indicadores definidas por González-Hernández *et al.* (2022a) ajustadas al contenido de Ingeniería del Software, las cuales se muestran en la Tabla 1.

Tabla 1. Dimensiones e indicadores del aprendizaje creativo de la Ingeniería del Software.

Dimensión 1. Personalización de los contenidos que conforman la Ingeniería del Software.	
Indicadores	Criterios de medida
1.1 Elabora una síntesis personalizada de las metodologías y sus artefactos, las tecnologías y las herramientas que conforman la Ingeniería del Software.	Utiliza comúnmente el lenguaje simbólico de la Ingeniería del Software para representar la solución de diversas situaciones problemáticas de la vida diaria.
	Extrae los elementos principales de las metodologías y sus artefactos, las tecnologías y las herramientas relacionados con la Ingeniería del Software útiles al proyecto.
	Compara las diversas metodologías y sus artefactos, las tecnologías y las herramientas relacionados con la Ingeniería del Software para aplicarlos en los roles que les corresponden.
	Elabora resúmenes sobre los diversos usos de las metodologías y sus artefactos, las tecnologías y las herramientas relacionadas con la Ingeniería del Software.
1.2 Procesa el reconocimiento de las metodologías y sus artefactos, las tecnologías y las herramientas que conforman la Ingeniería del Software.	Expresa sus ideas sintetizando las metodologías y sus artefactos, las tecnologías y las herramientas relacionados con la Ingeniería del Software.
	Determina los elementos más importantes a resumir de las metodologías y sus artefactos, las tecnologías y las herramientas relacionadas con la Ingeniería del Software, o no, necesarias al proyecto.
	Expresa los elementos esenciales identificados utilizando las formas simbólicas descritas en la de la Ingeniería del Software.
	Establece relaciones entre las metodologías y sus artefactos, las tecnologías y las herramientas que componen la Ingeniería del Software, o no, recibidos y lo que ya conoce.
1.3 Discrimina los contenidos relevantes a partir de los conocimientos que posee sobre Ingeniería del Software obtenidos en los diversos espacios de aprendizaje.	Selecciona la forma de registrar la información utilizando las formas simbólicas de la Ingeniería del Software.
	Registra las metodologías y sus artefactos, las tecnologías y las herramientas relacionados con la Ingeniería del Software, utilizando otras formas que viabilicen la comunicación asincrónica entre los integrantes del proyecto.
	Contrasta la información sobre Ingeniería del Software aportada por el resto de los integrantes del proyecto con las que posee.
	Determina los aspectos relevantes en la Ingeniería del Software que recibe.
	Establece relaciones no lineales acerca de la aplicación, o no, de las metodologías y sus artefactos, las tecnologías y las herramientas de Ingeniería del Software durante la ejecución de los procesos relacionados con el proyecto.
	Compara los procesos y las informaciones relacionadas con la Ingeniería del Software obtenidas en el proyecto para tomar decisiones.
	Actúa en consonancia con la información que considera relevante en dependencia de los roles asignados.

Fuente: Elaboración propia.

Tabla 1. Dimensiones e indicadores del aprendizaje creativo de la Ingeniería del Software (continuación).

Dimensión 1. Personalización de los contenidos que conforman la Ingeniería del Software.		
Indicadores	Criterios de medida	
1.4	Elabora escenarios personalizados de aplicación de las metodologías y sus artefactos, las tecnologías y las herramientas que conforman la Ingeniería del Software, necesarios en los procesos de informatización.	Elabora nuevos diseños de aplicación de las metodologías y sus artefactos, las tecnologías y las herramientas relacionados con la Ingeniería del Software necesarios al proyecto. Verifica lógicamente las metodologías y sus artefactos, las tecnologías y las herramientas relacionados con la Ingeniería del Software producidos en el marco del proyecto en el que se inserta. Argumenta la viabilidad de la aplicación de las metodologías y sus artefactos, las tecnologías y las herramientas relacionados con la Ingeniería del Software producidos en el marco del proyecto en el que se inserta.
	Aplica las metodologías y sus artefactos, las tecnologías y las herramientas que conforman la Ingeniería del Software aprendidas, a la solución de problemas informáticos en los diversos espacios de aprendizaje.	Reflexiona acerca de la factibilidad de introducir a las situaciones del proyecto las metodologías y sus artefactos, las tecnologías y las herramientas relacionados con la Ingeniería del Software obtenidos en los espacios de aprendizaje. Verifica la viabilidad de la introducción a las situaciones específicas del proyecto, las metodologías y sus artefactos, las tecnologías y las herramientas relacionadas con la Ingeniería del Software obtenidas en los espacios de aprendizaje. Introduce las metodologías y sus artefactos, las tecnologías y las herramientas relacionadas con la Ingeniería del Software, obtenidas en los espacios de aprendizaje, a las situaciones específicas del proyecto. Determina la validez de la introducción de las metodologías y sus artefactos, las tecnologías y las herramientas relacionadas con la Ingeniería del Software obtenidas en los espacios de aprendizaje a las situaciones específicas del proyecto.
Dimensión 2. Confrontación de los nuevos contenidos de la Ingeniería del Software con los contenidos ya dados.		
Indicadores	Criterios de medida	
2.1	Cuestiona las metodologías y sus artefactos, las tecnologías y las herramientas que conforman la Ingeniería del Software obtenidas en los diversos espacios de aprendizaje.	Realiza preguntas originales que demuestran reflexión acerca de las informaciones relacionadas con la Ingeniería del Software recibidas. Cuestiona las informaciones relacionadas con la Ingeniería del Software que recibe en el marco del proyecto. Identifica contradicciones y brechas en el conocimiento. Determina analogías de las informaciones relacionadas con la Ingeniería del Software recibidas en el marco del proyecto con las que ya posee. Argumenta sus posiciones sobre la base de la búsqueda de informaciones relacionadas con la Ingeniería del Software.
	Argumenta los procesos de desarrollo del proyecto que considera más eficientes y eficaces, a partir de los contenidos que conforman la Ingeniería del Software obtenidos en los diversos espacios de aprendizaje donde se implica.	Interpreta la información relacionada con la Ingeniería del Software necesaria en los proyectos, que obtiene de los espacios de aprendizaje donde se involucra. Busca en otras fuentes los juicios que corroboran el juicio inicial. Selecciona las reglas lógicas que sirven de base al razonamiento. Elabora conclusiones acerca de los elementos, relaciones y razonamientos que aparecen en el objeto o información a interpretar. Utiliza correctamente la simbología de la Ingeniería del Software que le permite expresar sus resultados al resto de los integrantes del proyecto.
	Argumenta las tensiones que detecta durante la ejecución de los procesos de desarrollo del proyecto, a partir de las metodologías y sus artefactos, las tecnologías y las herramientas que conforman Ingeniería del Software, obtenidos en los espacios de aprendizaje donde se implica.	Interpreta la información asociada a las tensiones que se manifiestan durante la ejecución del proyecto. Busca en otras fuentes las opciones para minimizar las tensiones que suceden durante la ejecución del proyecto. Selecciona las reglas lógicas que sirven de base al razonamiento para participar como mediador de las tensiones que ocurren durante la ejecución del proyecto. Elabora las conclusiones acerca de las tensiones que ocurren durante la ejecución del proyecto.
2.4	Selecciona las personas con mayores potencialidades para conformar los equipos de desarrollo del proyecto que considera a partir de criterios de selección obtenidos en los diversos espacios de aprendizaje donde se implica.	Determina criterios de selección del personal a participar en las tareas asociadas al proyecto. Elabora instrumentos para determinar las potencialidades y debilidades de las personas que van a trabajar en el proyecto. Aplica criterios de selección teniendo en cuenta las características de los roles a ejecutar. Potencia las relaciones interpersonales entre los integrantes del equipo de proyecto.

Fuente: Elaboración propia.

Tabla 1. Dimensiones e indicadores del aprendizaje creativo de la Ingeniería del Software (continuación).

Dimensión 3. Producción, generación de ideas propias y nuevas basadas en los contenidos que conforman la Ingeniería del Software, que le permitan la solucionar problemas informáticos en las organizaciones.	
Indicadores	Criterios de medida
3.1 Propone ideas y alternativas para solucionar problemas informáticos en las organizaciones, a partir de la aplicación de los nuevos aprendizajes relacionados con la Ingeniería del Software, obtenidos en los espacios de aprendizaje.	Participa activamente en la búsqueda de nuevas ideas, alternativas, conjeturas e hipótesis que permitan obtener procesos eficientes y eficaces.
	Muestra confianza en sí mismo, autonomía, iniciativa y perseverancia.
	Propone alternativas e hipótesis ante los problemas a resolver en el marco de los proyectos donde interactúa.
	Produce ideas nuevas relacionadas con las metodologías y sus artefactos, las tecnologías y las herramientas relacionadas con la Ingeniería del Software o no necesarias al proyecto. Comunica a los integrantes del proyecto las nuevas informaciones, ideas, alternativas, conjeturas e hipótesis relacionadas con la Ingeniería del Software, que resultan relevantes para el proyecto.
3.2 Elige la vía más eficiente y eficaz en el momento de solucionar problemas informáticos en las organizaciones, a partir de las nuevas metodologías y sus artefactos, las tecnologías y las herramientas que conforman la Ingeniería del Software, obtenidas en los espacios de aprendizaje.	Identifica diferentes vías de solución a las problemáticas que se plantean en el marco del proyecto.
	Utiliza los espacios de aprendizaje para indagar, buscar más información relacionada con la Ingeniería del Software que permita responder a inquietudes, curiosidades y necesidades propias del proyecto.
	Compara diferentes vías o caminos para determinar la más adecuada en la resolución de la situación planteada en el proyecto.
	Determina la mejor vía de solución a la problemática asociada al proyecto.
	Evalúa la mejor vía de solución a la problemática asociada al proyecto.
3.3 Elabora nuevos proyectos de software que provoquen satisfacción por lo realizado y generación de nuevas ideas vinculadas a su formación como ingeniero informático.	Presenta a los integrantes del proyecto las posibles vías para dar respuesta la situación planteada.
	Elabora nuevas situaciones que generalizan las situaciones actuales a nuevos contextos.
	Concretas situaciones nuevas que puedan aplicarse el marco de trabajo obtenido en el proyecto. Busca nuevas situaciones o procesos a informatizar en otras organizaciones.

Fuente: Elaboración propia.

Los contenidos de Ingeniería del Software brindan amplias potencialidades para contribuir al desarrollo de la creatividad en los estudiantes de la carrera Ingeniería Informática.

Materiales y métodos

El artículo es de investigación con un enfoque mixto, donde se maneja un diseño transversal exploratorio con elementos descriptivos (Hernández-Sampieri *et al.*, 2014) para examinar el ACI y describir el criterio de especialistas.

En Cuba, las carreras tienen tres espacios curriculares: el base, el propio y el optativo/electivo (González-Hernández, 2022). El currículo base es obligatorio para todas las carreras de Ingeniería Informática del país y cuenta con un número de horas de cumplimiento obligatorio. El currículo propio tiene definido la cantidad de horas a nivel nacional, pero los contenidos lo deciden las universidades para incluir asignaturas que respondan a necesidades del territorio. Y el optativo/electivo también tiene un conjunto de horas definido nacionalmente, pero pueden ser asignaturas que no sean de la carrera y que respondan a necesidades de investigación de los estudiantes. La carrera de Ingeniería Informática posee un diseño curricular de cuatro años. El proceso de formación está estructurado por disciplinas que agrupan las asignaturas por disciplinas científicas del área del conocimiento.

La Ingeniería Informática posee tres disciplinas fundamentales: Ingeniería y Gestión de Software, Infraestructuras Informáticas e Inteligencia Computacional. Los contenidos relacionados con la ingeniería del *software* están en la primera disciplina juntamente con los de programación. La Ingeniería del Software se comienza a impartir en el segundo año y se plantean tres asignaturas del currículo base. Se comienza con Ingeniería de Requisitos, posteriormente Diseño Orientado a Objetos y, por último, Patrones de Diseño. Después se dedican dos asignaturas más del currículo propio a contenidos de pruebas y usabilidad que se corresponden con la disciplina Ingeniería y Gestión de Software. Los roles declarados en el plan de estudios para estas asignaturas son analista de requisitos, analista, diseñador de *software*, probador, gestor y programador (González *et al.*, 2021). Hasta el momento se utilizan varios enfoques de enseñanza de la informática (González-Hernández, 2015), predominando los métodos activos propuestos en Mi *et al.* (2022).

Para evaluar el aprendizaje creativo de la ingeniería del *software* se tomó el tercer año de la carrera de Ingeniería Informática porque es donde se comienza a recibir dicha asignatura. En este año se reciben los elementos esenciales de Ingeniería de Requisitos y Diseño de Software y son las únicas que los abordan. Los profesores de estas asignaturas son los únicos encargados de enseñarles los diagramas UML (*Unified Modelling Language*) hasta llegar a la arquitectura.

El grupo de tercer año solamente presenta 11 estudiantes en la provincia de Matanzas. La encuesta a profesores (EP) se aplicó para conocer la valoración que poseen los profesores acerca del aprendizaje creativo de la informática. Se aplicó una encuesta a estudiantes (EE) para detectar la valoración que tienen los estudiantes de su aprendizaje creativo, así como las opiniones que ellos poseen acerca de su aprendizaje. Se realizó una observación a clases (OC) debido a que se consideró importante que los investigadores pudieran apreciar el comportamiento de los estudiantes. Se realiza una revisión de los informes de la práctica laboral (OPL), pues en ella es donde se aplican los contenidos de IS (Ingeniería del Software) para resolver problemáticas de la organización relacionadas con su transformación digital.

Los cuestionarios se derivan del cuestionario para aprendizaje creativo elaborado por Marcelina Moreno (Moreno-García, 2019), el cual fue perfeccionado posteriormente por (González-Hernández *et al.*, 2022b) para evaluar aprendizaje creativo de la informática. Todos los indicadores obtenidos en la operacionalización de la variable se miden en los métodos declarados. En cada uno de los criterios de medida se asume un valor de 0 a 1, siguiendo la siguiente escala de evaluación: verdadera (1), casi verdadera (0.9), bastante verdadera (0.8), algo verdadera (0.7), más verdadera que falsa (0.6), tan verdadera como falsa (0.5), más falsa que verdadera (0.4), algo falsa (0.3), bastante falsa (0.2), casi falsa (0.1), falsa 0. Se asume como hipótesis de trabajo que más del 80% de los estudiantes de la carrera poseen un aprendizaje creativo de la ingeniería del *software* evaluado como bien o excelente, por ser estos contenidos el centro de su formación profesional y por estar en año terminal.

Evaluar el desarrollo de la creatividad informática es complejo para un investigador novel, y sería muy conveniente en estos casos -así como para la selección del personal para un proyecto- tener medidas cuantitativas en dependencia de la importancia que tenga cada dimensión para la organización. La métrica para determinar el nivel de creatividad evaluada desde un método (C1M) se muestra en la ecuación 1:

$$C1M = \sum_{i=1}^n P_i D_i \quad (1)$$

donde h es la cantidad de dimensiones, D_i es la evaluación de la i -ésima dimensión y P_i es el peso.

$$D_i = \frac{1}{m_i} \sum_{j=1}^{m_i} P_{ij} \frac{1}{n_{ij}} \sum_{k=1}^{t_{ij}} I_{ijk} \quad (2)$$

donde mi es el total de indicadores de la dimensión i , tij es el total de aspectos a evaluar para el indicador j de la dimensión i , y $Iijk$ es la evaluación dada al aspecto k del indicador j en la dimensión i .

Para determinar el peso de cada dimensión, se recomienda el método de comparaciones pareadas, el cual, a pesar de clasificar dentro de los métodos de ponderación subjetivos (Herrera-Masó *et al.*, 2022) donde los especialistas que se consultan muestran sus preferencias entre criterios, es posible cuantificar la intensidad de esa preferencia utilizando la escala de valoración propuesta por Saaty (1987). Al aplicar las matrices de comparaciones pareadas para cada experto y los vectores de preferencias para tres dimensiones, se obtiene que los resultados son agregados, utilizando la media geométrica hasta llegar a un nuevo vector de prioridad consensuado, que es 0.163, 0.297, 0.54.

Una vez determinada la métrica de evaluación de la creatividad, es necesario obtener la escala de medición, y para ello es necesario obtener el intervalo $[VMínimo, VMáximo]$. El valor máximo obtenido $VMáximo$ es el valor que se obtiene de sustituir $Iijk$ con el valor máximo que otorga el investigador para cada aspecto en la ecuación (3):

$$escala_{indi} = (VMínimo - ValMaxInter_1, VMínimoInter_2 - ValMaxInter_2, \dots, VMínimoInter_{n-1} - VMáximo) \quad (3)$$

El valor mínimo $VMínimo$ es el valor que se obtiene de sustituir $Iijk$ con el valor mínimo que otorga el investigador para cada aspecto en la ecuación (2). Una vez que se obtengan estos valores, solo queda determinar la equivalencia con los valores de la escala que posea el evaluador en su país, y para ello se tiene en cuenta los intervalos de evaluación del sistema evaluativo, al cual se le llamará $NumInterva$. En el caso de esta investigación, el valor máximo es 5 y el mínimo es 2, teniendo en cuenta el sistema educativo cubano, por lo que son cuatro intervalos. Para obtener el valor máximo y mínimo de cada intervalo es necesario calcular $(VMáximo)/NumInterva$, lo que dará el número máximo del primer intervalo y se denomina $ValIntervalo$, por lo que $VMáximo_1 = VMínimo + ValIntervalo$. Para determinar el mínimo del segundo intervalo, se calcula $VMínimo_2 = VMáximo_1 + \frac{1}{10^{n_1+1}}$ siendo n la cantidad de decimales que posee $VMáximo_1$. En el caso del último valor mínimo de la escala, se calcularía: $VMínimo_x = VMáximo_{x-1} + \frac{1}{10^{n_{x-1}+1}}$, y el valor máximo sería $VMáxi$.

Los sistemas de evaluación cuantitativos en el mundo no pasan de dos cifras decimales, por lo que la suma del valor propuesto no ofrece una diferencia significativa y permite establecer claramente los límites del intervalo. De esta manera es posible evaluar la creatividad informática, independientemente de la escala de evaluación y los criterios que asuman en los diferentes países, homogeneizando la escala y posibilitando que sea comprensible para cualquier evaluador.

La integración de los métodos para evaluar la creatividad depende del concepto que se asuma con sus dimensiones e indicadores. Ello hace que cada investigación deba asumir la integración de diferentes métodos, y por ello no siempre se puede asumir un conjunto de métodos estándar para todas. En este caso, es necesario estructurar una configuración de métodos que sea coherente con el concepto de creatividad asumido por los investigadores.

En el caso que todos los criterios de medida sean evaluados en cada uno de los instrumentos, entonces los indicadores se calculan utilizando la siguiente ecuación: $Ind = \sum_{i=1}^l (\sum_{j=1}^m P_j \cdot C_{ij}) / m$ (4, donde P_j es el valor del peso de cada método, C_{ij} es el valor del criterio de medida y m es la cantidad de métodos aplicados.

En el caso que haya criterios de medida que no sean aplicados en algún método, entonces se propone la siguiente ecuación para la configuración:

$$Ind = \sum_{i=1}^I \left(\frac{1}{M_{ijh}} \sum_{m=1}^{M_{ijh}} \frac{P_m}{\sum_{l=1}^{M_{ijh}} P_l} C_{ijhm} \right) \quad (5)$$

donde M es la cantidad de métodos, C_{ijhm} es la evaluación del criterio de medida del indicador que está en cada dimensión, M_{ijh} es el número de métodos que miden el criterio i del indicador j de la dimensión h , P_m es el peso de cada método porque cada uno de los métodos tendrá un peso diferente en dependencia del criterio de los expertos que será tomado en cuenta utilizando el panel para esos efectos para el caso de Ind .

En el caso de la evaluación de la dimensión, la métrica sería:

$$D_h = \frac{1}{J_n} \sum_{j=1}^{J_n} P_j I_j \quad (6)$$

donde J_n sería la cantidad de indicadores de la dimensión, P_j sería el peso de cada indicador e I_j es el valor de cada indicador obtenido en la ecuación 4.

Para la evaluación de la variable, la expresión sería:

$$Var = \frac{1}{H} \sum_{h=1}^H P_h D_h \quad (7)$$

donde H es el total de dimensiones y P_h es el peso de la dimensión.

Para calcular la escala se aplica el mismo procedimiento que para el caso de un único método (5) y para sustituirlo en la (6) y (7) se debe tomar como valor máximo la resultante de la siguiente expresión:

$$ValorMax = \frac{1}{M} \sum_{i=1}^M ValorMax_{P_i C_1 M_i} \quad (8)$$

donde $ValorMax_{P_i C_1 M_i}$ es el valor que toma cada método si cada criterio de medida asume el mayor valor posible y lleva a la variable medida en ese método a tomar el mayor valor posible.

Las métricas obtenidas se ajustan a la definición de sistema de métodos, pues evalúan las relaciones entre los métodos y no son dependientes de la cantidad que sean utilizados. La relación entre ellos es la media del producto entre los pesos y la sumatoria de criterios, pues logra homogeneizar las diferencias de criterios de medidas y la cantidad de dimensiones e indicadores de cada tipo de método. Cada uno de ellos mide indicadores diferentes y calcula una visión diferente de la misma variable, reforzando su carácter configuracional. Aunque esta métrica fue desarrollada para medir la creatividad, puede ser utilizada para medir cualquier variable cuya operacionalización se realice descomponiéndola en dimensiones e indicadores.

Asumiendo que la escala de evaluación cubana es de 2 a 5, se adopta la escala calculada para el aprendizaje creativo de la IS en la Tabla 2.

Tabla 2: Escala Evaluativa.

Escala Cualitativa	Escala Cuantitativa	
	Valor Inicial	Valor Final
Excelente	0.753	1
Bien	0.502	0.752
Regular	0.251	0.501
Mal	0	0.25

Fuente: Elaboración propia.

Resultados

Una vez aplicados los métodos, se obtiene la Tabla 3, que permite visualizar los resultados de la evaluación del aprendizaje creativo en cada uno de los métodos.

Tabla 3. Resultados de la aplicación de cada método.

Método	EE		EP		OC		RIPL	
	Cuantitativa	Cualitativa	Cuantitativa	Cualitativa	Cuantitativa	Cualitativa	Cuantitativa	Cualitativa
E1	0.76958	Excelente	0.784425	Excelente	0.784425	Excelente	0.784425	Excelente
E2	0.6245	Bien	0.58515	Bien	0.6245	Bien	0.6245	Bien
E3	0.63285	Bien	0.536925	Bien	0.63285	Bien	0.63285	Bien
E4	0.3445	Regular	0.192925	Mal	0.382925	Regular	0.42635	Regular
E5	0.52503	Bien	0.417075	Mal	0.42135	Regular	0.405	Regular
E6	0.63843	Bien	0.29525	Mal	0.407425	Regular	0.425425	Regular
E7	0.6686	Bien	0.248625	Mal	0.389425	Regular	0.469575	Regular
E8	0.52255	Bien	0.4721	Mal	0.43305	Regular	0.436	Regular
E9	0.5772	Bien	0.30045	Mal	0.40335	Regular	0.4132	Regular
E10	0.41053	Regular	0.33285	Mal	0.436925	Regular	0.4067	Regular
E11	0.2981	Regular	0.54795	Regular	0.45085	Regular	0.428775	Regular

Fuente: Elaboración propia.

Los resultados de la aplicación de los diferentes métodos se grafica en la Figura 1, de manera que se puedan visualizar mejor los resultados.

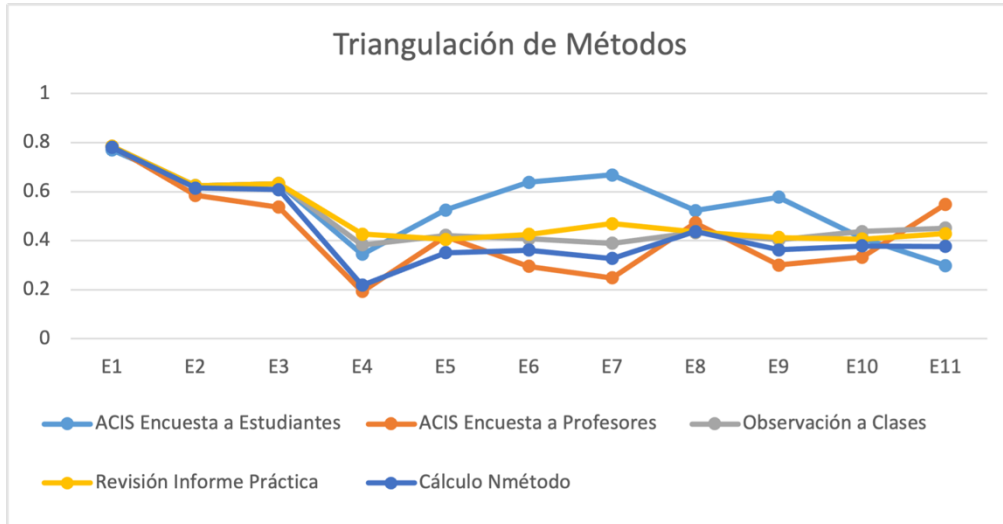


Figura 1. Representación de la triangulación de los resultados.
Fuente: Elaboración propia.

Como se puede apreciar en la gráfica, la encuesta de los estudiantes obtiene valores por encima de los restantes métodos, desde el estudiante 5 hasta el estudiante 10, lo que constituye el 50% del grupo. Ello implica que la autovaloración de los estudiantes del 5 al 10 está por encima de sus posibilidades reales. En la observación a clases se aprecia que los estudiantes se autoevalúan positivamente en las problemáticas complejas que no pueden resolver, tratan de culpar a los demás por el fracaso cuando trabajan en equipos, y al trabajar solos justifican resultados que no son adecuados en relación con el proceder de un informático -como son el uso de artefactos inadecuados, inadecuada selección de la metodología, incorrecto modelo de desarrollo, entre otros.

La dimensión más afectada es la tercera, donde los estudiantes reciben puntuaciones más bajas en los tres métodos aplicados. Generalmente, trabajan por implementar soluciones a las problemáticas tratadas en las conferencias, sin percatarse que cada problema relacionado con el desarrollo de un sistema informático es único e irrepetible, totalmente dependiente de los requisitos donde se expresan las necesidades del cliente. Tampoco se percatan de la necesidad de explorar la infraestructura donde será desplegado el futuro sistema, llevándolos a una incorrecta selección de las herramientas y tecnologías de desarrollo. Para levantar requisitos deben ser acompañados por el profesor, pues no logran estructurar métodos para extraer información del cliente, por lo que carecen de recursos para elaborar nuevos proyectos de *software*.

Los estudiantes 1, 2 y 3 obtienen evaluaciones entre bien y excelente en todos los métodos, mientras que con el resto no sucede así. Se realizó una entrevista adicional con estos estudiantes y manifiestan que desde pequeños la familia les enseñó que no deben aprender lo que se les enseña en clases, sino que deben profundizar con otras bibliografías que les permita buscar más allá de los contenidos estudiados en las clases. También les enseña a proponer ideas nuevas, aunque se alejen de lo que está establecido, y proponerlas aun cuando estén equivocados. Refiere que consideran estas asignaturas la columna vertebral de la carrera y que le prestan especial atención a buscar toda la información, aplicar los signos y símbolos propios de la ingeniería del *software*, donde sobresale UML como el lenguaje de modelado más usado. De la misma manera, producen esquemas utilizando símbolos que UML no posee para describir algunas actividades, sobre todo en el negocio, algunos de ellos muy parecidos a BPMN. En el caso de los estudiantes restantes, solamente aprenden para obtener buenas calificaciones y comprender los procesos de ingeniería del *software*, como artefactos y metodologías para aplicarlos en los proyectos de *software* de la manera en que los profesores les indican deben hacerlo.

En la entrevista, estos estudiantes refieren sentirse solos en el grupo, pues el resto no los acompañan en el estudio ni marchan de conjunto con ellos en los equipos que conforman los profesores. En muchas ocasiones tienen que realizar tareas en colectivo con estudiantes que no están interesados en aprender porque los profesores forman los equipos. Al estar integrados en estos equipos donde los otros no les interesa buscar ni producir elementos nuevos en informática, ellos se sienten mal y cargan con todo el peso del trabajo que donde deben producir un algoritmo, modelo u otro tipo de resultado en la informática.

Para corroborar esta afirmación se realizó un sociograma que permite hacer un análisis de la conformación del grupo bajo las oraciones: "¿Con quién te gustaría estudiar Ingeniería del Software en tu grupo? Escribe dos líneas del por qué tu elección". A partir de las respuestas se construye un mapa donde los estudiantes son los nodos y las flechas indican con quién es el estudiante que prefieren estudiar; en caso de que exista una doble saeta, quiere decir que cada uno prefiere estudiar con el otro y viceversa. Los resultados se muestran en la Figura 2.

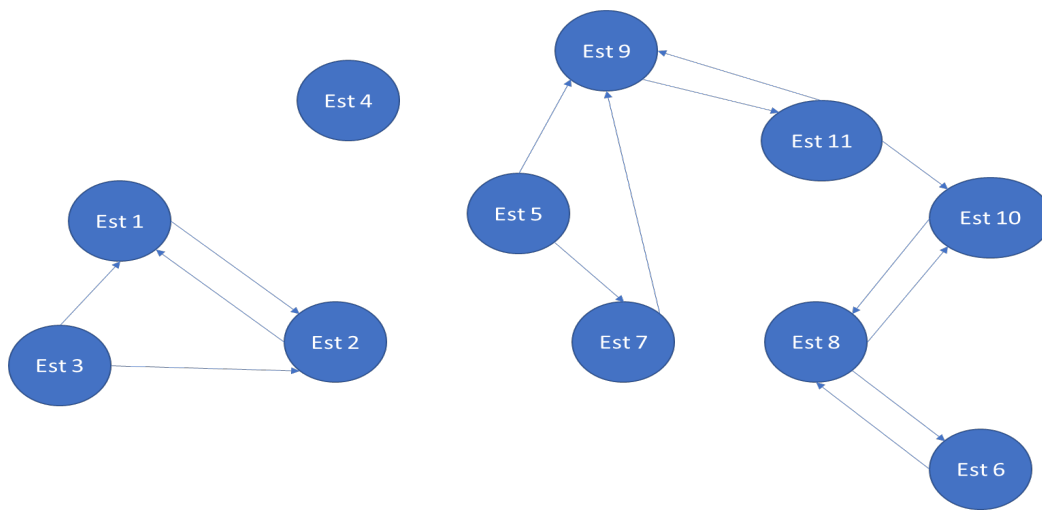


Figura 2. Resultado del sociograma.
Fuente: Elaboración propia.

El esquema presentado corrobora la afirmación de los estudiantes 1, 2 y 3 en cuanto a que ellos se sienten aislados del grupo y no los toman en cuenta para estudiar en las preferencias del grupo. Sucede porque el resto de los estudiantes solamente aprenden para comprender los contenidos que les imparten los profesores, utilizarlos en los ejercicios o problemas que deben resolver y profundizan cuando es una exigencia del proceso. No son capaces de buscar informaciones contradictorias en otros espacios que les permitan confrontar con lo dado en clases, por lo que no se sienten a gusto con estudiantes que sí lo hacen. Las frases más recurrentes en la argumentación son: "porque se utilizan los apuntes y conferencias del profesor, así vamos al seguro y no buscamos nada más". Uno de los estudiantes no prefiere estudiar con alguien del grupo pues refiere que es "mejor estudiar con especialistas de alta experiencia que le puedan enseñar las cuestiones importantes que él/ella desconoce". Ello demuestra dependencia de un adulto o alguien que sabe más para validar la información necesaria. El resto de los estudiantes seleccionan a los compañeros que prefieren estudiar teniendo en cuenta las relaciones de amistad y no el conocimiento que puedan poseer sobre ingeniería del *software*.

Se realiza un análisis de varianza (Anova) de dos factores para determinar si los promedios de los resultados obtenidos con la aplicación de cada método empírico son estadísticamente iguales. El cálculo de Anova se realiza primero con todos los datos obtenidos utilizando el complemento de análisis de datos en Microsoft Excel y se obtiene la Tabla 4.

Tabla 4. Resultado de aplicar Anova.

Anova						
Source of Variation	SS	df	MS	F	p-value	F crit
Rows	0.032118	3	0.010706	4.18178	0.014861	2.960351
Columns	0.341776	9	0.037975	14.8332	2.59E-08	2.250131
Error	0.069124	27	0.00256			
Total	0.443017	39				

Fuente: Elaboración propia.

El análisis de Anova se tomó en las columnas y filas, pero el más interesante para esta investigación es el de las filas, pues compara los valores obtenidos en cada método. El valor de F: 14.83 es mayor que el valor de F crítico = 2.25, por lo que se asume que los promedios de cada grupo son estadísticamente diferentes. Sin embargo, si se eliminan los datos resultantes de la aplicación de la encuesta a los estudiantes, se obtiene la Tabla 5.

Tabla 5. Resultados de Anova sin la encuesta a estudiantes.

Source of Variation	SS	df	MS	F	P-value	F crit
Rows	0.000367	1	0.000367	0.683764	0.429671	5.117355
Columns	0.140298	9	0.015589	29.0089	1.35E-05	3.178893
Error	0.004836	9	0.000537			
Total	0.145502	19				

Fuente: Elaboración propia.

En este caso, $F = 0.68$ es menor que F crítica = 5.12, lo que evidencia que son estadísticamente iguales los resultados de los tres métodos restantes, lo cual implica que la encuesta a los estudiantes era el método que hace no sean estadísticamente iguales en la evaluación anterior. Este resultado refuerza la suposición descrita con anterioridad en cuanto a que los estudiantes de este cuarto año no poseen una valoración adecuada de su aprendizaje creativo de la Ingeniería del Software. Por tanto, es uno de los aspectos necesarios a corregir en su formación como profesionales informáticos. Esta autovaloración inadecuada no les permite analizar sus errores y tratar de corregirlos, se observa que generalmente cuestionan los resultados obtenidos sin aportar argumentos que evidencien un análisis profundo. El 60% de los estudiantes argumentan la eficiencia y la eficacia de la metodología propuesta desde una posición autoritaria sin aceptar críticas del resto de los colegas o argumentos en contra de sus opiniones. Un 70% no logran la síntesis personal de las metodologías y artefactos, en ocasiones interrumpen al profesor o a otro compañero para pedir anotaciones sobre lo que fue dicho por el profesor.

Posteriormente a este análisis, se calcula el aprendizaje creativo de la Ingeniería del Software utilizando la ecuación 7 debido a que en todos los instrumentos de los diferentes métodos se tuvo en cuenta los indicadores declarados. En la Tabla 6 se muestran los resultados.

Tabla 6. Evaluación del aprendizaje creativo utilizando n métodos.

Estudiante	Evaluación Cuantitativa	Evaluación Cualitativa
E1	0.7807125	Excelente
E2	0.614663	Bien
E3	0.608869	Bien
E4	0.2182	Mal
E5	0.350456	Regular
E6	0.361144	Regular
E7	0.327513	Regular
E8	0.437806	Regular
E9	0.362856	Regular
E10	0.379038	Regular
E11	0.376644	Regular

Fuente: Elaboración propia.

En la Tabla 6 se muestra que solamente tres estudiantes de 11 muestran un aprendizaje creativo para un 27.27% de evaluados con bien y excelente. Para mayor fortaleza del rechazo a la hipótesis se toma uno cualquiera de los métodos aplicados que son estadísticamente iguales y se obtiene el mismo resultado, pues en los tres métodos solamente hubo tres estudiantes evaluados convenientemente. De esta manera se rechaza la hipótesis que afirma que el 80% de los estudiantes de la carrera poseen un aprendizaje creativo de la ingeniería del *software*, evaluado como bien o excelente por ser estos contenidos el centro de su formación profesional y estar en año terminal.

Fortalezas

Los contenidos de Ingeniería del Software para la Ingeniería Informática poseen potencialidades para propiciar el desarrollo del aprendizaje creativo.

- Los profesores comprometidos con la formación de los estudiantes de la Ingeniería Informática.
- Los estudiantes reconocen la importancia de esta asignatura para su formación profesional.
- Acceso a planes de clases de otras instituciones que permitan profundizar en temas de Ingeniería del Software.

Debilidades

- La mayoría de los estudiantes (70%) poseen una preparación limitada para la búsqueda de nuevas soluciones y la creación de nuevas ideas. Ello se evidencia en el pobre valor que le dan a la generación de nuevas ideas y la búsqueda de problemas en organizaciones que necesitan de informatizarse. Tampoco proveen soluciones ajustadas al contexto de la organización o del cliente, lo que lleva a propuestas formales, muy parecidas a las estudiadas en clases.

- La mayoría de los estudiantes (80%) aprenden los contenidos para saber lo que se les da, pero no para generar nuevas soluciones. Los estudiantes no están interesados en la búsqueda de nuevas metodologías o herramientas ni tecnologías que permitan hacer más eficiente o eficaces los procesos de informatización. No proveen a las organizaciones de soluciones que permitan obtener un valor agregado de la introducción de tecnologías. Solamente estudian para obtener el aprobado y poder pasar de grado.

Discusión

Una acción por sí sola no ayuda al logro de los objetivos de los procesos educativos, pues deben existir un conjunto de ellas integradas entre sí para lograr la intencionalidad educativa que se desee. Para definir esta integración es necesario abordar los sistemas definidos como el "...conjunto de componentes interrelacionados entre sí, desde el punto de vista estático y dinámico, cuyo funcionamiento está dirigido al logro de determinados objetivos" (Daraio *et al.*, 2022). Sin embargo, hablar de conjunto no implica las interacciones que se establecen entre los componentes y la estructura que conforman en sus relaciones, tampoco se expresa sus relaciones con el entorno, y ello es crucial para conocer el carácter de ese sistema. Para los autores, los sistemas -y en especial los educativos- son sistemas complejos pues están en constante interacción con un ambiente cada vez más cambiante, en intercambio de información que va a modificar la estructura y sus relaciones, al mismo tiempo que plantea un conjunto de componentes y sus relaciones que lo hacen único e irrepetible.

Varios autores clasifican los sistemas de actividades en docentes, metodológicos o de otra índole, teniendo en cuenta el tipo de actividades que predomina, así como la intencionalidad educativa que prevalece. Sin embargo, los sistemas de actividades para el aprendizaje creativo deben poseer actividades de índole pedagógico al integrarse a las organizaciones de desarrollo de *software*, al tener en cuenta diversos componentes y categorías didácticas o metodológicas cuando involucran las acciones que realiza el profesor para lograr el objetivo con los estudiantes. Se reconoce que un sistema de actividades debe ser una configuración de actividades de diversa índole para el logro de una intencionalidad educativa que se expresa en su objetivo. La estructura utilizada en la presente investigación propone como resultado final el desarrollo del aprendizaje creativo, a través de los contenidos de Ingeniería del Software, en la carrera Ingeniería Informática de la Universidad de Matanzas.

El objetivo general del sistema de actividades didácticas es: Estructurar un sistema de actividades que contribuya al desarrollo del aprendizaje creativo de los contenidos de Ingeniería del Software, en la carrera Ingeniería Informática de la Universidad de Matanzas.

El sistema de actividades didácticas está dirigido a realizar acciones encaminadas a incorporar el aprendizaje creativo, a través de los contenidos de Ingeniería del Software en la carrera Ingeniería Informática.

Las actividades didácticas tienen la siguiente estructura:

- Tipo de actividad
- Temática
- Objetivo
- Acciones del profesor
- Acciones del estudiante
- Resultados esperados
- Orientaciones metodológicas

La estructura del sistema de actividades se muestra a continuación en la Figura 3.

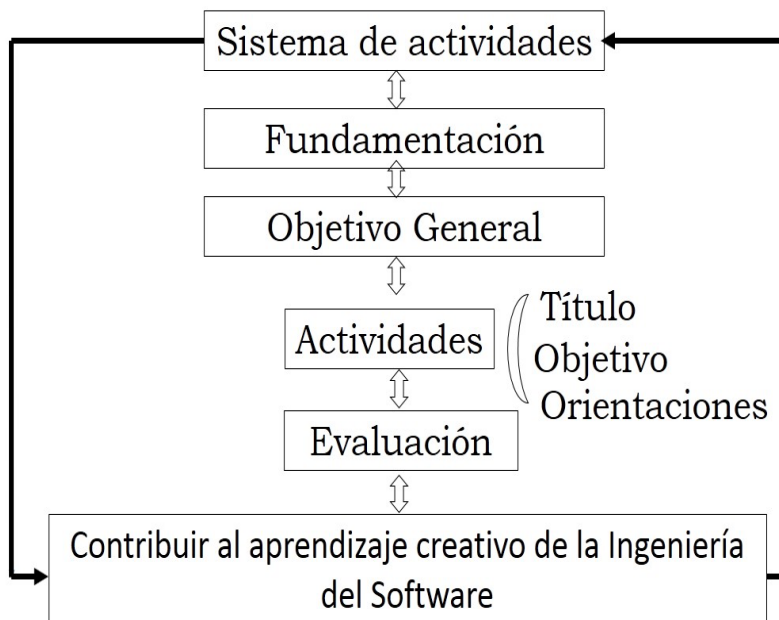


Figura 3. Representación esquemática del sistema de actividades.
Fuente: Elaboración propia.

Se planificaron 12 actividades que poseen la siguiente estructura: tipo de actividad, temática, objetivo, acciones del profesor, acciones del estudiante, resultados esperados y orientaciones metodológicas. Se desglosará solamente la primera actividad para mostrar su contenido por razones de espacio.

Una vez presentado el resumen de cada actividad en la Tabla 7 es necesario mostrar una actividad completa.

Tabla 7. Síntesis de las actividades propuestas.

Número	Temática	Tipo de actividad	Frecuencia	Objetivos	Formas de la Evaluación
1	La Ingeniería del Software como disciplina científica de la Ingeniería Informática	Preparación posterior a la Conferencia	Cada vez que se imparte nuevo contenido	<ul style="list-style-type: none"> · Caracterizar el concepto de ingeniería del software a partir de la necesidad de su surgimiento que les permita la formación de la disciplina en el desarrollo del <i>software</i>. · Crear una biblioteca digital personalizada a partir de una búsqueda en internet (WOS, Scopus y Google Académico) que les permita confrontar los contenidos estudiados con la búsqueda realizada. 	Coevaluación, pues los estudiantes se evalúan la elaboración de la biblioteca teniendo en cuenta la completitud y alcance de la biblioteca digital.
2	Ranking de las tecnologías, metodologías y framework más utilizados y divulgados, a nivel internacional.	Auto preparación.	Dos veces en el semestre	Elaborar un ranking de las tecnologías, metodologías y framework más utilizados y divulgados, a nivel internacional, mediante la búsqueda de información en Internet.	Coevaluación. Entrega de proyecto a otros estudiantes en el que aparezca el ranking

Fuente: Elaboración propia.

Tabla 7. Síntesis de las actividades propuestas (continuación).

Número	Temática	Tipo de actividad	Frecuencia	Objetivos	Formas de la Evaluación
3	Estado actual y tendencias de la Ingeniería del Software nivel internacional.	Autoestudio	Una vez en el semestre	Caracterizar el estado actual y tendencias de la Ingeniería del Software a nivel internacional, mediante la búsqueda de información en sitios de Internet.	Coevaluación. Los informes de estado actual se revisan por otros estudiantes con los criterios brindados por el profesor.
4	Participación en concursos de conocimientos relacionados con los contenidos de Ingeniería del Software.	Auto preparación	Tres veces en el semestre	Participar en concursos de conocimientos, mediante el estudio y análisis de los contenidos de Ingeniería del Software.	La evaluación que se realice en el concurso
5	Datos con framework (marco de trabajo) divulgados.	Auto preparación.	Una vez, cuando seleccionan el framework para el proyecto.	Elaborar de un fichero de datos con framework (marco de trabajo) divulgados, que contenga una ficha de cada uno de los framework referenciados	Presentación de fichero de datos
6	Problemas en el diseño de software relacionados con los contenidos de Ingeniería del Software	Torbellino de ideas para la detección de problemas relacionados con la Ingeniería del Software.	Tres veces en el semestre para evaluar los diseños del proyecto.	Detectar problemáticas sobre diseño de software en los proyectos de software desarrollados en la universidad teniendo en cuenta los contenidos de Ingeniería del Software.	Metaevaluación pues se evalúan las vías y las formas en que se detectan las problemáticas.
7	Estado de la Ingeniería del Software en la Universidad de Matanzas	Torbellino de ideas para la detección de aplicaciones de la Ingeniería del Software	Dos veces en el semestre, para valorar si existen problemáticas que no son detectadas.	Describir las aplicaciones de Ingeniería del Software en la Universidad de Matanzas a partir de los procesos de desarrollo que esta realiza.	Coevaluación pues todos evalúan las ideas de todos.
8	Modelos de desarrollo de software.	Conferencia	Una vez en el semestre, cuando levantan requisitos	Caracterizar los distintos modelos de desarrollo de software que les permita la selección del más adecuado para la empresa donde trabajarán.	Coevaluación pues varios estudiantes hacen función de evaluadores.
9	Implementación de la seguridad en la Ingeniería del Software.	Taller sobre seguridad	Una vez, al levantar requisitos	Valorar la importancia que tiene la seguridad y privacidad de la información desde las etapas tempranas del proceso de desarrollo de software.	La evaluación de las ideas las hace el profesor.
10	Ingeniería del Software en la informatización del país.	Análisis documental sobre informatización del país	Permanentemente	Proponer soluciones tecnológicas que teniendo en cuenta los contenidos de Ingeniería del Software en las organizaciones donde han realizado su práctica laboral.	Metaevaluación pues se evaluarán las vías y procedimientos que les permiten llegar a la propuesta.
11	Soluciones con Ingeniería del Software.	Auto preparación	Permanentemente pues van confrontando los proyectos con las soluciones propuestas.	Resolver problemas reales haciendo uso de las propuestas de la Ingeniería del Software.	
12	Ingeniería del Software en el campus universitario.	Torbellino de ideas sobre la introducción de la Ingeniería del Software en los procesos universitarios	Dos veces al semestre cuando se elaboran las propuestas de proyectos de informatización universitarios.	Elaborar propuestas para insertar arquitecturas, modelos, procesos de la Ingeniería del Software en el campus universitario.	Coevaluación pues todos evalúan las ideas de todos.

Fuente: Elaboración propia.

Actividad #1 (Con carácter permanente después de cada conferencia)

Tipo de actividad: Preparación posterior a la Conferencia.

Temática de la Conferencia #1: La Ingeniería del Software: necesidad de su surgimiento y desarrollo.

Frecuencia de la Actividad: Cada vez que se imparte nuevo contenido.

Objetivos

- Caracterizar el concepto de ingeniería del software a partir de la necesidad de su surgimiento que les permita la formación de la disciplina en el desarrollo del software.
- Crear una biblioteca digital personalizada a partir de una búsqueda en internet (WOS, Scopus y Google Académico) que les permita confrontar los contenidos estudiados con la búsqueda realizada.

Acciones del profesor

Orienta a los estudiantes a:

- Profundizar en el estudio de los contenidos de Ingeniería del Software.
- Realizar búsqueda de libros en Google Académico, Scopus y la web de la ciencia (WOS, por sus siglas en inglés), relacionados con los conceptos fundamentales abordados en la clase.
- Descargar toda la información posible para su estudio y análisis.
- Propicia el debate de la actividad usando diversos formatos y espacios de aprendizaje que les permita a los estudiantes diversificar las vías y formatos del debate.

Acciones del estudiante

- Profundiza en los contenidos relacionados con la temática estudiados en clase
- Realiza búsqueda de libros en Google Académico, Scopus y la web de la ciencia (WOS), relacionados con los conceptos fundamentales abordados en la clase utilizando los medios digitales que posea o sean garantizados por la institución. En Cuba todos los centros universitarios garantizan una conexión de 15 Mb de internet y una descarga de 10 Gb mensual totalmente gratis a los estudiantes. Pone a disposición de ellos los laboratorios de computadoras las 24 horas del día con personal técnico que les presta ayuda en caso de que tengan algún problema.
- Descarga todos los documentos asociados a los conceptos fundamentales de la conferencia que se haya impartido.
- Compila la información para su estudio y análisis.
- Se introducen los documentos descargados en la biblioteca digital personalizada utilizando la opción de importar carpeta u otra que el estudiante conozca.
- Se escriben los metadatos de cada una de ella para su posterior citación en trabajos o su procesamiento ulterior.

Resultados esperados

- El estudiante desarrolla la biblioteca digital según lo orientado por el profesor.

Orientaciones metodológicas

- Se orienta al profesor para utilizar esta actividad en las clases donde se desarrollan los contenidos de Ingeniería del Software.

Con los profesores se realizaron otras actividades:

Actividad #2. Título: Introducción a las pruebas de *software*. Objetivo: profundizar en los métodos y modelos de desarrollo orientados a pruebas que les permita la corrección los defectos que tengan los sistemas desarrollados en la universidad.

Actividad #3. Título: Enfoques de enseñanza de la informática. Objetivo: determinar los principales enfoques de enseñanza de Ingeniería del Software.

Actividad #4. Título: El enfoque de proyectos y el enfoque problémico. Objetivo: Elaborar un sistema de clases orientado al aprendizaje de los contenidos de Ingeniería del Software por parte de los estudiantes.

Actividad #5. Título: Aprendizaje creativo de Ingeniería del Software. Objetivo: caracterizar el aprendizaje creativo de Ingeniería del Software con sus dimensiones, indicadores y criterios de medida.

Actividad #6. Título: La integración de enfoques como una vía para el desarrollo del aprendizaje creativo. Objetivo: Diseñar un sistema de clases que permita el desarrollo del aprendizaje creativo de Ingeniería del Software.

En el caso de los profesores, las actividades de la 1 a la 5 están orientadas a crear las bases para revertir la situación detectada en el diagnóstico, mientras que las dos últimas actividades están dedicadas al aprendizaje creativo de la Ingeniería del Software y la vía para su desarrollo. Se ha demostrado en varios estudios (Castro-Martín & Silva-Lorente, 2022; Houghton *et al.*, 2022; Purwanto *et al.*, 2022) que la introducción del aprendizaje basado en problemas y la enseñanza por problemas es una vía efectiva para el desarrollo de la crítica, así como para la generación y producción de nuevas ideas.

Las restantes actividades están encaminadas al desarrollo de los criterios de medida que se estructuran para el aprendizaje creativo. La actividad 6 va adentrando a los estudiantes en la detección de problemáticas propias a resolver utilizando los conocimientos de la Ingeniería del Software. Varios estudios (Beda *et al.*, 2020; Stolz *et al.*, 2022) coinciden en que una expresión de la creatividad es la búsqueda de problemas que se corresponde con el indicador 3.3. La séptima actividad va desarrollando los indicadores 3.1 y 3.2 al mismo tiempo que otros indicadores, como el 2.3, 2.2 y el 2.1, a partir de la detección de las problemáticas que realiza y las soluciones que propone para corregirlas. De la 7 a la 9 le va permitir a los estudiantes transgredir más allá de lo dado y cuestionarse todos los contenidos estudiados en su plan de estudios, lo que llevaría a un desarrollo de la dimensión 2 con todos sus indicadores. Las restantes actividades están orientadas fundamentalmente a la dimensión 3, pues va a la búsqueda de soluciones propuestas en el proceso de desarrollo de los sistemas a nivel de país y a nivel de universidad.

Validación del sistema de actividades didácticas propuesto para el desarrollo del aprendizaje creativo en la carrera Ingeniería Informática

Los especialistas consultados incluyen a: un profesor (20%), Doctor en Ciencias, con la categoría docente de Profesor Titular; tres profesores (60%), Máster en Educación, con categoría docente de Asistente y un profesor, (20%), Máster en Educación, con categoría docente de Auxiliar. El 100% de los participantes son docentes que se desempeñan como profesores en la carrera Ingeniería Informática. La experiencia promedio de los cinco especialistas es de nueve años, y todos ellos gozan de reconocido prestigio entre sus compañeros y en el contexto laboral de la Universidad de Matanzas.

Los especialistas fueron informados sobre la importancia de este trabajo y la necesidad de recopilar sus experiencias y opiniones, a fin de perfeccionar el sistema de actividades y llevarlo a su culminación mediante un criterio colegiado de personas dispuestas a mejorar la labor que, de manera cotidiana, se lleva a cabo con los estudiantes de la carrera Ingeniería Informática de la Universidad de Matanzas. Los resultados obtenidos se detallan a continuación.

En cuanto a la primera actividad, referida a la caracterización del estado actual y tendencias del desarrollo de la Ingeniería del Software a nivel internacional, mediante la búsqueda de información en la plataforma Moodle y otros sitios de Internet, los cinco especialistas (el 100%) marcaron el indicador 4 (de acuerdo), expresando que la misma favorece la actualización de los contenidos de Ingeniería del Software no solo a nivel nacional, sino a nivel internacional. Estimaron, además, que este tipo de actividad es muy necesaria y que muchas veces no se le presta la atención requerida.

Referente a la segunda actividad, la cual trata sobre la compilación de textos de Ingeniería del Software, a partir de la construcción de una biblioteca digital con libros relacionados con la temática, el 100% de los especialistas consultados marcaron el indicador 5 (muy de acuerdo), a lo cual se le prestó especial atención a lo largo de todo su desempeño científico.

En una tercera actividad, que se refiere a la construcción de un ranking de las tecnologías, metodologías y *framework* (marco de trabajo) más utilizados y divulgados a nivel internacional, relacionados con la Ingeniería del Software, el 100% de los especialistas consultados estuvieron de acuerdo con ella.

La cuarta actividad, referente a la preparación para la participación en concursos de conocimientos relacionados con los contenidos de Ingeniería del Software, dos de los especialistas consultados (40%) marcaron el indicador 4 (de acuerdo) y tres de ellos (60%) marcaron el indicador 5 (muy de acuerdo).

En la quinta actividad, donde se hace referencia a la construcción de un fichero de datos con *framework* (marco de trabajo) divulgados, que contenga una ficha de cada uno de los *framework*, cuatro de los especialistas consultados (80%) marcaron el indicador 5 (muy de acuerdo) y uno de los especialistas (20%) marcó el indicador 3 (ni de acuerdo, ni en desacuerdo).

Finalmente, en la actividad 6, la cual trata sobre la resolución de problemas relacionados con la Ingeniería del Software, el 100% de los especialistas entrevistados marcaron el indicador 5 (muy de acuerdo), pues consideran la actividad de gran importancia para las asignaturas que trabajan con los contenidos de Ingeniería del Software.

En general, las sugerencias ofrecidas por los especialistas, una vez llevada a cabo la revisión del sistema de actividades elaborado, permitieron ganar en claridad y tomar en consideración tales sugerencias para llevar a cabo el perfeccionamiento de la labor investigativa desarrollada.

Conclusiones

El estudio de los fundamentos teóricos permitió determinar la definición de aprendizaje creativo de la Ingeniería del Software, sus dimensiones, indicadores y criterios de medida. Las métricas obtenidas permitieron cuantificar una variable compleja, como el aprendizaje creativo, y pueden ser aplicadas también en entornos empresariales, ajustando la variable a ese entorno.

Los resultados del diagnóstico inicial evidenciaron la necesidad de trabajar con mayor rigor, sistematicidad y profundidad en el fomento del aprendizaje creativo en los estudiantes de la carrera Ingeniería Informática en la Universidad de Matanzas. La evaluación de la variable lleva a buscar soluciones que permitan llevarlo a su estado deseado, teniendo en cuenta que las asignaturas se imparten en años terminales.

El sistema de actividades didácticas propuesto se corresponde con las debilidades y fortalezas identificadas a partir del diagnóstico realizado y se orienta a fomentar el aprendizaje creativo de la Ingeniería del Software en la carrera Ingeniería Informática. El sistema está orientado a estudiantes y profesores. Para los primeros se realizan 12 actividades que van complejizando el aprendizaje de la Ingeniería del Software hasta la formación de los tres procesos que lo caracterizan.

La consulta a especialistas permitió corroborar el valor del resultado elaborado como producto de la investigación y que puede resolver la problemática planteada. Los especialistas votaron favorablemente y se tuvieron en cuenta sus recomendaciones para mejorar el sistema de actividades.

Conflicto de interés

Los autores declaran que no existen conflictos de interés.

Referencias

- Alvarez, D. M. P. (2024). Formación de competencias informacionales e investigativas en el profesional de las ciencias informáticas. *Revista Cubana de Ciencias Informáticas*, 18(1).
- Beda, Z., Smith, S. M., & Orr, J. (2020). Review Creativity on demand – Hacking into creative problem solving. *NeuroImage*, 216, 1-13. <https://doi.org/10.1016/j.neuroimage.2020.116867>
- Castro-Martín, B., & Silva-Lorente, I. (2022). Aprendizaje Basado en Problemas (ABP) e interdisciplinariedad como ejes para el desarrollo profesional. *Aula de Encuentro*, 24(1), 77-101. <https://doi.org/10.17561/ae.v24n1.6773>
- Ciriello, R. F., Richter, A., & Mathiassen, L. (2024). Emergence of creativity in IS development teams: A socio-technical systems perspective. *International Journal of Information Management*, 74, 1-14. <https://doi.org/https://doi.org/10.1016/j.ijinfomgt.2023.102698>
- Daraio, C., Leo, S. D., & Scannapieco, M. (2022). Accounting for quality in data integration systems: a completeness-aware integration approach. *Scientometrics*, 127, 1465-1490. <https://doi.org/10.1007/s11192-022-04266-0>
- González-Hernández, W. (2015). *Apuntes sobre Didáctica de la Informática*. Editorial Universitaria.
- González-Hernández, W. (2016). La implementación de procesos de informatización en organizaciones como competencia en la formación del profesional informático. *e-Ciencias de la Información*, 6(2), 3-19. <https://doi.org/10.15517/eci.v6i2.25276>
- González-Hernández, W., & Coloma-Carrasco, Á. L. (2018). Estado actual de la competencia modelar en la formación del profesional informático de la Universidad de Matanzas, Cuba. *Paideia*(60), 105-124.
- González-Hernández, W. (2019). La competencia depurar sistemas informáticos en la Carrera de Ingeniería Informática de la Universidad de Matanzas. *ReiDoCrea*, 8(11), 119-132.
- González-Hernández, W. (2022). Modelo de aprendizaje desarrollador de la informática para la secundaria básica cubana. *Revista Espiga*, 21(43), 108-125.
- González-Hernández, W., Petersson Roldán, M., & García Moreno, M. (2022a). *El aprendizaje creativo de la informática: definición de una variable* XXIII Evento Internacional de la Enseñanza de la Matemática, la Computación y la Estadística, Matanzas-Cuba.
- González-Hernández, W., Petersson Roldán, M., & García Moreno, M. (2022b). *Métrica para evaluar el aprendizaje creativo de la informática* XXIII Evento Internacional de la Enseñanza de la Matemática, la Computación y la Estadística, Matanzas-Cuba.
- González, J. A. T., Velazco, A. E., Lantigua, M. E., & Álvarez, C. A. S. (2021). *Evaluation of an accreditation variable for university institutions using 2 tuple linguistic representation model*. Conferencia Científica Internacional Uciencia.

- Guo, Y., Tao, J., Yang, F., Chen, C., & Reniers, G. (2022). An evaluation of the information literacy of safety professionals. *Safety Science*, 151, 105734. <https://doi.org/https://doi.org/10.1016/j.ssci.2022.105734>
- Hernández-Sampieri, R., Fernández Collado, C., & Baptista Lucio, M. d. P. (2014). *Metodología de la investigación*. McGRAW-HILL / Interamericana Editores.
- Herrera-Masó, J. R., Ricardo, J. L. C., Rangel, M. Á. G., Ramos, M. I. C., & González, Y. T. (2022). El método de consulta a expertos en tres niveles de validación. *Revista Habanera de Ciencias Médicas*, 21(1), 1-11.
- Houghton, T., Lavicza, Z., Weinhandl, R., Diego-Mantecón, J. M., & Fitri Rahmadi, I. (2022). Hothousing: utilising industry collaborative problem-solving practices for steam in schools. *Journal of Technology and Science Education*, 12(1), 274-289. <https://doi.org/10.3926/jotse.1324>
- Korhonen, T., Salo, L., Reinius, H., Malander, S., Tiippana, N., Laakso, N., Lavonen, J., & Hakkarainen, K. (2024). Creating Transformative Research-Practice Partnership in Collaboration with School, City, and University Actors. *education sciences*, 14(4).
- Kruchten, P., Fraser, S., & Coallier, F. (2019). *Agile Processes in Software Engineering and Extreme Programming*. Springer.
- Mellor, A. (2023). *Test-Driven Development with Java. Create higher-quality software by writing tests first with SOLID and hexagonal architecture*. Packt Publishing.
- Méndez-Fernández, D., & Passoth, J.-H. (2019). Empirical software engineering: From discipline to interdiscipline. *Journal of Systems and Software*, 148, 170-179. <https://doi.org/10.1016/j.jss.2018.11.019>
- Mi, W., Li, Y., Wen, M., & Chen, Y. (2022). Using active learning selection approach for cross-project software defect prediction. *Connection Science*, 34(1), 1482-1499. <https://doi.org/10.1080/09540091.2022.2077913>
- Monat, J., Gannon, T., & Amissah, M. (2022). The Case for Systems Thinking in Undergraduate Engineering Education. *ijEP*, 12(3), 50-88. <https://doi.org/10.3991/ijep.v12i3.25035>
- Moreno-García, M. C. (2019). El aprendizaje creativo en la matemática, su contribución a la formación del ingeniero industrial. *Atenas*, 2(46), 47-63.
- Opdahl, J. (2024). Embedded information literacy interventions in an experiential business learning capstone. *The Journal of Academic Librarianship*, 50(2), 102860. <https://doi.org/https://doi.org/10.1016/j.acalib.2024.102860>
- Purwanto, A., Rahmawati, Y., Rahmayanti, N., Mardiah, A., & Amalia, R. (2022). Socio-critical and problem-oriented approach in environmental issues for students' critical thinking skills development in chemistry learning. *Journal of Technology and Science Education*, 12(1), 50-67. <https://doi.org/10.3926/jotse.1341>
- Saaty, R. W. (1987). The analytic hierarchy process—what it is and how it is used. *Mathematical modelling*, 9(3-5), 161-176.
- Soto-Duran, D. E., Acosta Gómez, J. A., & Vargas Agudelo, F. A. (2010). *Aplicación del proceso personal de software en profesional informático*. 8th Latin American and Caribbean Conference for Engineering and Technology WE, Arequipa-Perú.
- Stephenson, L. (2023). Collective creativity and wellbeing dispositions: Children's perceptions of learning through drama. *Thinking Skills and Creativity*, 47, 1-15. <https://doi.org/10.1016/j.tsc.2022.101188>
- Stolz, R. C., Tucker Blackmon, A., Engerman, K., Tonge, L., & McKayle, C. A. (2022). Poised for creativity: Benefits of exposing undergraduate students to creative problem-solving to moderate change in creative self-efficacy and academic achievement. *Journal of Creativity*, 32, 1-7. <https://doi.org/10.1016/j.yjoc.2022.100024>
- Torres-Oliveira, C., & Mitjans-Martínez, A. (2020). Expresiones de la subjetividad social en el aula y creatividad en el aprendizaje: un estudio de caso. *Alternativas cubanas en Psicología* 8(23), 126-144.
- Uzair-Khan, M., Sherin, S., Zohaib-Iqbal, M., & Zahid, R. (2019). Landscaping systematic mapping studies in software engineering: A tertiary study. *The Journal of Systems and Software*, 149, 396-436. <https://doi.org/10.1016/j.jss.2018.12.018>
- Waki de-Oliveira, A., de-Carvalho, T. O., de-Oliveira, M. G., Abreu, I. C. C., & Nakano, T. d.-C. (2023). The perception of creativity among Brazilian students and teachers. *Psicologia Escolar e Educacional*, 27, 1-10. <https://doi.org/10.1590/2175-35392023-241073-T>

Wang, L. (2022). Influence of Teacher Behaviors on Student Activities in Information-Based Classroom Teaching. *ijET*, 17(02), 19-31. <https://doi.org/10.3991/ijet.v17i02.28271>

Williams, A., & Rainer, A. (2019). Do software engineering practitioners cite software testing research in their online articles? A larger scale replication EASE '19, Copenhagen-Denmark.

Zweeris, K., Tigelaar, E. H., & Janssen, F. J. J. M. (2023). Studying curriculum orientations in teachers' everyday practices: A goal systems approach. *Teaching and Teacher Education*, 122, 103969. <https://doi.org/https://doi.org/10.1016/j.tate.2022.103969>