



REVISTA MULTIDISCIPLINAR EPISTEMOLOGÍA DE LAS CIENCIAS

Volumen 2, Número 4
Octubre-Diciembre 2025

Edición Trimestral

CROSSREF PREFIX DOI: 10.71112

ISSN: 3061-7812, www.omniscens.com

Revista Multidisciplinar Epistemología de las Ciencias

Volumen 2, Número 4
octubre-diciembre 2025

Publicación trimestral
Hecho en México

La Revista Multidisciplinar Epistemología de las Ciencias acepta publicaciones de cualquier área del conocimiento, promoviendo una plataforma inclusiva para la discusión y análisis de los fundamentos epistemológicos en diversas disciplinas. La revista invita a investigadores y profesionales de campos como las ciencias naturales, sociales, humanísticas, tecnológicas y de la salud, entre otros, a contribuir con artículos originales, revisiones, estudios de caso y ensayos teóricos. Con su enfoque multidisciplinario, busca fomentar el diálogo y la reflexión sobre las metodologías, teorías y prácticas que sustentan el avance del conocimiento científico en todas las áreas.

Contacto principal: admin@omniscens.com

Las opiniones expresadas por los autores no necesariamente reflejan la postura del editor de la publicación

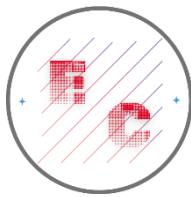
Se autoriza la reproducción total o parcial del contenido de la publicación sin previa autorización de la Revista Multidisciplinar Epistemología de las Ciencias siempre y cuando se cite la fuente completa y su dirección electrónica.



9773061781003

Cintillo legal

Revista Multidisciplinar Epistemología de las Ciencias Vol. 2, Núm. 4, octubre-diciembre 2025, es una publicación trimestral editada por el Dr. Moises Ake Uc, C. 51 #221 x 16B , Las Brisas, Mérida, Yucatán, México, C.P. 97144 , Tel. 9993556027, Web: <https://www.omniscens.com>, admin@omniscens.com, Editor responsable: Dr. Moises Ake Uc. Reserva de Derechos al Uso Exclusivo No. 04-2024-121717181700-102, ISSN: 3061-7812, ambos otorgados por el Instituto Nacional del Derecho de Autor (INDAUTOR). Responsable de la última actualización de este número, Dr. Moises Ake Uc, fecha de última modificación, 1 octubre 2025.



Revista Multidisciplinar Epistemología de las Ciencias

Volumen 2, Número 4, 2025, octubre-diciembre

DOI: <https://doi.org/10.71112/1zhy1358>

**PERCEPCIONES, MOTIVACIÓN Y ACEPTACIÓN DE LAS PLATAFORMAS DE
PROGRAMACIÓN COMPETITIVA EN ESTUDIANTES DE INGENIERÍA**

**PERCEPTIONS, MOTIVATION, AND ACCEPTANCE OF COMPETITIVE
PROGRAMMING PLATFORMS IN ENGINEERING EDUCATION**

Dora Ivette Rivero Caraveo

Claudia Anglés Barrios

María Eugenia Sánchez Leal

Alma Patricia Gallegos Borunda

Irving Bruno López Santos

México

Percepciones, motivación y aceptación de las plataformas de programación competitiva en estudiantes de ingeniería

Perceptions, Motivation, and Acceptance of Competitive Programming Platforms in Engineering Education

Dora Ivette Rivero Caraveo¹

dora.rc@cdjuarez.tecnm.mx

<https://orcid.org/0000-0003-3428-5246>

Tecnológico Nacional de México, Instituto

Tecnológico de Ciudad Juárez

México

María Eugenia Sánchez Leal

maria.sl@cdjuarez.tecnm.mx

<https://orcid.org/0000-0003-1410-2019>

Tecnológico Nacional de México, Instituto

Tecnológico de Ciudad Juárez

México

Irving Bruno López Santos

irving.ls@cdjuarez.tecnm.mx

<https://orcid.org/0009-0003-9947-3288>

Tecnológico Nacional de México, Instituto

Tecnológico de Ciudad Juárez

México

Claudia Anglés Barrios

claudia.ab@cdjuarez.tecnm.mx

<https://orcid.org/0000-0002-8935-0968>

Tecnológico Nacional de México, Instituto

Tecnológico de Ciudad Juárez

México

Alma Patricia Gallegos Borunda

alma.gb@cdjuarez.tecnm.mx

<https://orcid.org/0009-0003-0046-8199>

Tecnológico Nacional de México, Instituto

Tecnológico de Ciudad Juárez

México

¹ Correspondencia: dora.rc@cdjuarez.tecnm.mx

RESUMEN

El propósito del presente estudio es analizar las percepciones, motivaciones y nivel de aceptación de las plataformas de programación competitiva entre estudiantes de programas de ingenierías dentro del Instituto Tecnológico de Ciudad Juárez, quienes en su mayoría pertenecen a la Generación Z. Mediante un enfoque cuantitativo descriptivo, se aplicó un cuestionario a 77 estudiantes para evaluar su conocimiento, experiencia previa, barreras percibidas y disposición para integrar estas plataformas en cursos de programación. Los resultados muestran un bajo nivel de familiaridad y práctica, pero una alta utilidad percibida y una actitud favorable hacia su adopción, especialmente mediante talleres y actividades guiadas. Las principales barreras fueron la falta de exposición inicial y la percepción de dificultad. Se concluye que la programación competitiva posee un alto potencial educativo para esta generación, favoreciendo el aprendizaje activo y el pensamiento computacional cuando se implementa de manera gradual y acompañada.

Palabras clave: programación competitiva; motivación académica; aprendizaje activo, desempeño estudiantil, aprendizaje de programación.

ABSTRACT

This study examines students' perceptions, motivation, and acceptance of competitive programming platforms among students at the Instituto Tecnológico de Ciudad Juárez, most of whom belong to Generation Z. Using a descriptive quantitative approach, a survey was administered to 77 students to evaluate their knowledge, prior experience, perceived barriers, and intention to adopt these platforms within programming courses. The results show a low level of familiarity and practice, but a high perceived usefulness and a favorable attitude toward adoption, particularly through workshops and guided activities. The main barriers identified were limited initial exposure and the perception of difficulty. The study concludes that competitive

programming holds strong educational potential for this generation, supporting active learning and computational thinking when implemented gradually and with appropriate guidance.

Keywords: competitive programming; academic motivation; active learning; student performance; programming learning.

Recibido: 11 de diciembre 2025 | Aceptado: 25 de diciembre 2025 | Publicado: 26 de diciembre 2025

INTRODUCCIÓN

Dada la importancia que representa el sistema educativo universitario resulta relevante profundizar en el análisis del rendimiento académico de los estudiantes. Uno de los mayores retos que enfrentan las instituciones de educación superior (IES) es la eficiencia terminal, la cual es un asunto de interés para toda sociedad. Como mencionan Herrera y otros autores (2024): “medir la eficiencia terminal en las instituciones de educación superior, es fundamental; pero no de la forma como históricamente se viene adoptando. El indicador cuantitativo, es un punto de partida que establece una línea base, para luego medir cualitativa y cuantitativamente los planes de mejora institucional y su impacto en la mejora de la ET”, indica la existencia de múltiples factores internos y externos que al interrelacionarse dan pauta a que sea multifactorial.

Los autores mencionados también indican que estos factores pueden dividirse en dos bloques: los que se relacionan con las características no-académicas de los estudiantes (nivel económico, entorno socioeconómico y sociocultural, modalidad de colegio, nivel educativo del padre y de la madre, etc.), y las que están relacionadas al área académica como las calificaciones en exámenes de admisión, de desempeño, por mencionar los principales. El estudio presentado aborda el segundo bloque descrito, en particular en el aprendizaje de la programación de estudiantes de ingeniería de nivel superior.

Aunado a lo anterior, es muy importante contemplar la acelerada transformación que sufre actualmente el entorno industrial y de servicios, debido a que las IES forman futuros profesionistas. En el contexto mencionado, la educación superior enfrenta el reto de preparar a los futuros ingenieros para entornos laborales marcados por la interacción humano-tecnología, donde la automatización inteligente coexiste con el pensamiento crítico, la colaboración interdisciplinaria y la toma de decisiones sostenible (Rollnik-Sadowska et al., 2024). En América Latina, este desafío se acentúa debido a la necesidad simultánea de reducir brechas digitales estructurales y fortalecer ecosistemas educativos orientados a las competencias que el entorno actual demanda. Diversos países de Latinoamérica, incluido México, han replanteado la formación en ingeniería mediante modelos de colaboración universidad-empresa, esquemas de educación dual, proyectos aplicados e incorporación de tecnologías emergentes en la enseñanza (Arregi et al., 2025; Peláez-Sánchez et al., 2024). Este trabajo se centra en la aplicación de tecnologías emergentes en la enseñanza de la programación, cuya complejidad y desafíos han sido ampliamente documentados en la literatura.

Dado que estas transformaciones repercuten directamente en la manera en que se enseñan y desarrollan las competencias tecnológicas, es pertinente examinar cómo se manifiestan en ámbitos formativos específicos, particularmente en el aprendizaje de la programación. La literatura reciente documenta que los estudiantes suelen enfrentar barreras cognitivas asociadas a la abstracción algorítmica, dificultades afectivas como la ansiedad ante tareas complejas y retos contextuales derivados de modelos de enseñanza centrados en la exposición teórica y la práctica aislada (Eckerdal, 2023; Ngadengon, 2025). Estas dificultades se intensifican cuando no existen mecanismos de retroalimentación inmediata o estrategias pedagógicas que fomenten el aprendizaje activo, la resolución de problemas y la autonomía del estudiante.

A este contexto, se suma el perfil de los estudiantes universitarios actuales, mayoritariamente pertenecientes a la Generación Z, caracterizada por estilos de aprendizaje fuertemente visuales, preferencia por la práctica sobre la teoría, baja tolerancia a actividades extensas sin estímulo, y necesidad de retroalimentación casi inmediata (Aviña Camacho, 2023; Saxena & Mishra, 2021; Wajdi et al., 2024). Si bien esta generación destaca por su alfabetización digital, también enfrenta brechas en habilidades interpersonales, gestión del tiempo y comunicación escrita, elementos críticos en la formación de profesionales de ingeniería (Deloitte, 2024). Estas características demandan estrategias didácticas que combinen dinamismo, interactividad, autonomía y acompañamiento, favoreciendo aprendizajes más significativos en contextos reales.

En ese tenor, la programación competitiva ha surgido como una alternativa formativa de alto valor en cursos introductorios y avanzados de Ciencias de la Computación. Estudios recientes destacan que su incorporación puede fortalecer habilidades como el pensamiento lógico, la creatividad algorítmica, la eficiencia en la resolución de problemas y la capacidad de trabajar bajo presión (Friss De Kereki & Luna, 2022; Lora Patiño et al., 2021; Moreno-Cadavid et al., 2025; Nair, 2020).

Asimismo, en un estudio realizado por Yuen (2023) con más de 400 estudiantes, se encontró que una mayoría sostiene que los concursos les ayudan a aumentar la confianza al resolver problemas y afinar sus habilidades de programación. En un desarrollo más reciente (Luo, 2025) propone un diseño curricular basado en concursos como evaluación formativa, integrando presión de tiempo, colaboración y observaciones, donde los estudiantes manifiestan notables mejoras en el razonamiento algorítmico, capacidad de depuración, así como mayor rapidez para abordar problemas nuevos.

Sin embargo, otros estudios advierten que, sin un adecuado andamiaje pedagógico, la programación competitiva puede generar frustración, ansiedad o desigualdades entre

estudiantes con distintos niveles de experiencia, destacando la importancia del diseño instruccional y de la retroalimentación formativa (Messer, Brown & Kölling, 2023). Como antecedente directo de esta investigación, un estudio previo realizado en el Instituto Tecnológico de Ciudad Juárez analizó la percepción docente sobre el uso de la programación competitiva como estrategia didáctica, identificando una actitud positiva hacia su implementación, así como barreras relacionadas con la falta de capacitación, tiempo y recursos (Rivero Caraveo et al., 2025). Dicho estudio reveló el potencial de estas plataformas para apoyar el proceso de enseñanza, pero también mostró la necesidad de conocer la perspectiva del estudiantado para comprender de manera integral los factores que influyen en su adopción en el aula.

Con el fin de complementar el estudio mencionado, surge la necesidad de explorar la percepción estudiantil respecto al conocimiento, utilidad, dificultad percibida, motivación e intención de uso de las plataformas de programación competitiva. Esto resulta especialmente pertinente en el contexto actual, donde la adopción de tecnologías educativas debe alinearse tanto a los estilos de aprendizaje de la Generación Z como a los fundamentos teóricos de aceptación tecnológica. En este estudio, la interpretación de los datos se apoya en el Modelo de Aceptación Tecnológica (TAM), planteado por Davis (1989) utilizado previamente en el análisis docente, pero ahora adaptado para comprender cómo la utilidad percibida, la facilidad de uso percibida, la actitud y la intención de adopción se manifiestan desde la perspectiva del estudiante.

Por lo anterior, el objetivo de este trabajo es analizar las percepciones, motivaciones y nivel de aceptación de las plataformas de programación competitiva en estudiantes de ingeniería del ITCJ, identificando tanto las oportunidades como las barreras que enfrentan en su integración como estrategia didáctica. Los resultados permiten comprender cómo estas herramientas pueden contribuir al aprendizaje activo, al fortalecimiento del pensamiento

computacional y a la mejora de la experiencia educativa, ofreciendo evidencia empírica que complementa y amplía los hallazgos previos desde la mirada docente. Asimismo, este estudio aporta insumos relevantes para el diseño de intervenciones pedagógicas basadas en retos y evaluaciones automatizadas, acordes a las necesidades de la Generación Z y a las exigencias de la formación actual en ingeniería.

Diversos estudios han documentado que la programación es una de las áreas más complejas para estudiantes de ingeniería por su carácter abstracto, la necesidad de dominar estructuras formales y el pensamiento computacional requerido para formular soluciones algorítmicas (Agbo et al., 2019; Figueiredo & García-Peñalvo, 2019; Medeiros et al., 2019; Santos et al., 2020). Los autores mencionados han identificado dificultades persistentes como la falta de habilidades previas, la desmotivación, la ansiedad ante errores, la confusión entre sintaxis y lógica y la desconexión entre teoría y práctica.

Otra barrera para que los estudiantes aprendan la programación a nivel superior es la diversidad de perfiles de los estudiantes: algunos ingresan con experiencia previa en codificación, mientras que otros carecen de nociones básicas de algoritmia, lo que genera brechas de desempeño y dificultades para avanzar a un ritmo uniforme (Brito & Gonçalves, 2019; Sinza-Díaz et al., 2023). Debido a lo anterior y a que el enfoque tradicional de enseñanza es mayormente expositivo y centrada en el docente, se incrementa la frustración, el abandono temprano y la percepción negativa de la asignatura. Por lo anteriormente expuesto, surge la necesidad de incorporar metodologías activas y recursos tecnológicos que fomenten la autonomía, el aprendizaje significativo, la retroalimentación continua expedita y la resolución de problemas en escenarios reales (Gavrilas & Kotsis, 2025).

Las características de los estudiantes universitarios de la actualidad, quienes en su mayoría pertenecen a la Generación Z, destacan por su familiaridad con entornos digitales, pero también enfrentan desafíos como menor tolerancia a la frustración, dispersión atencional y

dificultad para sostener procesos analíticos prolongados (Jawad & Tout, 2021; O'Farrell & Weaver, 2024; Shinta et al., 2024). Por lo anteriormente mencionado, las estrategias formativas que integren retos progresivos, práctica deliberada, elementos de gamificación, trabajo colaborativo y retroalimentación inmediata resultan especialmente adecuadas para sus características cognitivas y socioemocionales. La programación competitiva es pertinente con los estilos mencionados y las características de la Generación Z, pues combina dificultad incremental, desafíos breves, métricas instantáneas y sentido de logro.

La programación competitiva se ha consolidado en la última década como una metodología de enseñanza que combina resolución de problemas, evaluación automatizada, competencia académica y práctica deliberada (Brito & Gonçalves, 2019; Di Mascio & Luigi, 2018; Raman et al., 2018). Su estructura permite al estudiantado enfrentar problemas reales o semirrealistas en tiempo limitado, promoviendo principalmente: desarrollo del pensamiento computacional, creatividad algorítmica y eficiencia en el diseño de soluciones, habilidades de trabajo bajo presión y autorregulación incremento en la motivación y el compromiso estudiantil (Agbo et al., 2019; Lora Patiño et al., 2021; Morales Ortiz et al., 2024; Nair, 2020; Sinza-Díaz et al., 2023).

En la tabla 1, se presenta una relación entre las ventajas que promueve la programación competitiva con respecto a las preferencias y características de la Generación Z.

Por lo tanto, la programación competitiva, cuando se diseña con cuidado (niveles de desafío adecuados, soporte, evaluación formativa, espacio para la retroalimentación), puede actuar como un puente pedagógico que conecta los resultados de aprendizaje positivos e integrales: apoya no solamente las habilidades duras de programación y algoritmos, sino también competencias cognitivas, metacognitivas y socio-afectivas. En resumen, la programación competitiva es una estrategia formativa de gran potencial, aunque su efectividad depende fuertemente del contexto: perfil del estudiante, estructura del curso, recursos de evaluación, y del acompañamiento del docente.

Tabla 1

Relación de las ventajas de la programación competitiva con las características de la Generación Z

Ventajas o características de la programación competitiva	Preferencias y características de la Generación Z	Fuentes
Desarrollo del pensamiento computacional mediante retos incrementales y resolución algorítmica.	Preferencia por actividades prácticas, visuales y orientadas a la acción; aprendizaje basado en problemas reales.	Agbo et al. (2019); Sinza-Díaz et al. (2023); O'Farrell & Weaver (2024).
Creatividad algorítmica y eficiencia en el diseño de soluciones mediante experimentación y mejora continua.	Gusto por entornos que permitan exploración autónoma, retroalimentación inmediata y experimentación digital.	Lora Patiño et al. (2021); Nair (2020); Shinta et al. (2024)
Trabajo bajo presión, autorregulación y resiliencia derivadas de competencia y límites temporales.	Capacidad de respuesta rápida, preferencia por tareas breves y retadoras; alta afinidad con dinámicas gamificadas.	Morales Ortiz et al. (2024); Deloitte (2024)
Incremento de la motivación y compromiso por medio de rankings, insignias, logros y retos progresivos.	Alta motivación por gamificación, recompensas inmediatas, métricas visibles y progresión por niveles.	Morales Ortiz et al. (2024); Cipriano & Alves (2024)
Evaluación automatizada y retroalimentación inmediata , reduciendo tiempos de espera y permitiendo mejora iterativa.	Necesidad de retroalimentación instantánea para mantener atención y seguridad en el proceso de aprendizaje.	Dueñas Orozco & Ávalos Piñón (2024); Friss de Kereki & Luna (2022); Shinta et al. (2024)
Resolución de problemas en escenarios reales o competitivos , con relevancia académica y profesional.	Preferencia por aprendizajes aplicados y contextualizados; búsqueda de sentido práctico y empleabilidad.	Raman et al. (2018); Brito & Gonçalves (2019); Aviña Camacho (2023).
Pensamiento sistémico, adaptabilidad y trabajo colaborativo , alineadas con el entorno laboral actual	Preferencia por las dinámicas colaborativas, entornos flexibles y herramientas tecnológicas para el trabajo en equipo.	Arregi et al. (2025); Cheng & Huang (2025); O'Farrell & Weaver (2024).

A pesar de que la generación que actualmente cursa la educación superior es nativa digitalmente, cuando se pretenden implementar tecnologías es importante tomar como referencia un modelo de aceptación. Un modelo ampliamente documentado y aceptado en la literatura es el Modelo de Aceptación Tecnológica (conocido como TAM, por sus siglas en inglés), el cual fue propuesto por Davis (1989). En el modelo mencionado, se propone que la adopción de una herramienta tecnológica depende principalmente de dos percepciones:

1. Utilidad percibida (UP): Se refiere al grado en el que la plataforma tecnológica mejora su aprendizaje, desempeño o comprensión.
2. Facilidad de uso percibida (FUP): Se refiere a qué tan sencillo es interactuar con la herramienta, aprender a usarla o resolver ejercicios en ella.

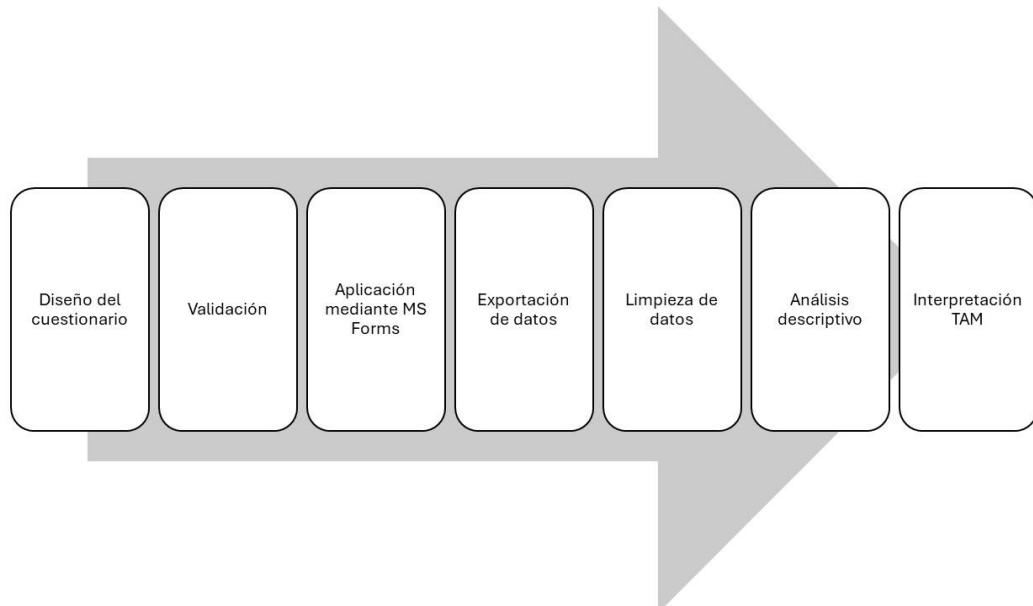
A estos factores se suma la actitud hacia el uso y la intención de adopción (IU), determinantes en el comportamiento final.

En resumen, la revisión del contexto educativo, las características de la Generación Z, la evidencia previa sobre programación competitiva y el enfoque teórico del modelo TAM permiten fundamentar la necesidad de un análisis empírico desde la perspectiva de los estudiantes. Con base en esta justificación, el presente estudio se diseñó para caracterizar el conocimiento, la percepción, la utilidad percibida, la dificultad, la motivación y la intención de uso de plataformas de programación competitiva entre estudiantes de ingeniería. A continuación, se describe el enfoque metodológico empleado, el diseño del instrumento, los procedimientos de recolección y análisis de datos, así como las consideraciones éticas que guiaron la investigación. Posteriormente se presentan los resultados obtenidos y se discuten los hallazgos principales de la investigación presentada.

METODOLOGÍA

El estudio presentado se desarrolló con un enfoque cuantitativo, de tipo descriptivo y exploratorio, complementado con elementos cualitativos procedentes de cuatro preguntas abiertas. El objetivo principal fue analizar el conocimiento, percepción, aceptación, motivación e intención de uso de herramientas de programación competitiva por parte de estudiantes de ingeniería del Instituto Tecnológico de Ciudad Juárez (ITCJ), con énfasis en su potencial como estrategia didáctica en cursos de programación. El diseño fue no experimental, transversal y observacional, dado que los datos se recopilaron en un único momento temporal, sin manipulación de variables.

Las fases metodológicas de este estudio se muestran en la figura 1. En primer lugar, se llevó a cabo el diseño del cuestionario, en el cual se definieron las dimensiones, categorías e indicadores alineados al Modelo TAM. Posteriormente se realizó una validación del instrumento mediante revisión experta para asegurar coherencia y pertinencia de los reactivos. Una vez validado, el instrumento se aplicó mediante Microsoft Forms a estudiantes de diversas carreras que cursaban asignaturas de programación. Concluida la aplicación, las respuestas fueron exportadas a Excel para su procesamiento y sometidas a una etapa de limpieza de datos, en la que se verificó la integridad y consistencia de la información. Posteriormente, se efectuó el análisis descriptivo, que incluyó frecuencias, porcentajes y cruces de variables. Finalmente, los resultados fueron interpretados con base en los constructos del Modelo de Aceptación Tecnológica (TAM), permitiendo contextualizar las percepciones y actitudes del estudiantado respecto al uso de plataformas de programación competitiva.

Figura 1*Fases del procedimiento metodológico del estudio*

Con el propósito de identificar el nivel de conocimiento de la programación competitiva, se diseñó un cuestionario estructurado utilizando Microsoft Forms institucional, con el fin de aplicarlo a estudiantes de diferentes carreras donde se imparten asignaturas de programación. El instrumento diseñado se compone de 19 preguntas distribuidas en siete categorías generales:

1. Datos generales (carrera, semestre).
2. Conocimiento sobre programación competitiva (familiaridad, fuentes de conocimiento).
3. Experiencia previa (plataformas utilizadas, participación en concursos, tiempo practicado).
4. Lenguajes de programación empleados en prácticas relacionadas.
5. Motivaciones y habilidades desarrolladas.
6. Barreras o dificultades percibidas.

7. Actitud e intención de uso, incluyendo utilidad percibida y formatos preferidos de integración

El diseño del instrumento se fundamentó explícitamente en el Modelo de Aceptación Tecnológica (TAM) de Davis (1989). Las preguntas del cuestionario se organizaron para evaluar:

- Utilidad percibida (UP): beneficios académicos y profesionales.
- Facilidad de uso percibida (FUP): dificultad, familiaridad, barreras.
- Actitud hacia el uso (A): motivación, nivel de interés.
- Intención de uso (IU): deseo de incorporarla en cursos regulares.

En la tabla 2, se describe cada una de las categorías del instrumento mencionado, el contenido evaluado y su correspondencia con el modelo TAM. Cabe mencionar que los datos generales es la única categoría que no se relaciona con el modelo mencionado, pero se consideró para conocer la población estudiada.

Tabla 2

Categorías del instrumento y su correspondencia con el modelo TAM

Categoría	Contenido evaluado	Correspondencia TAM
Datos generales	Carrera y semestre	—
Conocimiento	Concepto de programación competitiva y fuentes	FUP
Experiencia previa	Plataformas, participación, tiempo, lenguajes	FUP
Motivación y habilidades	Razones de uso, habilidades desarrolladas	A
Barreras	Dificultades, impedimentos	FUP negativa
Intención de uso	Preferencias de integración, utilidad percibida	IU / UP
Preguntas abiertas	Plataformas conocidas, dificultades, beneficios	A / FUP / UP

El cuestionario se diseñó y revisó con apoyo de docentes expertos para asegurar pertinencia y coherencia. Posteriormente, se distribuyó mediante Microsoft Forms utilizando cuentas institucionales del ITCJ. La participación fue voluntaria, anónima y sin recolección de datos personales sensibles. Las respuestas obtenidas fueron exportadas a Excel para su procesamiento, aplicando:

- Estadística descriptiva (frecuencias y porcentajes).
- Análisis cruzado por carrera, semestre, motivación y nivel de conocimiento.
- Clasificación inductiva de respuestas abiertas.

Las representaciones gráficas generadas automáticamente por Forms fueron utilizadas como material base para construir figuras y tablas interpretativas que se presentan en las secciones de resultados y discusión.

En cuanto a consideraciones éticas del estudio elaborado, se les informó a los participantes sobre el propósito académico del estudio y se garantizó la confidencialidad de sus respuestas. No se recopilaron datos sensibles ni información que permitiera su identificación.

RESULTADOS

La muestra estuvo conformada de 77 estudiantes: 49 de la Licenciatura de Ingeniería en Sistemas Computacionales, 5 de Ingeniería en Tecnologías de Información y Comunicación, 13 de Ingeniería Industrial, 8 de Ingeniería Mecatrónica, 1 de Ingeniería electromecánica y 1 estudiante de Maestría.

De los estudiantes encuestados, 24 (31%) habían escuchado hablar de la programación competitiva, mientras que 53 (69%) desconocían sobre el tema. Estos resultados se muestran en la figura 2.

En cuanto al semestre cursado, las respuestas se concentraron principalmente entre quinto y sexto semestre, con presencia minoritaria de niveles inferiores. El muestreo fue no

probabilístico por conveniencia, dada la disponibilidad de estudiantes en las asignaturas donde se distribuyó el cuestionario.

Figura 2

Conocimiento previo sobre programación competitiva en estudiantes



En la figura 3 se presentan los medios a través de los cuales los estudiantes conocieron la programación competitiva. Del total de 24 estudiantes, 10 (42%) indicó haberse enterado por medio de un profesor, 7 (29%) mencionaron que fue por amigos o compañeros, 6 (25%) señalaron que fue a través de internet o redes sociales, y 1 (4%) escuchó hablar de programación competitiva en una conferencia.

Figura 3

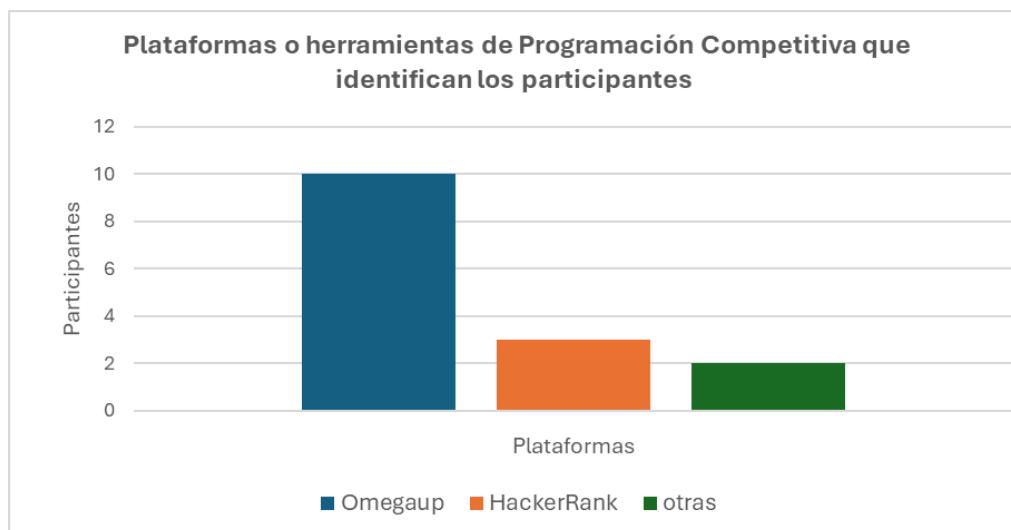
Medio por el cual conocieron la programación competitiva



De los 24 estudiantes que habían escuchado hablar sobre el tema solo 16 mencionaron conocer alguna plataforma o herramienta de programación competitiva (por ejemplo, Codeforces, HackerRank, OmegaUp, entre otras. Lo anterior implica que el 22% de los encuestados ha utilizado al menos una plataforma de programación competitiva. OmegaUp fue la plataforma más reconocida entre los encuestados que las han utilizado, con un 62%, HackerRank, obtuvo un 18%, y el 20% restante se distribuyó entre otras plataformas. Se descartaron 6 respuestas debido a que no correspondían a plataformas o herramientas de programación competitiva. Estos resultados se muestran en la figura 4.

Figura 4

Plataformas de programación competitiva que los participantes identifican



En la figura 5 se muestra el tipo de participación que han tenido los estudiantes en actividades relacionada con programación competitiva. De los datos obtenidos se encontró que 9 de los estudiantes han participado en concursos de la localidad o escolares, 5 en concursos regionales, 5 en plataformas en línea, 2 en Hackatón y 2 en concursos nacionales. Cabe mencionar que, un mismo estudiante puede haber participado en más de una de estas actividades. La mayoría de los estudiantes reportó tener 3 meses participando en este tipo de actividades, mientras que un grupo reducido mencionó que tiene entre 3 meses y un año, lo

que indica que en general los participantes tienen un tiempo relativamente corto practicando este tipo de programación.

Figura 5

Tipo de actividades de programación competitiva donde han participado



Los motivos por lo que los estudiantes utilizan la programación competitiva de acuerdo con el estudio realizado los resultados son los siguientes: 29% como pasatiempo, 29% para preparación para participar en concursos, 21 % para incrementar su lógica de programación, 18 % para resolver problemas y 1 % para entrevistas. En la figura 6 se muestran los porcentajes de cada categoría.

Figura 6

Razones por las que los estudiantes utilizan plataformas de programación competitiva



En cuanto a las habilidades que se adquieren al utilizar la programación competitiva se obtuvieron los siguientes datos que se muestran en la figura 7. Habilidades de trabajar bajo presión fue la más alta con un 31%, seguida por la resolución de problemas que obtuvo un 30%, enseguida esta mejorar en la lógica y uso de estructuras con un 21% y finalmente con un 18% la habilidad para mejorar en las habilidades de programación.

Figura 7

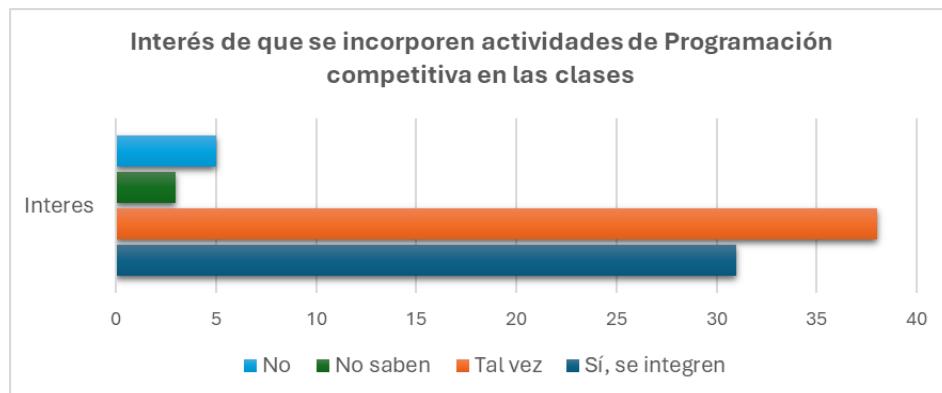
Habilidades adquiridas gracias a la programación competitiva



Los resultados revelan que los estudiantes muestran interés en que se incorporen actividades de programación competitiva en sus asignaturas. En total, 31 estudiantes expresaron que sí desean que estas actividades se integren a sus clases, 38 respondieron que tal vez, 3 no saben y solo 5 manifestaron no estar de acuerdo. Estos datos se ilustran en la

Figura 8

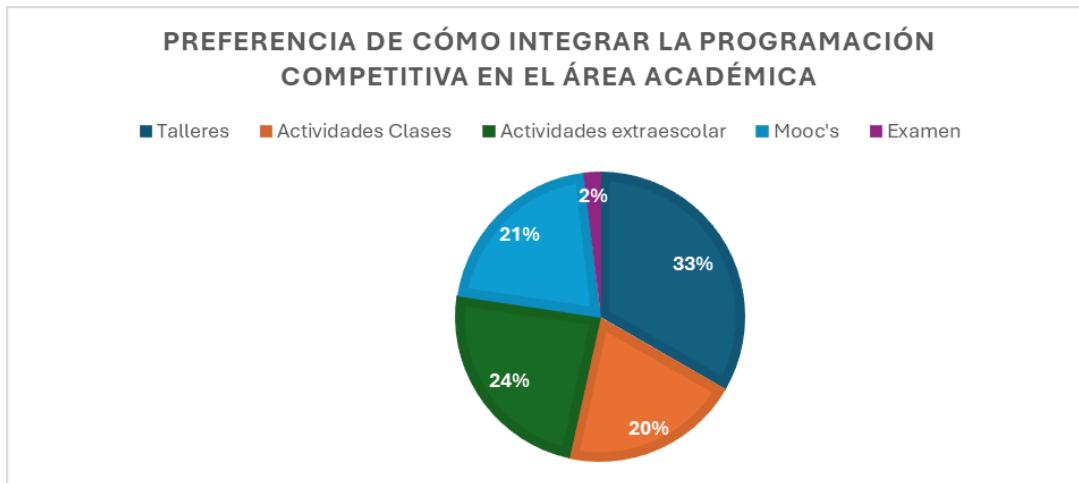
Interés de incorporar actividades de programación competitiva en las clases



En la Figura 9 se muestran las preferencias de cómo integrar la programación competitiva como parte de la educación superior, el estudio sugiere que se realice por medio de la impartición de talleres, ya que obtuvo el 33%, el siguiente con un 24% fue establecer la programación competitiva como una actividad extraescolar, seguido de Mooc que alcanzó un 21% y en actividades en clase con un 20%, finalmente con un 2 % fue considerado en la realización de exámenes.

Figura 9

Cómo integrar la programación competitiva en el ámbito académico



Sobre el hecho de si la programación competitiva puede mejorar las oportunidades de trabajo los resultados revelan que el 70% de los estudiantes opina que, si puede ayudar a obtener un mejor empleo, el 28% considera que no y el 13% restante no está seguro. Lo anteriormente mencionado se muestra en la figura 10.

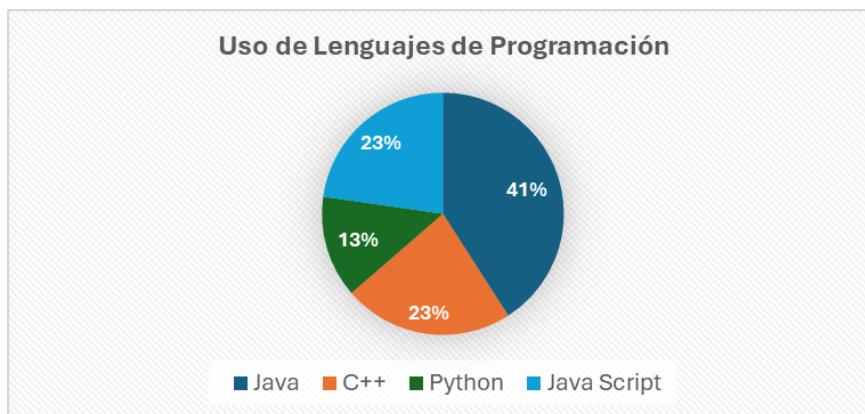
Los resultados indican que el lenguaje más utilizado por los participantes es Java con un 41 por ciento, seguido por Python y lenguaje C++ con un 23 por ciento respectivamente y java script con un 13 por ciento. En la figura 11 se muestran los resultados mencionados.

Figura 10

Percepción de la mejora de las oportunidades laborales gracias a la programación competitiva

**Figura 11**

Utilización de lenguajes de programación



En la figura 12 se muestran los motivos por los cuales los estudiantes no practican la programación competitiva encontrándose en primer lugar el hecho de desconocer sobre el tema, el segundo motivo es que consideran que es una actividad con un grado de dificultad alto, y la falta de interés que arroja el mismo porcentaje, por último, se encuentra la falta de tiempo.

Figura 12

Barreras percibidas por los alumnos para adentrarse en la programación competitiva



En resumen, los resultados permiten identificar el nivel de conocimiento, las experiencias previas, las motivaciones, las habilidades percibidas, las barreras y la intención de uso de la programación competitiva entre los estudiantes participantes. Aunque se observa una participación limitada y un desconocimiento generalizado del tema, también se identifican actitudes positivas hacia su integración en la formación profesional, así como un reconocimiento claro de los beneficios percibidos en términos de desarrollo de habilidades cognitivas y laborales.

Con base en los hallazgos presentados, en la siguiente sección se discute cómo estas percepciones se relacionan con los fundamentos teóricos presentados en la revisión de literatura presentada referente a las características de la Generación Z y con los elementos del modelo TAM, con el fin de comprender el potencial de la programación competitiva y las condiciones necesarias para su implementación como estrategia educativa.

DISCUSIÓN

Desde el punto de vista del modelo TAM, los resultados obtenidos permiten interpretar el grado de aceptación de la programación competitiva por parte de los estudiantes y las características pedagógicas asociadas a la Generación Z. Primeramente, la facilidad de uso

percibida (FUP) se ve afectada por el limitado conocimiento y experiencia inicial: solo el 31% había escuchado el término y apenas el 22% había utilizado alguna plataforma. Estos indicadores demuestran que, en la población de estudio, representan una barrera a considerarse, coincidiendo con estudios que destacan la necesidad de introducir estas herramientas gradualmente y con acompañamiento y seguimiento continuo por parte de los docentes. La percepción de dificultad, reportada por un porcentaje significativo, refuerza esta interpretación y explica por qué muchos estudiantes aún no participan de manera activa.

En contraste, la utilidad percibida (UP) muestra valores significativos. La mayoría de los estudiantes considera que la programación competitiva puede apoyar el desarrollo de habilidades esenciales como la resolución de problemas, la lógica algorítmica, el trabajo bajo presión y la rapidez para procesar información, lo cual coincide con la literatura especializada. Asimismo, el 70% reconoce su potencial en la empleabilidad, lo que sugiere una valoración positiva respecto a su pertinencia profesional.

En cuanto a la actitud hacia el uso (A) de la programación competitiva, se obtuvieron resultados favorables en la intención de integrar actividades de programación competitiva a las asignaturas: la mayoría de los alumnos se muestran interesados o dispuestos a participar en talleres, actividades extracurriculares o cursos formales. Estos resultados son congruentes con características propias de la Generación Z reportadas en la literatura, en donde se ha encontrado que esta cohorte valora el aprendizaje activo, la retroalimentación inmediata y los entornos gamificados. La preferencia por talleres también sugiere que los estudiantes requieren un espacio guiado y estructurado antes de interactuar autónomamente con estas plataformas.

Finalmente, en cuanto a la intención de uso (IU), se encontraron motivadores positivos, que este tipo de actividades pueden ser un pasatiempo, servir como preparación para concursos y mejorar de la lógica. En cuanto a las barreras percibidas por los participantes del estudio se encontró que el desconocimiento, la percepción de dificultad y la falta de tiempo

pueden ser obstáculos para adentrarse en esta disciplina. Estas barreras coinciden con los hallazgos del estudio docente previo, donde la falta de capacitación y familiaridad también apareció como limitante. Así, ambas perspectivas confirman que la implementación exitosa depende de procesos institucionales de acompañamiento, formación docente y estrategias pedagógicas progresivas.

En resumen, los resultados sugieren que la programación competitiva es valorada positivamente por los estudiantes y que su potencial educativo es alto; sin embargo, su adopción depende de reducir barreras iniciales y diseñar experiencias de aprendizaje coherentes con sus estilos cognitivos y motivacionales.

CONCLUSIONES

El presente estudio permite concluir que, aunque el conocimiento y la experiencia previa con la programación competitiva entre los estudiantes es limitada, existe una percepción positiva respecto a su valor formativo y profesional. Los alumnos que participaron en el estudio manifestaron que las plataformas de programación competitiva son herramientas capaces de fortalecer habilidades cognitivas esenciales, mejorar la empleabilidad y promover el aprendizaje activo, especialmente cuando se implementan mediante talleres o actividades guiadas.

Los resultados también evidencian barreras asociadas al desconocimiento y la percepción de dificultad, lo que sugiere la necesidad de incorporar estrategias de introducción gradual, acompañamiento docente y actividades formativas diseñadas con niveles progresivos de desafío. Se recomienda que las instituciones de educación superior consideren integrar la programación competitiva como complemento en cursos de programación, actividades extracurriculares o proyectos de formación integral, especialmente en contextos donde la Generación Z demanda experiencias interactivas, dinámicas y significativas.

La contribución de este trabajo de investigación es el entendimiento de los factores que favorecen la adopción de este tipo de herramientas y proporcionan evidencia para diseñar estrategias pedagógicas alineadas a las necesidades actuales de la enseñanza de la programación básica en la ingeniería.

Declaración de conflicto de interés

El autor y coautores declaran no tener ningún conflicto de interés relacionado con esta investigación.

Declaración de contribución a la autoría

Dora Ivette Rivero Caraveo: conceptualización, metodología, investigación, análisis formal, administración del proyecto, supervisión, recursos, curación de datos, validación, visualización, software, redacción del borrador original, revisión y edición de la redacción.

Claudia Anglés Barrios: curación de datos, análisis formal, investigación, visualización, software, metodología, validación, redacción y edición de la redacción.

María Eugenia Sánchez Leal: investigación, recursos, revisión de la literatura, validación, redacción y edición de la redacción.

Alma Patricia Gallegos Borunda: investigación, recursos, revisión de la literatura, validación, redacción y edición de la redacción.

Irving Bruno López Santos: metodología, investigación, recursos y software

Declaración de uso de inteligencia artificial

Los autores declaran que utilizaron herramientas de inteligencia artificial exclusivamente como apoyo en tareas auxiliares del proceso de investigación, principalmente para identificar palabras clave y términos de búsqueda, así como la revisión de la claridad y coherencia en algunos fragmentos de la redacción. Las herramientas de IA Generativa no fueron empleadas para generar contenido original del artículo. Esta herramienta no sustituye de ninguna manera la tarea o proceso intelectual. Después de rigurosas revisiones con diferentes herramientas en

la que se comprobó que no existe plagio como constan en las evidencias, los autores manifiestan y reconocen que este trabajo fue producto de un trabajo intelectual propio, que no ha sido escrito ni publicado en ninguna plataforma electrónica o de IA.

REFERENCIAS

- Abdelhamid, S., & Aly, M. (n.d.). Generation Z and Programming Education: Bridging the Gap with Innovative Tech-Driven Learning Approaches.
- Agbo, F. J., Oyelere, S. S., Suhonen, J., & Adewumi, S. (2019, November 21). A systematic review of computational thinking approach for programming education in higher education institutions. ACM International Conference Proceeding Series.
- <https://doi.org/10.1145/3364510.3364521>
- Arregi, A., Eguren, J. A., Retegi, J., Ibarra, D., & Igartua, J. I. (2025). Instructional Design as a Key Factor for Industry 5.0 Engineering Education. *Procedia Computer Science*, 253, 985–994. <https://doi.org/10.1016/j.procs.2025.01.160>
- Aviña Camacho, I. (2023). La generación Z en el ámbito educativo superior; sus retos. Dilemas Contemporáneos: Educación, Política y Valores.
- Brito, M., & Gonçalves, C. (2019). Codeflex: A Web-based Platform for Competitive Programming. 411.
- Davis, F. D., Bagozzi, R. P., & Warshaw, P. R. (1989). Technology acceptance model. *J Manag Sci*, 35, 982–1003.
- Deloitte. (2024). 2024 Gen Z and Millennial Survey: Living and working with purpose in a transforming world.
- Eckerdal, A., Berglund, A., & Thuné, M. (2024). Learning programming practice and programming theory in the computer laboratory. *European Journal of Engineering Education*, 49(2), 330–347. <https://doi.org/10.1080/03043797.2023.2294953>

Figueiredo, J., & García-Péalvo, F. J. (2019). Teaching and learning strategies of programming for university courses. ACM International Conference Proceeding Series, 1020–1027.

<https://doi.org/10.1145/3362789.3362926>

Friss De Kereki, I., & Luna, C. (2022). Reporte de Experiencia: Alcances del Concurso de Programación. 1.

Gavrilas, L., & Kotsis, K. T. (2025). The evolution of STEM education and the transition to STEAM/STREAM. Aquademia, 9(1), ep25002.

<https://doi.org/10.29333/aquademia/16313>

Herrera, L. C., Montenegro, M., Torres-Lista, V., Martínez, L. A., & López, V. (2024). Eficiencia terminal en la educación superior: Hacia un nuevo enfoque. Revista Educación Superior y Sociedad (ESS), 36(1), 244–260. <https://doi.org/10.54674/ess.v36i1.694>

Jawad, H. M., & Tout, S. (2021). Gamifying computer science education for z generation. In Information (Switzerland) (Vol. 12, Issue 11). MDPI.

<https://doi.org/10.3390/info12110453>

Leso, B. H., & Cortimiglia, M. N. (2022). The influence of user involvement in information system adoption: an extension of TAM. Cognition, Technology and Work, 24(2), 215–231.

<https://doi.org/10.1007/s10111-021-00685-w>

Lora Patiño, G. A., Suaza Jiménez, J. H., Rodríguez Marín, P. A., & Taborda Blandón, G. (2021). Programación competitiva como estrategia didáctica para potenciar el aprendizaje colaborativo en estudiantes de ingeniería. Revista Ibérica de Sistemas e Tecnologias de Informação, 539–556.

Luo, Z., & Dickey, E. (n.d.). Evaluating Performance Consistency in Competitive Programming: Educational Implications and Contest Design Insights.

<https://nerc.itmo.ru/information/selection-rules.html>

- Medeiros, R. P., Ramalho, G. L., & Falcao, T. P. (2019). A Systematic Literature Review on Teaching and Learning Introductory Programming in Higher Education. *IEEE Transactions on Education*, 62(2), 77–90. <https://doi.org/10.1109/TE.2018.2864133>
- Messer, M., Brown, N. C. C., Kölling, M., & Shi, M. (2024). Automated Grading and Feedback Tools for Programming Education: A Systematic Review. *ACM Transactions on Computing Education*, 24(1). <https://doi.org/10.1145/3636515>
- Moreno-Cadavid, J., Clara Gómez-Álvarez, M., & Pridrahita-Ospina, A. A. (2025). Programación competitiva como herramienta didáctica en cursos de ciencias de la computación. In *Actas de las Jenui* (Vol. 10). <https://www.domjudge.org/>
- Nair, P. R. (2020). Increasing employability of Indian engineering graduates through experiential learning programs and competitive programming: Case study. *Procedia Computer Science*, 172, 831–837. <https://doi.org/10.1016/j.procs.2020.05.119>
- Ngadengon, Z., Subramaniam, T. S., Yasak, Z., Ideris, N. A., & Mohd Ramly, Z. (2025). Evaluating the Difficulties in Programming Learning: Insights from Polytechnic Students. *International Journal of Academic Research in Progressive Education and Development*, 14(1). <https://doi.org/10.6007/ijarped/v14-i1/24518>
- O'farrell, J., & Weaver, C. (2024). Technology and Transformation: Redefining Higher Education for Generations Z and Alpha. <https://ssrn.com/abstract=4833043>
- Pelaez-Sanchez, I. C., Glasserman-Morales, L. D., & Rocha-Feregrino, G. (2024). Exploring digital competencies in higher education: design and validation of instruments for the era of Industry 5.0. In *Frontiers in Education* (Vol. 9). Frontiers Media SA. <https://doi.org/10.3389/feduc.2024.1415800>
- Raveica, I. C., Olaru, I., Hergheliegiu, E., Tampu, N. C., Radu, M. C., Chirita, B. A., Schnakovszky, C., & Ciubotariu, V. A. (2024). The Impact of Digitalization on Industrial Engineering Students' Training from the Perspective of Their Insertion in the Labor

Market in a Sustainable Economy: A Students' Opinions Survey. *Sustainability*

(Switzerland), 16(17). <https://doi.org/10.3390/su16177499>

Rivero Caraveo, D. I., Anglés Barrios, C., Sánchez Leal, M. E., & Gallegos Borunda, A. P.

(2025). Percepciones y Experiencia Docente en el Uso de Herramientas TIC de Programación Competitiva en la Educación Superior. *Ciencia Latina Revista Científica Multidisciplinaria*, 9(5), 9173–9196. https://doi.org/10.37811/cl_rcm.v9i5.20247

Rollnik-Sadowska, E., Bartkutė-Norkūnienė, V., & Grabińska, V. (2024). Digital Competencies in EU Countries – Adaptability to the “The Future Of Work” Paradigm. *Scientific Papers of Silesian University of Technology. Organization and Management Series*, 2024(201), 303–317. <https://doi.org/10.29119/1641-3466.2024.201.23>

Santos, S. C., Tedesco, P. A., Borba, M., & Brito, M. (2020). Innovative approaches in teaching programming: A systematic literature review. *CSEDU 2020 - Proceedings of the 12th International Conference on Computer Supported Education*, 1, 205–214.

<https://doi.org/10.5220/0009190502050214>

Saxena, M., & Mishra, D. K. (2021). Gamification and gen Z in higher education: A systematic review of literature. In *International Journal of Information and Communication Technology Education* (Vol. 17, Issue 4). IGI Global.

<https://doi.org/10.4018/IJICTE.20211001.0a10>

Shinta, S., Azhar, Muh., Partino, P., & Suud, F. M. (2024). The Role of Technological Media in the Character Building Education Process for Generation Z: A Literature Review. *Jurnal Penelitian Pendidikan IPA*, 10(6), 284–291. <https://doi.org/10.29303/jppipa.v10i6.7449>

Sinza-Díaz, Y. E., Oliva-Caípe, J. E., & Guerrero-Calvache, S. M. (2023). Análisis de los componentes relacionados en programación competitiva: un mapeo sistemático de literatura. *Revista Politécnica*, 19(38), 212–230.

<https://doi.org/10.33571/rpolitec.v19n38a14>

- Wajdi, M., Susanto, B., Made Sumartana, I., Agus Sutiarso, M., Hadi, W., & Negeri Bali, P. (2024). Profile of generation Z characteristics: Implications for contemporary educational approaches (Vol. 1, Issue 1). <https://ojs.ympn2.or.id/index.php/KPSBSL>
- Yuen, K. K., Liu, D. Y., & Leong, H. V. (2023). Competitive programming in computational thinking and problem solving education. *Computer Applications in Engineering Education*, 31(4), 850-866. (Luo & Dickey, n.d.)