

Ciencia Latina Revista Científica Multidisciplinar, Ciudad de México, México.
ISSN 2707-2207 / ISSN 2707-2215 (en línea), septiembre-octubre 2025,
Volumen 9, Número 5.

https://doi.org/10.37811/cl_rcm.v9i5

PERCEPCIONES Y EXPERIENCIA DOCENTE EN EL USO DE HERRAMIENTAS TIC DE PROGRAMACIÓN COMPETITIVA EN LA EDUCACIÓN SUPERIOR

**UNIVERSITY FACULTY PERCEPTIONS AND EXPERIENCES
IN THE USE OF ICT-BASED COMPETITIVE
PROGRAMMING TOOLS**

Dora Ivette Rivero Caraveo

Tecnológico Nacional de México

Claudia Anglés Barrios

Tecnológico Nacional de México

María Eugenia Sánchez Leal

Tecnológico Nacional de México

Alma Patricia Gallegos Borunda

Tecnológico Nacional de México

Percepciones y Experiencia Docente en el Uso de Herramientas TIC de Programación Competitiva en la Educación Superior

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 Campus I
Ciudad Juárez, Chihuahua México

Claudia Anglés Barrios

clauda.ab@cdjuarez.tecnm.mx

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

Tecnológico Nacional de México, Instituto
Tecnológico de Ciudad Juárez Campus I
Ciudad Juárez, Chihuahua 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 Campus I
Ciudad Juárez, Chihuahua 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 Campus I
Ciudad Juárez, Chihuahua México

RESUMEN

Este estudio analiza la percepción, experiencia y disposición de docentes del área de programación en diversas instituciones educativas de la región de Ciudad Juárez, México, principalmente del Instituto Tecnológico de Ciudad Juárez (ITCJ), con respecto al uso de herramientas TIC de programación competitiva como estrategia didáctica. Mediante un cuestionario aplicado a 27 profesores, se identificaron las plataformas más conocidas y utilizadas, el interés que tienen hacia su integración en la enseñanza de asignaturas relacionadas con programación básica, así como de algoritmos y estructuras de datos. Los resultados muestran que a pesar de que existe un reconocimiento generalizado del valor pedagógico de plataformas de programación competitiva como lo son OmegaUp y Codeforces, su adopción como herramientas didácticas es limitada. Entre los principales obstáculos destacan la falta de capacitación específica, la carga administrativa y una cultura educativa que aún no incorpora ampliamente la gamificación. A partir de estos hallazgos, se plantea la necesidad de impulsar estrategias institucionales que promuevan la formación docente y la adecuación curricular, con el fin de facilitar la incorporación efectiva de estas herramientas en la práctica educativa.

Palabras clave: programación competitiva, docencia universitaria, TIC en educación, gamificación, estrategias didácticas

¹ Autor principal.

Correspondencia: dora.rc@cdjuarez.tecnm.mx

University Faculty Perceptions and Experiences in the Use of ICT-Based Competitive Programming Tools

ABSTRACT

This study analyzes the perceptions, experiences, and dispositions of programming instructors at various educational institutions in the Ciudad Juárez region of Mexico, predominantly at the Instituto Tecnológico de Ciudad Juárez (ITCJ), regarding the use of ICT tools for competitive programming as an instructional approach. Through a questionnaire applied to 27 teachers, the most well-known platforms, the level of familiarity with their use, and the degree of interest in their integration into the teaching of subjects such as introductory programming, algorithms, and data structures were identified. The results suggest that, although there is broad recognition of the pedagogical value of competitive programming platforms such as OmegaUp and Codeforces, their adoption as instructional tools remains limited. Main barriers include a lack of specialized training, administrative workload, and an educational culture that has yet to fully embrace gamification. Based on these findings, there is a clear need to promote institutional strategies that support teacher training and curricular adaptation, with the aim of facilitating the effective integration of these tools into educational practice.

Keywords: competitive programming, university teaching, ICT in education, gamificación, instructional strategies

*Artículo recibido 09 agosto 2025
Aceptado para publicación: 13 septiembre 2025*



INTRODUCCIÓN

Enseñar programación a nivel superior sigue siendo un reto importante, especialmente en las áreas de ingeniería. A pesar de los avances en los métodos de enseñanza y la tecnología educativa, las materias relacionadas con programación siguen teniendo altos índices de reprobación y deserción. En este escenario, las herramientas TIC orientadas a la programación competitiva se presentan como una alternativa relevante a explorar, ya que no solo permiten la evaluación automática, sino que también aportan elementos de motivación, trabajo en equipo y un enfoque lúdico en el proceso de enseñanza-aprendizaje.

El principal problema que aborda esta investigación es la escasa adopción de herramientas de programación competitiva por parte del profesorado en el contexto regional de Ciudad Juárez, México, a pesar de su potencial pedagógico ampliamente reconocido en la literatura. En instituciones como el Instituto Tecnológico de Ciudad Juárez (ITCJ), estas plataformas han sido utilizadas con buenos resultados en concursos estudiantiles; sin embargo, aún no se han integrado formalmente como parte de las estrategias docentes en asignaturas clave como Fundamentos de Programación, Programación Estructurada, Programación Básica y Estructura de Datos. La poca utilización de este tipo de estrategias y plataformas en la práctica educativa, plantea la necesidad de explorar las percepciones, experiencias y barreras que enfrentan los docentes al intentar implementar dichas herramientas en el aula. Para complementar el análisis del contexto regional, se incluyó también la participación de docentes de otras instituciones, tanto de nivel superior como medio superior.

Estudiar este tema resulta relevante por tres razones fundamentales. En primer lugar, permite entender cómo los docentes enfrentan el reto de innovar en sus métodos de enseñanza ante generaciones de estudiantes con estilos de aprendizaje distintos y familiarizados desde temprana edad con las tecnologías digitales. En segundo lugar, porque aporta información sobre prácticas que todavía no están bien sistematizadas en América Latina. Por último, porque ofrece recursos valiosos para el desarrollo de estrategias de formación docente, rediseño curricular e integración tecnológica, con el objetivo de mejorar el desempeño académico en las asignaturas de programación.

La presente investigación se sustenta en un marco teórico que integra tres enfoques clave: la gamificación, el aprendizaje colaborativo y la adopción tecnológica. La gamificación, entendida como



el uso de elementos de juego en contextos no lúdicos (Andújar et al., 2020; Gonzalez-Escribano et al., 2019), ha demostrado ser una estrategia efectiva para aumentar la motivación y el compromiso del alumnado. Desde la perspectiva del aprendizaje colaborativo, se reconoce que la programación competitiva puede fomentar la interacción entre estudiantes, así como el desarrollo de habilidades como el liderazgo y la resolución conjunta de problemas (Lora Patiño et al., 2021).

Por su parte, para analizar la disposición del profesorado hacia el uso de tecnologías educativas, se recurre a modelos de adopción tecnológica, en particular al Modelo de Aceptación Tecnológica (TAM, por sus siglas en inglés) propuesto por Davis (1989), el cual plantea que la aceptación de una tecnología depende de la utilidad percibida y la facilidad de uso. Este modelo ha sido ampliamente utilizado en la literatura especializada.

Diversos antecedentes reportados en la literatura respaldan el potencial educativo de la Programación Competitiva y sus plataformas. Los autores Coore y Fokum (2019) destacan que estas herramientas permiten automatizar procesos de evaluación y retroalimentación, mientras que Moreno Cadavid y Pineda Corcho (2018) evidencian mejoras significativas en el rendimiento académico al integrarlas en cursos formales. Asimismo, Teran-Pomier (2016) subraya su relevancia en el ámbito laboral y su valor como estrategia de empleabilidad. Sin embargo, la mayoría de estas investigaciones se enfocan en los beneficios para el estudiantado, dejando en segundo plano el rol del profesorado como agente clave en su implementación, así como las ventajas de su uso como estrategia didáctica y de evaluación automatizada.

Este estudio se realiza en el contexto del Instituto Tecnológico de Ciudad Juárez, donde desde hace varios años se han promovido concursos internos, estatales e internacionales de programación competitiva. Si bien estas actividades han tenido buena recepción entre los estudiantes, su integración formal como estrategia de enseñanza en los programas curriculares aún es incipiente. Por ello, se vuelve necesario explorar las condiciones institucionales, formativas y actitudinales que afectan su adopción docente.

La investigación se enmarca en un enfoque cuantitativo-descriptivo y tiene como objetivo general determinar las percepciones y experiencias del profesorado del área de programación respecto al uso de herramientas TIC de programación competitiva como estrategia didáctica en la educación superior.



Específicamente, se busca identificar las plataformas más conocidas, evaluar el nivel de experiencia docente, reconocer las barreras percibidas y valorar la disposición de los docentes para su integración en el aula.

Marco teórico

A continuación se profundiza en el marco teórico que sustenta este trabajo de investigación y se estructura en subsecciones que abordan los siguientes temas: los retos formativos y pedagógicos de la enseñanza de la programación; definición y aplicaciones educativas de la programación competitiva en carreras de ingeniería; el potencial didáctico de la programación competitiva frente a los retos formativos actuales; experiencias y aprendizajes derivados de la programación competitiva en el ITCJ; y finalmente se abordan los modelos de adopción tecnológica en la docencia, con especial énfasis en el modelo TAM planteado por Davis (1989).

Enseñanza de la programación: retos formativos y pedagógicos actuales

La enseñanza de la programación enfrenta retos persistentes y multifactoriales, que van desde la falta de habilidades previas y la complejidad técnica del contenido, hasta la desmotivación estudiantil y la rigidez de los métodos tradicionales. Por lo anteriormente mencionado, es necesario transformar la enseñanza mediante estrategias activas, personalizadas y centradas en el desarrollo del pensamiento computacional y la resolución de problemas (Agbo et al., 2019; Figueiredo & García-Pévalo, 2019; Medeiros et al., 2019; Santos et al., 2020). Los autores mencionados coinciden en que la programación es difícil para principiantes, por su naturaleza abstracta y porque requiere habilidades lógico-matemáticas y de pensamiento computacional. También encontraron que la motivación y el compromiso de los estudiantes son factores clave para el proceso de aprendizaje de la programación.

La Figura 2.1 presenta una síntesis de los estudios revisados en la literatura, en relación con los conceptos clave vinculados a la enseñanza de la programación. A continuación, se describen dichos conceptos junto con los principales desafíos asociados a cada uno de ellos.



Figura 2.1. Los retos de la enseñanza de la programación



Fuente: Elaboración propia con base en (Agbo et al., 2019; Figueiredo & García-Pêalvo, 2019; Medeiros et al., 2019; Santos et al., 2020)

- *Pensamiento Computacional (CT)*. Muchos estudiantes carecen de esta habilidad clave para resolver problemas mediante conceptos informáticos, lo que dificulta el aprendizaje inicial (Agbo et al., 2019; Figueiredo & García-Pêalvo, 2019).
- *Resolución de problemas*. Es una de las competencias más difíciles de desarrollar, ya que implica traducir problemas reales a código. (Agbo et al., 2019; Medeiros et al., 2019).
- *Motivación y compromiso estudiantil*. Los alumnos se desmotivan y desertan de las materias de programación debido a la percepción de dificultad y el fracaso temprano (Medeiros et al., 2019; Santos et al., 2020).
- *Diversidad de perfiles estudiantiles*. Las diferencias en conocimientos y estilos de aprendizaje demandan enfoques personalizados (Figueiredo & García-Pêalvo, 2019; Medeiros et al., 2019).
- *Evaluación y retroalimentación*. La falta de herramientas eficaces y el gran tamaño de los cursos dificultan una retroalimentación oportuna, continua y útil. (Medeiros et al., 2019; Santos et al., 2020).
- *Metodologías activas de aprendizaje*. Aunque favorecen un aprendizaje significativo, su implementación requiere más planificación, recursos y superación de la resistencia al cambio. (Agbo et al., 2019; Figueiredo & García-Pêalvo, 2019; Medeiros et al., 2019; Santos et al., 2020).

Definición y aplicaciones educativas de la programación competitiva en carreras de ingeniería

La **programación competitiva** es una disciplina intelectual donde los participantes deben resolver problemas de carácter lógico-matemático utilizando técnicas de programación, integrando dos competencias fundamentales: el **diseño de algoritmos** y su **implementación efectiva** en un tiempo limitado (Lora Patiño et al., 2021; Sinza-Díaz et al., 2023). El concurso de programación más icónico es la Competición Internacional Universitaria de Programación de ACM (ICPC), el cual se realizó por primera vez en 1970 y en la actualidad sigue siendo el más importante a nivel mundial (Brito & Gonçalves, 2019; Friss De Kereki & Luna, 2022; Lora Patiño et al., 2021; Nair, 2020; Raman et al., 2018).

En un estudio realizado por Sinza-Díaz y colaboradores (2023), mediante un mapeo sistemático de la literatura identificaron que, al practicar programación competitiva, se desarrollan habilidades clave como la resolución de problemas y el trabajo en equipo.

Friss De Kereki y Luna (2022), reportan que, tras dos décadas de organización de concursos en la Universidad ORT Uruguay, se han beneficios significativos como la mejora académica, el fortalecimiento del cuerpo docente con exalumnos participantes y una mayor vinculación con el sector tecnológico, a pesar de resultados modestos en competencias internacionales.

En la Universidad de Amrita (India), Nair (2020) documentó una experiencia de más de 15 años integrando la programación competitiva tanto dentro del currículo y como actividad extracurricular. El autor observó una fuerte correlación entre los resultados de aprendizaje (COs) y los resultados del programa (POs), especialmente en las siguientes áreas: conocimiento en ingeniería, análisis de problemas, uso de herramientas modernas, trabajo individual y en equipo, así como en la comunicación efectiva. Gracias a la implementación continua de este enfoque pedagógico, se tuvo un impacto positivo en cuanto a empleabilidad, habilidades técnicas, participación en competencias internacionales e inserción laboral en empresas como Google, Facebook, Amazon.

Lora Patiño y otros autores (2020) proponen la programación competitiva como estrategia para fomentar el aprendizaje colaborativo en ingeniería, destacando el uso de maratones de programación, semilleros de investigación en algoritmia, plataformas de evaluación automática y el desarrollo de herramientas como MaratonApp. Los resultados mostraron que durante los últimos tres años, los estudiantes perciben



mejoras en trabajo en equipo, pensamiento crítico, creatividad e innovación.

Tabla 2.1. Resumen de hallazgos al implementar programación competitiva como estrategia didáctica

Tipo de habilidad	Descripción del Hallazgo		Fuente(s)
Técnicas (Hard Skills)		Mejora en resolución de problemas, pensamiento algorítmico, dominio de estructuras de datos y lenguajes de programación.	Friss de Kereki & Luna (2022); Nair (2020); Morales Ortiz et al. (2024); Sinza Díaz et al. (2023)
Lógico-interpretativas		Desarrollo de la capacidad para analizar y resolver problemas de la vida real, fortaleciendo el razonamiento lógico y la interpretación de situaciones complejas.	Lora Patiño et al. (2021)
Trabajo en equipo		Fortalecimiento de la colaboración, comunicación y toma de decisiones en equipo, especialmente en competencias como ICPC.	Friss de Kereki & Luna (2022); Morales Ortiz et al. (2024); Sinza Díaz et al. (2023)
Pensamiento crítico y lógico		Desarrollo de habilidades para analizar, abstraer y resolver problemas complejos bajo presión.	Nair (2020); Morales Ortiz et al. (2024); Sinza Díaz et al. (2023)
Autonomía autoaprendizaje	y	Fomento del aprendizaje autodirigido mediante plataformas como OmegaUp, con retroalimentación inmediata y gamificación.	Morales Ortiz et al. (2024); Sinza Díaz et al. (2023)
Motivación confianza	y	Aumento del interés por la programación, sentido de logro y confianza en competencias técnicas.	Morales Ortiz et al. (2024); Friss de Kereki & Luna (2022); Sinza Díaz et al. (2023)
Inserción laboral y profesional		Participantes en programación competitiva acceden a mejores oportunidades laborales en empresas como Google, Amazon y Facebook.	Nair (2020); Friss de Kereki & Luna (2022); Sinza Díaz et al. (2023)
Impacto académico		Mejora en el rendimiento académico en asignaturas clave como Fundamentos de Programación y Estructura de Datos.	Morales Ortiz et al. (2024); Sinza Díaz et al. (2023)
Innovación educativa		Integración de la programación competitiva como estrategia pedagógica en cursos, clubes y proyectos institucionales.	Friss de Kereki & Luna (2022); Morales Ortiz et al. (2024); Sinza Díaz et al. (2023)

Fuente: Elaboración propia

Por su parte, Morales Ortiz et al. (2024) reportan que la implementación de clubes de programación y el uso de plataformas como OmegaUp con un enfoque gamificado, aumentaron la motivación, participación y tasas de aprobación en asignaturas básicas de programación en el Instituto Tecnológico de Cancún. Gracias a su implementación, aumentaron los índices de aprobación de materias básicas de programación de la carrera de Ingeniería en Sistemas Computacionales.

El uso de plataformas tecnológicas que permiten la evaluación automática ha sido clave para la expansión de la programación competitiva como estrategia educativa (Brito & Gonçalves, 2019; Coore & Fokum, 2019; Friss De Kereki & Luna, 2022; Lora Patiño et al., 2021; Morales Ortiz et al., 2024; Nair, 2020).

Los estudios coinciden en que la programación competitiva, además de fortalecer habilidades técnicas



como la lógica algorítmica, también promueve el desarrollo de habilidades blandas como la comunicación, la colaboración y la resolución de problemas, altamente valoradas en el ámbito profesional. La Tabla 2.1 resume las principales habilidades desarrolladas mediante su implementación como estrategia didáctica.

Por lo anteriormente mencionado, la utilización de la programación competitiva como una estrategia didáctica ayuda no solamente a desarrollar las competencias de programación, sino otras habilidades importantes en el ámbito laboral.

El potencial didáctico de la programación competitiva frente a los retos formativos actuales

La enseñanza de la programación enfrenta retos persistentes como el bajo rendimiento académico, la falta de motivación y las dificultades para desarrollar el pensamiento computacional, lo cual demanda enfoques pedagógicos más activos, significativos y contextualizados (Agbo et al., 2019; Figueiredo & García-Pévalo, 2019). En este contexto, la programación competitiva como estrategia didáctica, ofrece múltiples beneficios, para los alumnos como, por ejemplo: fomenta la resolución de problemas en contextos reales, fortalece el trabajo colaborativo, fortalece el pensamiento lógico y mejora la capacidad de trabajar bajo presión (Lora Patiño et al., 2021; Nair, 2020; Sinza-Díaz et al., 2023).

Aunque una sola estrategia didáctica difícilmente resolverá por sí misma un problema tan complejo y ampliamente estudiado, las fuentes consultadas indican que puede favorecer el proceso de enseñanza-aprendizaje en materias de programación. Con base en los retos de la enseñanza de la programación descritos en la sección 2.1 y con las ventajas que brinda la programación competitiva como estrategia didáctica explicados en la sección 2.2, se plantea el potencial de este enfoque didáctico para un mejor proceso de enseñanza. Lo anterior se resume en la tabla 2.2 y se describe detalladamente a continuación.



Tabla 2.2. La programación competitiva y sus beneficios educativos

Reto de enseñanza de la programación	Ventaja relacionada de la programación competitiva
Pensamiento computacional	Desarrollo de pensamiento lógico y creativo (Lora Patiño et al., 2021). Gracias a la práctica constante se mejora pensamiento algorítmico (Sinza-Díaz et al., 2023)
Resolución de problemas	Se desarrolla la capacidad de resolución eficiente de problemas (Sinza-Díaz et al., 2023). La participación en maratones, concursos de programación como el ICPC ayuda a resolver problemas reales bajo presión (Lora Patiño et al., 2021; Morales Ortiz et al., 2024; Nair, 2020)
Motivación y compromiso estudiantil	El enfoque gamificado y el carácter competitivo inherente a este tipo de programación, son un motivador clave (Sinza-Díaz et al., 2023). La participación en maratones y concursos como ICPC incrementan el compromiso de los estudiantes (Lora Patiño et al., 2021; Morales Ortiz et al., 2024; Nair, 2020)
Diversidad de perfiles estudiantiles	Las plataformas de programación competitiva permiten clasificar los desafíos por niveles, en función del dominio que tengan los estudiantes (Brito & Gonçalves, 2019; Sinza-Díaz et al., 2023). Además, estudiantes con diferentes niveles de experiencia pueden trabajar colaborativamente utilizando estas plataformas (Morales Ortiz et al., 2024; Nair, 2020)
Evaluación y retroalimentación	Plataformas como CodeForces, BOCA y Omegaup ofrecen un sistema de evaluación gamificado, automatizado y con retroalimentación en tiempo real (Brito & Gonçalves, 2019; Lora Patiño et al., 2021; Morales Ortiz et al., 2024; Nair, 2020; Sinza-Díaz et al., 2023)
Metodologías activas de aprendizaje	Se fomenta mediante semilleros, clubes y participación en concursos y hackatones (Brito & Gonçalves, 2019; Lora Patiño et al., 2021; Morales Ortiz et al., 2024). También se logra integrando la programación competitiva en las materias de la retícula y como actividad extraescolar (Lora Patiño et al., 2021; Nair, 2020; Sinza-Díaz et al., 2023)

Fuente: Elaboración propia

Con respecto al **pensamiento computacional**, el cual consiste en la habilidad para abstraer y modelar soluciones, la programación competitiva exige descomponer problemas complejos, abstraer soluciones, diseñar algoritmos eficientes; fortaleciendo así habilidades transferibles a otros contextos académicos y profesionales (Agbo et al., 2019; Figueiredo & García-Péalo, 2019; Nair, 2020; Sinza-Díaz et al., 2023).

Cada ejercicio de programación competitiva es un reto de **resolución de problemas** bajo presión, la capacidad de diseñar soluciones algorítmicas, lo cual potencia la agilidad mental, estimula la creatividad algorítmica y fortalece la habilidad para identificar soluciones óptimas (Agbo et al., 2019; Lora Patiño et al., 2021; Medeiros et al., 2019; Nair, 2020).

La **motivación y el compromiso estudiantil**, claves en el aprendizaje, se ven beneficiados por el formato competitivo, el sistema de rankings y la posibilidad de superación personal, lo cual incrementa el interés y la participación activa del alumnado (Lora Patiño et al., 2021; Medeiros et al., 2019; Morales Ortiz et al., 2024; Nair, 2020; Santos et al., 2020).



Ante el reto de la **diversidad de perfiles estudiantiles**, la programación competitiva permite adoptar niveles de dificultad, fomentar el aprendizaje autónomo y promover el trabajo colaborativo entre estudiantes con distintos grados de experiencia; lo anterior permite que estudiantes de diferentes grados de experiencia y saberes colaboren entre sí (Brito & Gonçalves, 2019; Figueiredo & García-Péalo, 2019; Morales Ortiz et al., 2024; Sinza-Díaz et al., 2023).

En lo relativo a la **evaluación y retroalimentación**, el uso de plataformas como Codeforces, OmegaUp o BOCA permite una retroalimentación inmediata, automatizada y gamificada, favoreciendo un seguimiento puntual del proceso de aprendizaje (Lora Patiño et al., 2021; Medeiros et al., 2019; Morales Ortiz et al., 2024; Santos et al., 2020; Sinza-Díaz et al., 2023).

En la actualidad, la mayoría los estudiantes de educación a nivel superior pertenecen a la Generación Z, esta cohorte generacional se distingue por su familiaridad con las tecnologías digitales, su autonomía en el aprendizaje, su inclinación hacia el trabajo colaborativo y preferencia por contenidos visuales y con aplicación práctica en el mundo real (Saxena & Mishra, 2021; Wajdi et al., 2024). En este sentido, las **metodologías de aprendizaje activo** resultan especialmente pertinentes al fomentar la participación activa y colaborativa del estudiante (Agbo et al., 2019; Figueiredo & García-Péalo, 2019; Medeiros et al., 2019; Santos et al., 2020). No obstante, su implementación implica desafíos como la necesidad de mayor planificación, recursos y disposición al cambio metodológico por parte del profesorado (Agbo et al., 2019; Medeiros et al., 2019; Santos et al., 2020).

Finalmente, la programación competitiva presenta un alto potencial para integrarse en asignaturas como Fundamentos de Programación o Estructuras de Datos. Además de reforzar habilidades técnicas, también desarrolla competencias transversales como liderazgo, resiliencia y gestión del tiempo. Diversos autores han documentado experiencias exitosas mediante clubes, semilleros, eventos académicos y su incorporación curricular (Brito & Gonçalves, 2019; Lora Patiño et al., 2021; Morales Ortiz et al., 2024; Nair, 2020; Sinza-Díaz et al., 2023).

Experiencias y aprendizajes derivados de la programación competitiva en el ITCJ

A pesar de que en el Instituto Tecnológico de Ciudad Juárez (ITCJ) no se ha utilizado la **programación competitiva** como enfoque didáctico, se tiene experiencia aplicándola en contextos competitivos, desde el 2014. En la figura 2.2, se presenta una línea del tiempo de su utilización y evolución a lo largo del



tiempo.

Figura 2.2. Línea de tiempo de la programación competitiva en el ámbito del ITCJ



Fuente: Elaboración propia

Los eventos de programación competitiva, organizados por el Departamento de Sistemas y Computación, se evaluaban manualmente hasta 2018, en donde se utilizó la plataforma DOMjudge, cuando el instituto fue sede del Concurso Estatal de Programación del Estado de Chihuahua.

Además de ser un proceso engorroso, la evaluación manual presentaba diversas complicaciones como: susceptibilidad a errores humanos y falta de objetividad. La presión por entregar resultados con rapidez, especialmente para la elaboración de reconocimientos, generaba un ambiente de estrés entre los miembros del jurado. Además, la falta de automatización dificultaba la transparencia del proceso evaluativo. Por lo anteriormente mencionado y ante la buena experiencia de utilizar la plataforma DOMjudge, a partir de 2019 se decidió utilizar para todos los eventos la plataforma OmegaUp, lo cual fue comunicado tanto a los alumnos del plantel como a las escuelas de nivel medio superior que habían participado en los concursos.

Durante la pandemia se suspendieron los concursos, pero en 2022 se retomaron utilizando nuevamente **OmegaUp**. Ese mismo año, estudiantes de la Universidad Autónoma de Ciudad Juárez (UACJ), con experiencia en el ICPC, impartieron un taller de algoritmia en el ITCJ, lo que motivó a estudiantes de segundo semestre a involucrarse en competencias como el **ICPC**, el **CodingCup** del TecNM y concursos estatales. En 2023, un equipo del ITCJ clasificó al regional del ICPC en Guadalajara y, posteriormente, se fundó el **Club de Programación y Algoritmia**, lo que marcó un hito en la institucionalización de esta práctica.

Desde entonces, los logros han sido significativos: en 2022 se obtuvo el cuarto lugar en el concurso estatal, mientras que en 2023 y 2024 los equipos del ITCJ obtuvieron el primer lugar estatal. Además, equipos emergentes lograron el segundo y tercer lugar en esos años, respectivamente. El instituto también retomó su papel como sede del concurso estatal en 2024, capacitando a otros docentes en el uso de plataformas como OmegaUp.

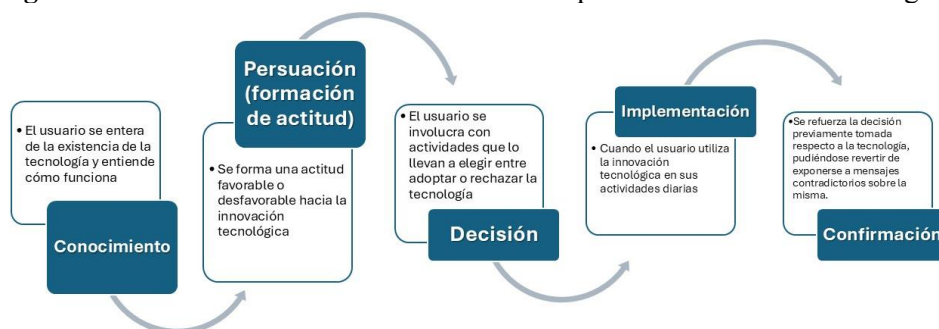
En resumen, la experiencia del ITCJ con la programación competitiva ha transitado de eventos manuales y extracurriculares a una práctica institucional con impacto positivo en el desarrollo de competencias técnicas y habilidades blandas en los estudiantes que se involucran en este tipo de actividades. Sin embargo, estos beneficios han sido aprovechados principalmente por estudiantes que participan voluntariamente, sin una integración plena en el plan de estudios formal. Por ello, resulta oportuno explorar formas de incorporar la programación competitiva en el desarrollo regular de las asignaturas, ampliando así su alcance e impacto en la formación profesional de los estudiantes.

Modelos de adopción tecnológica en la docencia

La adopción tecnológica es el proceso mediante el cual una persona, grupo u organización decide incorporar y utilizar una nueva tecnología en sus actividades cotidianas, laborales o educativas (Rogers, 2003). El proceso descrito por el autor se compone de cinco etapas, las cuales se ilustran en la figura 2.3. De acuerdo con el proceso descrito por Rogers (2003), para que una innovación tecnológica sea adoptada en cualquier contexto primeramente, hay que conocerla y entenderla. Lo anterior no es suficiente, ya que la siguiente fase consiste en formar una actitud favorable o adversa a la tecnología recién conocida. Posteriormente el usuario u organización se involucra en actividades que lo llevarán a elegir entre adoptar o rechazar dicha tecnología. Si es que se decide adoptar la tecnología se utilizará en las actividades diarias. Conforme se utiliza la innovación tecnológica, se refuerza la decisión tomada anteriormente al respecto, pudiéndose revertir. Como se observa el proceso no necesariamente es lineal; incluso puede ser que, aunque una tecnología se implemente, su utilización no se consolide.



Figura 2.3. Proceso de toma de decisiones al adoptar una innovación tecnológica



Fuente: Elaboración propia, con base en el proceso descrito por Rogers (2003)

El hecho de que se desarrolle una nueva tecnología que prometa ser muy eficiente para resolver una necesidad dada, no necesariamente implica su adopción. Por lo anteriormente mencionado, también se han estudiado los factores que favorecen la aceptación de una nueva tecnología. El Modelo de Aceptación Tecnológica (TAM, por sus siglas en inglés) propuesto por Fred Davis (1989), se ha utilizado ampliamente para predecir la aceptación de tecnologías. Su propósito es explicar cómo dos percepciones clave: utilidad percibida (qué tanto mejora el desempeño) y facilidad de uso percibida (qué tan fácil es de usar) influyen en la intención de adoptar una tecnología. El modelo mencionado es útil para diseñar, evaluar e implementar tecnologías en contextos educativos, organizacionales y sociales. En educación, la adopción tecnológica implica la incorporación de herramientas digitales (como plataformas, software, recursos interactivos) en los procesos de enseñanza y aprendizaje (Teo, 2011). Este proceso está influido por factores como: la percepción de utilidad pedagógica, facilidad de uso, capacitación docente, apoyo institucional e infraestructura tecnológica.

En el contexto del ITCJ, la adopción tecnológica de la programación competitiva está consolidada en cuanto al uso de plataformas como OmegaUp para la organización de concursos de programación. No obstante, aún queda por explorar su adopción como apoyo didáctico para la impartición de materias relacionadas con la programación en los planes de estudio de las ingenierías. La adopción de este recurso tecnológico en los concursos se dio de manera paulatina y sin una planeación o directriz institucional. Sin embargo, se pueden aprovechar los aprendizajes empíricos al respecto y los casos de estudio analizados en la literatura para implementar de manera sistémica este tipo de innovación tecnológica en el contexto pedagógico de la enseñanza de la programación.

METODOLOGÍA

Este estudio se desarrolló bajo un enfoque mixto, predominantemente cuantitativo y descriptivo, complementado por elementos cualitativos mediante preguntas abiertas, con el objetivo de profundizar en percepciones y comentarios de los docentes que participaron. La investigación se clasifica como descriptiva y exploratoria, ya que busca caracterizar el nivel de conocimiento, experiencia, percepción e interés de los profesores con respecto a la programación competitiva, así como identificar barreras y posibles formas de implementación en el contexto educativo.

El diseño metodológico fue observacional, transversal y no experimental, ya que se recolectaron los datos en un único momento en el tiempo, sin manipulación de variables y a partir de la observación de respuestas expresadas por los docentes.

La población de estudio estuvo conformada por docentes de nivel medio superior y superior del sistema educativo público mexicano en Ciudad Juárez, Chihuahua; principalmente en áreas de ciencias computacionales, tecnología y programación. Se utilizó un muestreo no probabilístico, considerando a los docentes que voluntariamente aceptaron participar en la encuesta. La muestra final fue de 28 docentes, quienes respondieron de manera anónima y voluntaria.

Para recopilar los datos se utilizó un formulario en línea estructurado, compuesto por 24 elementos que incluyeron preguntas cerradas, de selección múltiple y preguntas abiertas. El cuestionario fue elaborado concretamente para este estudio, con base en el Modelo de Aceptación de Tecnología (TAM) desarrollado por Davis (1989). En su diseño se incluyeron aspectos clave planteados en el modelo como: la percepción de utilidad y facilidad de uso de la herramienta, así como la actitud de los docentes, su disposición a utilizarla y los obstáculos que podrían enfrentar para implementarla. La encuesta fue implementada a través de la plataforma Microsoft Forms, utilizando la infraestructura institucional y distribuida mediante correo electrónico y redes institucionales. Las respuestas fueron exportadas a una hoja de cálculo en Excel para su posterior análisis. El instrumento de recolección de datos, consideró los siguientes puntos:

- *Datos generales:* Se recopilaron datos demográficos como género, edad, nivel de estudios y el sistema educativo donde imparten clases (media superior o superior).
- *Familiaridad con la programación competitiva:* Se indagó si conocían el concepto, cómo lo



habían conocido y si sabían usar plataformas específicas como OmegaUp, HackerRank o similares.

- *Experiencia en uso académico o profesional:* Se preguntó si han utilizado programación competitiva en clases o en contextos profesionales, incluyendo si han configurado problemas o concursos.
- *Aplicación en el aula:* Se exploró si han usado problemas de programación competitiva con sus estudiantes, qué plataformas emplearon y cómo integraron estas actividades.
- *Percepción de beneficios:* Se pidió su opinión sobre qué beneficios aporta la programación competitiva al desarrollo de habilidades de los estudiantes.
- *Áreas educativas relevantes:* Se les preguntó en qué asignaturas o niveles consideran que esta estrategia puede ser más útil (como cursos avanzados, introducción a la programación, competencias académicas, etc.).
- *Opinión sobre su incorporación al currículo:* Se investigó si creen que debería integrarse formalmente en los planes de estudio de nivel medio superior o superior.
- *Propuestas de integración:* Se les solicitó ideas sobre cómo podría incorporarse en las clases (por ejemplo, como actividades evaluables o proyectos prácticos).
- *Nivel de dificultad adecuado:* Se consultó qué grado de dificultad consideran apropiado para los ejercicios, según el nivel educativo.
- *Barreras percibidas:* Se identificaron factores que dificultan la implementación en el aula, como falta de tiempo, capacitación o recursos.
- *Apoyos necesarios:* Se preguntó qué tipo de apoyo necesitarían para integrar la programación competitiva en sus clases.
- *Interés en formación:* Finalmente, se indagó si estarían interesados en recibir capacitación o talleres sobre el tema y si tenían comentarios adicionales.

En cuanto a las consideraciones éticas, se garantizó la confidencialidad y anonimato de los participantes, quienes fueron informados del propósito de la investigación y dieron su consentimiento informado implícito al completar el cuestionario. No se recolectaron datos personales sensibles y se aseguró que



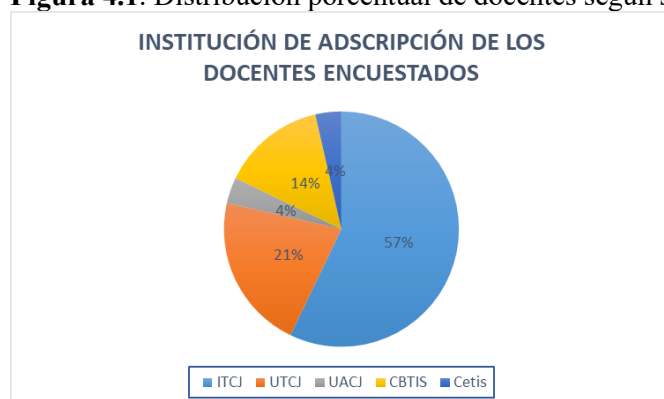
los resultados serían utilizados únicamente con fines académicos.

Las limitaciones del estudio son el tamaño reducido de la muestra y la aplicación en un solo momento temporal, lo cual limita la generalización de los hallazgos. Sin embargo, los resultados obtenidos brindan elementos para diagnosticar la aceptación, condiciones actuales y viabilidad para incorporar la programación competitiva como estrategia didáctica educativa en instituciones educativas donde se impartan materias relacionadas con la programación computacional.

RESULTADOS Y DISCUSIÓN

Los resultados obtenidos se organizan con base en los principales constructos del Modelo de Aceptación de Tecnología (TAM) de Davis (1989), considerando la utilidad percibida, la facilidad de uso percibida, la actitud hacia el uso, la intención de integración y las barreras identificadas por el profesorado.

Figura 4.1. Distribución porcentual de docentes según su institución de adscripción



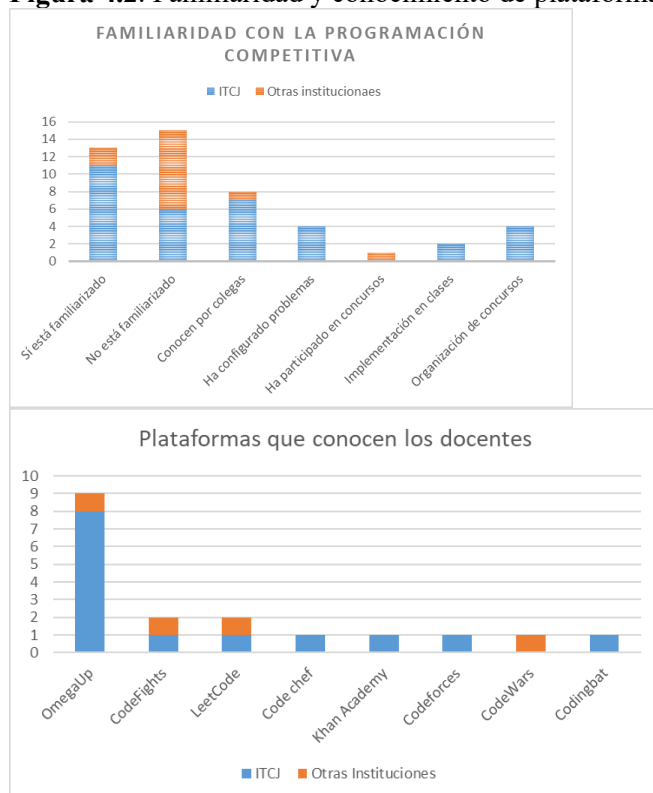
Fuente: Elaboración propia

El cuestionario aplicado fue contestado en su mayoría por docentes de nivel superior (82%), el resto son docentes de nivel medio superior. En su mayoría lo contestaron docentes del ITCJ, en un 57%. La idea de incluir docentes de otras instituciones fue analizar si es que la programación competitiva se ha implementado en otras instituciones de la localidad. En la figura 4.1 se muestra la distribución de las instituciones donde laboran los docentes encuestados.

Las primeras preguntas del cuestionario aplicado se refieren a la familiaridad con la programación competitiva. Solamente el 46% de los docentes manifestaron conocer el concepto de programación competitiva, sin embargo, se encontró que en el ITCJ la proporción de docentes que está familiarizado con el concepto es mayor (69%). En general, los docentes conocen acerca de la programación competitiva por medio de algún colega (62%). A pesar de esa familiaridad, solamente el 30% de los

docentes ha configurado problemas de programación competitiva en alguna plataforma, en especial para la organización de concursos de programación. De los docentes que han configurado ejercicios en plataformas, solamente la mitad lo ha implementado en sus clases, lo cual representa el 15% de los que conocen acerca de la programación competitiva. Lo anterior se resume en la primera gráfica de la figura 4.2.

Figura 4.2. Familiaridad y conocimiento de plataformas de los docentes



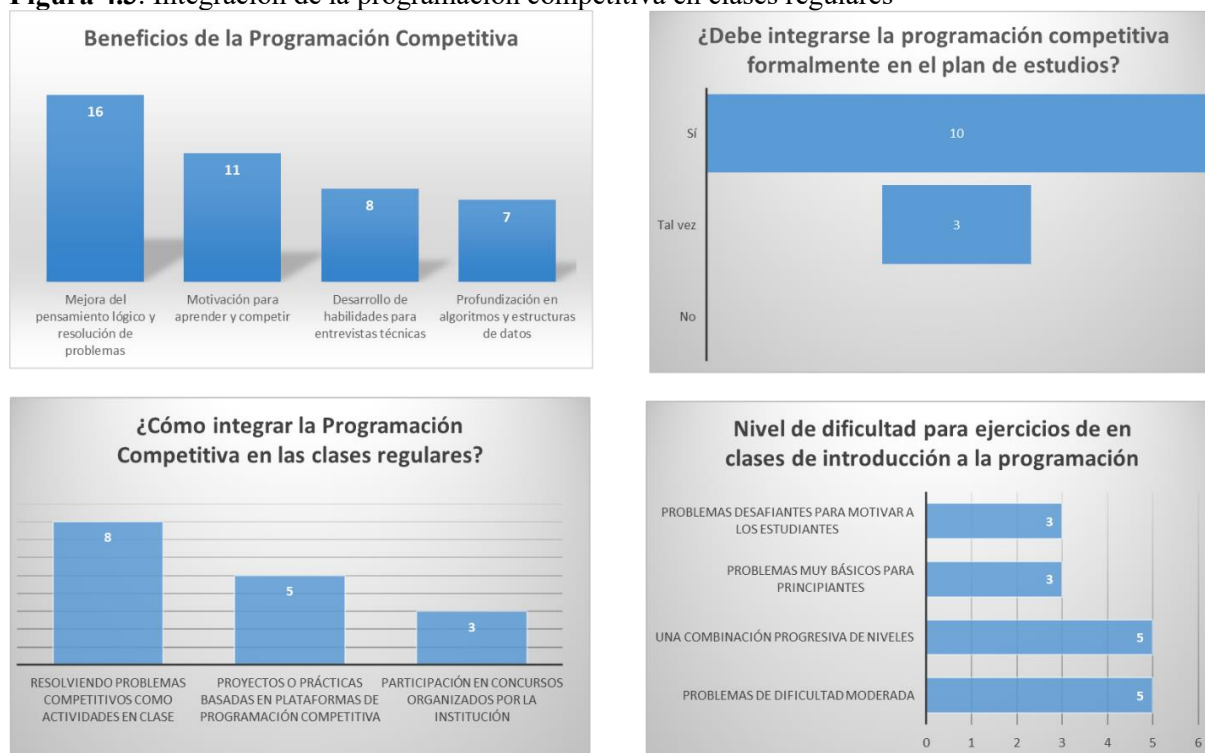
Fuente: Elaboración propia

En cuanto al conocimiento específico de plataformas, de los 13 docentes que están familiarizados con el concepto de programación competitiva, solamente 10 de ellos conocen alguna(s) plataforma(s), lo cual representa al 36% del total de los encuestados. De los diez profesores que conocen alguna plataforma, ocho pertenecen al ITCJ, mientras que los otros dos son docentes de otras instituciones de nivel superior. Del nivel media superior, ninguno manifestó conocer alguna plataforma. El porcentaje de desconocimiento de plataformas de programación competitiva de los docentes del ITCJ fue de un 47%, mientras que de las otras instituciones se reportó en un 85%. En la segunda gráfica de la figura 4.2, se muestran las plataformas que el grupo de los diez docentes mencionados conoce, siendo OmegaUp la que predomina. Cabe aclarar que algunos docentes manifestaron conocer más de una

plataforma.

En la primera gráfica de la figura 4.3 se muestran los beneficios de la programación competitiva que perciben los docentes encuestados, siendo el principal la mejora del pensamiento lógico y resolución de problemas, lo cual se relaciona con el concepto de pensamiento computacional mencionado por los autores Agbo y otros (2019), así como por Figueiredo y García-Pévalo (2019) en cuanto a los desafíos en la enseñanza de la programación.

Figura 4.3. Integración de la programación competitiva en clases regulares

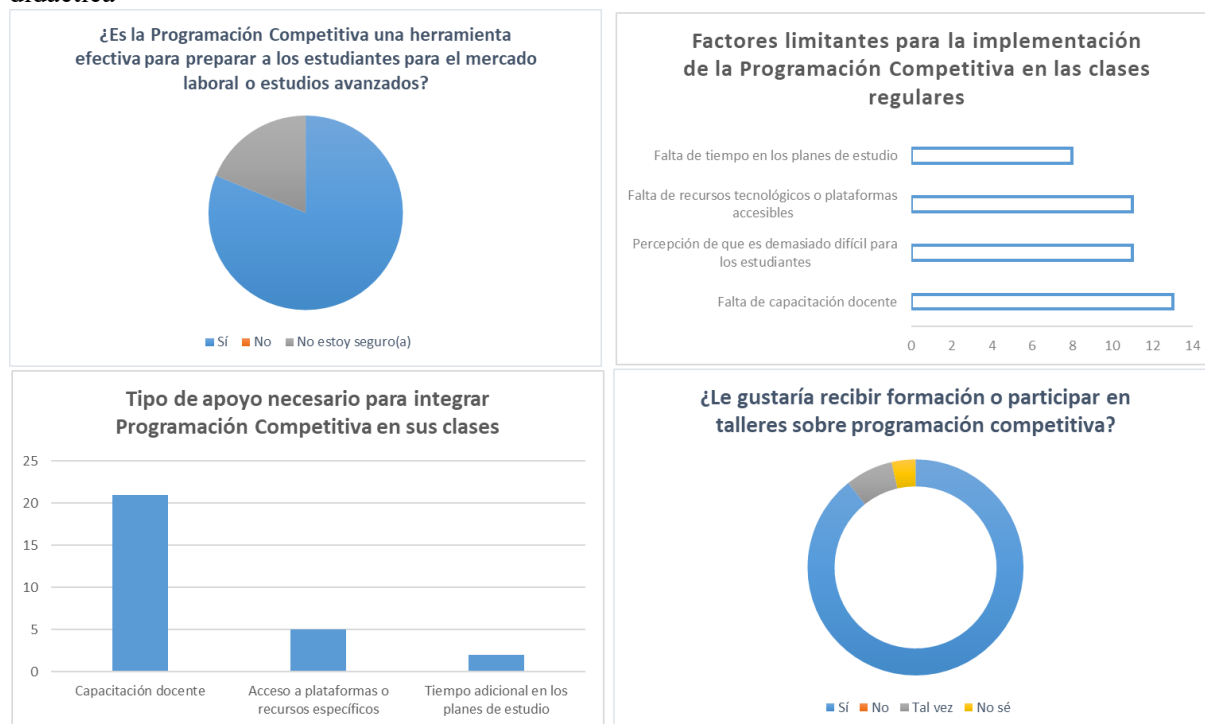


Fuente: Elaboración propia

De los docentes familiarizados con el concepto de programación competitiva el 77% considera que debe integrarse formalmente en los planes de estudio, lo cual se muestra en la segunda gráfica de la figura 4.3. Los resultados muestran que la mayoría de los docentes considera viable integrar la programación competitiva mediante actividades en clase con problemas competitivos. También destacan las prácticas en plataformas especializadas y la participación en concursos institucionales como estrategias complementarias, lo cual se muestra en la tercera gráfica de la figura 4.3. Además, con respecto al nivel de dificultad adecuado que consideran los docentes para estos ejercicios en cursos introductorios, predomina la preferencia por problemas de dificultad moderada o por una combinación progresiva de

niveles, lo cual refleja una tendencia a equilibrar el reto con el acompañamiento pedagógico, sin descuidar la motivación ni las necesidades de los estudiantes principiantes. Finalmente, en la primera gráfica de la figura 4.4 se muestra que el 89% del total de los docentes encuestados consideran que la programación competitiva es una herramienta efectiva para preparar al alumnado para el mercado laboral o estudios avanzados de computación.

Figura 4.4. Barreras y necesidades para la adopción de la programación competitiva como estrategia didáctica



Fuente: Elaboración propia

Los resultados también revelan una serie de factores limitantes que dificultan la integración de la programación competitiva en las clases regulares, lo cual se resumen en la segunda gráfica de la figura 4.4. La falta de capacitación docente es el obstáculo más señalado, con 13 menciones, seguida por la percepción de que los ejercicios son demasiado difíciles para los estudiantes y la falta de recursos tecnológicos o plataformas accesibles, ambos con 11 respuestas. Asimismo, la falta de tiempo en los planes de estudio fue identificada por 8 docentes como una barrera importante. Ante este panorama, la necesidad más urgente identificada por el profesorado es **la capacitación especializada en programación competitiva**, seleccionada por 21 participantes como el apoyo principal requerido para implementar estas estrategias y se observa en la tercera gráfica de la figura 4.4. Esta necesidad se ve reforzada por la disposición expresada por los docentes: el 100% respondió afirmativamente o con

apertura (tal vez) a la posibilidad de participar en talleres o procesos de formación sobre programación competitiva, lo cual se muestra en la cuarta gráfica de la figura 4.4.

CONCLUSIONES

Los resultados del estudio muestran que existe un reconocimiento generalizado del valor didáctico de la programación competitiva, pero su incorporación en el aula está lejos de ser una práctica común. Las barreras no son únicamente tecnológicas, sino también pedagógicas y organizacionales. Sin embargo, a pesar de las barreras identificadas, los docentes muestran disposición a formarse en estas herramientas si se les proporciona apoyo institucional.

Los hallazgos sobre las limitaciones percibidas por los docentes al integrar la programación competitiva pueden analizarse con enfoque en el Modelo de Aceptación Tecnológica (TAM) de Davis (1989), que plantea que la utilidad percibida y la facilidad de uso son determinantes clave en la adopción de nuevas tecnologías. En este caso concreto, si bien los docentes reconocen el valor didáctico de la programación competitiva (utilidad percibida), la **falta de capacitación**, la **percepción de dificultad para los estudiantes** y la **ausencia de recursos tecnológicos adecuados** reducen la percepción de facilidad de uso, lo cual obstaculiza su integración efectiva. Esta relación se refuerza con la alta demanda de **formación docente** (21/28 respuestas) como principal apoyo necesario, y el interés generalizado en participar en talleres (25/28 respuestas afirmativas), lo que evidencia una actitud positiva hacia la innovación, pero limitada por barreras prácticas.

A pesar de que aparentemente hay una apertura para recibir información y capacitación con respecto a la programación competitiva por parte de los docentes que participaron en el estudio, esto representaría solamente un primer paso para la adopción como tecnología para la implementación en los cursos regulares. Tal como se establece en el modelo de difusión de innovaciones propuesto por Rogers (2003) mostrado en la figura 2.3, el conocimiento de la innovación es solamente el primer paso del proceso. Rogers identifica cinco atributos que influyen en la adopción: ventaja relativa, compatibilidad, complejidad, posibilidad de prueba y observabilidad. En este sentido, los resultados muestran que aunque los docentes reconocen la ventaja relativa de esta estrategia (motivación, aprendizaje activo), identifican alta complejidad (por la percepción de dificultad y falta de preparación) y baja compatibilidad con las condiciones institucionales (limitado tiempo en planes de estudio y recursos).



Por lo anterior se concluye que la programación competitiva aún se encuentra en fases tempranas del proceso de adopción dentro del contexto educativo estudiado.

Como trabajo futuro y con el fin de lograr avanzar a su difusión efectiva, es necesario facilitar experiencias piloto, acompañamiento docente y entornos de prueba que reduzcan la incertidumbre. Los modelos de Rogers (2003) y la consideración de la utilidad percibida y la facilidad de uso planteada en el modelo de Davis (1989), serían modelos de referencia para lograr la adopción tecnológica de la programación competitiva y aprovechar sus ventajas como herramienta didáctica, las cuales se abordaron en la sección 2.3 y están sustentadas en la literatura consultada.

Se recomienda promover programas de formación docente, el diseño de bancos de ejercicios adaptados a planes de estudio y espacios de colaboración entre docentes para intercambiar experiencias y materiales. Finalmente es importante acompañar al profesorado en este proceso, primeramente, con una capacitación orientada a su práctica docente, así como seguimiento y orientación para que se aprovechen las ventajas discutidas en el marco teórico. Se busca iniciar con este proceso primeramente dentro de la institución, para posteriormente expandirlo a otras instituciones del ecosistema del nivel medio superior y superior de nuestra ciudad.

REFERENCIAS BIBLIOGRÁFICAS

- 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>
- Andújar, F. J., González-Escribano, A., Bastida, J., & Torres de la Sierra, Y. (2020). Aplicación de gamificación competitiva y colaborativa en asignaturas básicas de arquitectura de computadoras. Asociación de Enseñantes Universitarios de La Informática (AENUI).
- Brito, M., & Gonçalves, C. (2019). Codeflex: A Web-based Platform for Competitive Programming. 411.
- Coore, D., & Fokum, D. (2019). Facilitating course assessment with a competitive programming platform. SIGCSE 2019 - Proceedings of the 50th ACM Technical Symposium on Computer Science Education, 449–455. <https://doi.org/10.1145/3287324.3287511>



- Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 319–340.
- Figueiredo, J., & García-Pévalo, 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). Education, Research and Leadership in Post-pandemic Engineering: Resilient, Inclusive and Sustainable Actions. 1.
- Gonzalez-Escribano, A., Lara-Mongil, V., Rodriguez-Gutierrez, E., & Torres, Y. (2019). Toward improving collaborative behaviour during competitive programming assignments. *Proceedings of EduHPC 2019: Workshop on Education for High Performance Computing - Held in Conjunction with SC 2019: The International Conference for High Performance Computing, Networking, Storage and Analysis*, 68–74. <https://doi.org/10.1109/EduHPC49559.2019.00014>
- 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 Tecnologías de Informação*, 539–556.
- 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>
- Morales Ortiz, L. V., Ucan Cih, J. M., & Ramos Santiago, A. (2024). Clubes de programación y aprendizaje activo: del aula al éxito en programación competitiva con el uso de OmegaUp. Basachí. www.utpn.edu.mx.
- Moreno Cadavid, J., & Pineda Corcho, A. F. (2018). Competitive programming and gamification as strategy to engage students in computer science courses. *Espacios*, 39.
- 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>



- Raman, R., Vachharajani, H., & Achuthan, K. (2018). Students motivation for adopting programming contests: Innovation-diffusion perspective. *Education and Information Technologies*, 23(5), 1919–1932. <https://doi.org/10.1007/s10639-018-9697-3>
- Rogers, E. M. . (2003). *Diffusion of innovations* (5th ed.). Free Press.
- 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>
- Sinza-Diaz, Y. E., Oliva-Caipe, 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>
- Teo, T. (2011). Factors influencing teachers' intention to use technology: Model development and test. *Computers and Education*, 57(4), 2432–2440. <https://doi.org/10.1016/j.compedu.2011.06.008>
- Teran-Pomier, J. H. (2016). De las competencias de programación a una pedagogía educativa en La Paz Bolivia. *Innoeduca. International Journal of Technology and Educational Innovation*, 2(2), 128. <https://doi.org/10.20548/innoeduca.2016.v2i2.1897>
- Wajdi, M., Susanto, B., Made Sumartana, I., Agus Sutiarto, 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>

