

**Cita APA 7ma edición:** Morales Reyes, E., Gómez Manuel, E, Dominguez Campomanes, M. & Gilbón Aburto, A. (2025). Estrategias tecno-pedagógicas para el pensamiento computacional en POO: Impacto en rendimiento y motivación universitaria. En Gómez Manuel, E., Morales Reyes, E. & Juárez Juárez, Y. J. (Coords). *Innovación Educativa y Desarrollo Empresarial*. (pp. 25-43). Editorial Sinergy

# Capítulo 1

## **Estrategias tecno-pedagógicas para el pensamiento computacional en POO: Impacto en rendimiento y motivación universitaria**

Techno-Pedagogical Strategies for Computational Thinking in OOP: Impact on University Students' Performance and Motivation

Eunice Morales Reyes

Universidad Tecnológica del Sureste de Veracruz

 0000-0003-0658-6957 | [eunice.morales@utsv.edu.mx](mailto:eunice.morales@utsv.edu.mx)

Esbeidy Gómez Manuel

Universidad Tecnológica del Sureste de Veracruz

 0000-0003-0765-3402 | [esbeidy.gomez@utsv.edu.mx](mailto:esbeidy.gomez@utsv.edu.mx)

Margarita Dominguez Campomanes

Universidad Tecnológica del Sureste de Veracruz

 0000-0003-0736-6112 | [margarita.campomanes@utsv.edu.mx](mailto:margarita.campomanes@utsv.edu.mx)

Antonio Gilbón Aburto

Universidad Tecnológica del Sureste de Veracruz

 0000-0003-2548-6855 | [antonio.gilbon@utsv.edu.mx](mailto:antonio.gilbon@utsv.edu.mx)

## Resumen

En la formación de profesionales dedicados al desarrollo de software, las habilidades de pensamiento computacional se han convertido en competencias fundamentales. En este contexto, las estrategias tecno-pedagógicas adquieren un papel muy importante en la enseñanza de programación dentro de la educación superior. Este estudio tuvo como propósito examinar el efecto de las estrategias tecno-pedagógicas en el desarrollo del pensamiento computacional, el rendimiento académico y la motivación de estudiantes que cursan la asignatura de Programación Orientada a Objetos (POO) en la Universidad Tecnológica del Sureste de Veracruz. En la investigación se usó un enfoque cualitativo y cuantitativo. En la parte de medición, se realizó el estudio, con una versión adaptada del Test Bebras para las fases preliminares de la investigación y posteriores a ella, en ésta parte se incluyó un análisis del desempeño académico. Por otra parte, en la investigación de campo se realizaron entrevistas semiestructuradas. Los hallazgos fueron validados mediante triangulación metodológica. Los resultados mostraron avances importantes y significativos, con respecto a la tasa de reprobación, está reflejo una disminución de 32.18% a 21.19%, por otro lado, los puntajes del Test Bebras, aumentaron, especialmente en habilidades de mayor complejidad. Las entrevistas semiestructuradas realizadas a los estudiantes indicaron que al incorporar la herramienta Scratch y la retroalimentación inmediata, fueron pieza clave para favorecer la motivación de los estudiantes. Los hallazgos indican que las estrategias tecno-pedagógicas favorecen un aprendizaje activo y significativo, permitiendo no solo, el desarrollo de habilidades del pensamiento computacional, sino también, la mejora en los indicadores de desempeño académico. Se recomienda que otras Instituciones o programas formativos repliquen este enfoque para fortalecer en sus estudiantes las competencias del siglo XXI.

**Palabras clave:** Educación superior en computación, gamificación educativa, motivación pensamiento computacional, programación orientada a objetos y retención estudiantil.

## Abstract

In the training of professionals dedicated to software development, computational thinking skills have become fundamental competencies. In this context, techno-pedagogical strategies play a very important role in the teaching of programming within higher education. This study aimed to examine the effect of techno-pedagogical strategies on the development of computational thinking, academic performance, and motivation of students enrolled in the Object-Oriented Programming (OOP) course at the Technological University of the Southeast of Veracruz. The research employed both qualitative and quantitative approaches. For the measurement phase, the study used an adapted version of the Bebras Test in the preliminary and subsequent stages of the

research, which also included an analysis of academic performance. On the other hand, semi-structured interviews were conducted as part of the field research. The findings were validated through methodological triangulation. The results showed significant and meaningful progress: the failure rate decreased from 32.18% to 21.19%, while the Bebras Test scores increased, especially in higher-order skills. Semi-structured interviews with students indicated that the incorporation of the Scratch tool and immediate feedback were key elements in fostering student motivation. The findings suggest that techno-pedagogical strategies promote active and meaningful learning, enabling not only the development of computational thinking skills but also improvements in academic performance indicators. It is recommended that other institutions or training programs replicate this approach to strengthen 21st-century competencies among their students.

**Keywords:** Higher education in computing, educational gamification, computational thinking motivation, object-oriented programming, and student retention.

## INTRODUCCIÓN

La POO constituye un pilar fundamental en la formación de profesionales de tecnologías de la información (TI), no solo por su relevancia en el desarrollo de software escalable, sino como herramienta para cultivar el pensamiento computacional —definido por Wing (2006) como el proceso de resolver problemas mediante habilidades como la abstracción, descomposición, reconocimiento de patrones y diseño algorítmico—. Este enfoque, esencial para traducir desafíos del mundo real en estructuras de software eficientes (Wing, 2010), enfrenta obstáculos críticos en su enseñanza: estudios globales revelan que hasta el 32% de los estudiantes fracasan en cursos introductorios al no dominar la aplicación práctica de estos principios en la POO (Muñoz et al., 2012; Compañ-Rosique et al., 2015; García-Vega, 2019; Palma & Sarmiento, 2015). De acuerdo con datos recopilados, en contexto latinoamericanos, este desafío aumenta considerablemente, con tasas históricas de reprobación superiores al 30% (Argoti Álvarez, 2023; Jiménez y Carmona, 2023), en el contexto de la Universidad Tecnológica del Sureste de Veracruz, de acuerdo, con los datos cualitativos recopilados, el 67% de los estudiantes reportaron complicaciones, para representar problemas reales en diagramas de clases, que son, fundamentales en el análisis y diseño del desarrollo de software.

Actualmente la Industria 4.0, refleja la importancia de incorporar, el pensamiento computacional en programas afines a las ciencias computacionales, ya que se proyecta la

necesidad de contratar a 69 millones de profesionales en TI para 2026 (Bureau of Labor Statistics, 2019; Flores, 2023), permitiendo el diseño e implementación de sistemas programados por capas, o modulares, la reutilización de código y la optimización de procesos mediante algoritmos (Wing, 2006; García-Peñalvo, 2018). Sin embargo, las metodologías tradicionales de enseñanza, enfocadas en la sintaxis de lenguajes como Java o C++, han demostrado ser insuficientes para desarrollar estas competencias. Investigaciones en instituciones técnicas latinoamericanas identifican que 45% de los abandonos en carreras de TI están asociados a cursos intermedios de programación, donde la POO actúa como "cuello de botella" formativo (Mazaika, 2017), evidenciando una brecha entre las demandas laborales y las prácticas educativas.

Frente a este escenario, el presente estudio implementó un modelo pedagógico innovador que articula dos componentes clave: 1) retos adaptados del Test Bebras (Bebras, s. f.), herramienta validada internacionalmente para evaluar habilidades de pensamiento computacional, y 2) la plataforma PiensaCODE, que integra actividades gamificadas en Scratch bajo principios construccionistas (Papert, 1980). El objetivo general fue evaluar el impacto de estrategias tecno-pedagógicas sobre el desarrollo del pensamiento computacional, el rendimiento académico y la motivación en estudiantes de POO de la UTSV.

En esta investigación, se definieron las siguientes preguntas de investigación.

- ¿La incorporación de estrategias tecno-pedagógicas permitirá el desarrollo de habilidades del pensamiento computacional en los estudiantes de POO?
- ¿Cómo influyen dichas estrategias en el rendimiento académico y la tasa de reprobación de la asignatura?
- ¿Qué percepciones tienen los estudiantes sobre su motivación y participación al trabajar con estas estrategias?

Los resultados demostraron una reducción del 21.9% en la reprobación y un incremento promedio de 16.6 puntos en el Test Bebras, con mejoras del 40% en reconocimiento de patrones y 35% en abstracción, habilidades críticas para diseñar sistemas orientados a objetos eficientes.

El presente artículo, se distribuye en cuatro partes. En primer lugar, se muestra el marco metodológico, que detalla el diseño de Piensa CODE y la adaptación de los desafíos Bebras. Luego, se presentan los resultados, combinando el análisis cuantitativo con las experiencias de los estudiantes. Después, se discuten las implicaciones pedagógicas y se relacionan los resultados con la teoría de Wing (2006) y con los requisitos presentados por la industria del software. Para finalizar, se realizan recomendaciones para replantear la enseñanza de POO en América Latina, centrándose en el hecho de que el pensamiento computacional, en lugar de la codificación, es la base para educar a profesionales capaces de innovar en la era digital.

## METODOLOGÍA

En la investigación se usó un enfoque metodológico mixto, con un carácter explicativo secuencial, para analizar cómo las estrategias tecno-pedagógicas afectan el desarrollo del pensamiento computacional, el rendimiento académico y la motivación de los estudiantes que estudian Programación Orientada a Objetos (POO) en la Universidad Tecnológica del Sureste de Veracruz (UTSV), lo cual se respalda en Creswell y Plano Clark (2018), quienes señalan que los diseños mixtos secuenciales permiten comprender de manera más profunda los fenómenos al integrar fases cuantitativas y cualitativas. La figura 1, muestra la secuencia metodológica que se implementó en esta investigación, destacando la relación entre las dos fases del enfoque mixto, es importante mencionar que, esta combinación metodológica permitió no solo medir los cambios observables en las variables, sino también interpretar los factores que los explican desde la experiencia de los propios participantes.

**Figura 1.**

*Diseño metodológico mixto secuencial explicativo*



La población de estudio se conformó por 179 estudiantes de los programas de Técnico Superior Universitario (TSU) en Tecnologías de la Información e Ingeniería en Desarrollo y Gestión de Software, de la Universidad Tecnológica del Sureste de Veracruz. Para la selección de la muestra, se realizó, un muestreo probabilístico estratificado, seleccionando a 123 participantes, con un 95% de confianza y 5% de margen de error calculado con la herramienta Raosof.

La muestra final fue de 123 estudiantes y se distribuyó proporcionalmente por nivel académico y cuatrimestres, de tal manera que cada estrato estuviera representado de acuerdo con su peso en la población. La distribución fue la siguiente:

### Nivel TSU:

2do. cuatrimestre:

- Grupo 201: 20 estudiantes (aproximadamente 69% del grupo)
- Grupo 202: 19 estudiantes (aproximadamente 68% del grupo)

5to. cuatrimestre:

- Grupo único: 25 estudiantes (aproximadamente 69% del grupo)

### Nivel Ingeniería:

8vo. cuatrimestre:

- Grupo 801: 23 estudiantes (aproximadamente 70% del grupo)
- Grupo 802: 22 estudiantes (aproximadamente 69% del grupo)

11vo. cuatrimestre:

- Grupo único: 14 estudiantes (aproximadamente 67% del grupo) (Figura 2.)

Figura 2.

*Tamaño de la muestra*

Sample Size Calculator by Raosoft

What margin of error can you accept? 5%

What confidence level do you need? 95%

What is the population size? 179

What is the response distribution? 50%

Your recommended sample size is 123

Online surveys with Yoozi have completion rates of 66%!

Alternate scenarios

| With a sample size of         | 100   | 200   | 300   | With a confidence level of        | 90  | 95  | 99  |
|-------------------------------|-------|-------|-------|-----------------------------------|-----|-----|-----|
| Your margin of error would be | 6.53% | 4.61% | 3.46% | Your sample size would need to be | 108 | 123 | 142 |

Save effort, save time. Conduct your survey online with Yoozi.

More information

If 50% of all the people in a population of 20000 people drink coffee in the morning, and if you were to repeat the survey of 377 people ("Did you drink coffee this morning?") many times, then 95% of the time, your survey would find that between 45% and 55% of the people in your sample answered "yes". The remaining 5% of the time, or for 1 in 20 survey questions, you would expect the survey response to be more than the margin of error away from the true answer. When you survey a sample of the population, you don't know that you've found the correct answer, but you do know that there's a 95% chance that you're within the margin of error of the correct answer. Try changing your sample size and watch what happens to the alternate scenarios. That tells you what happens if you don't use the recommended sample size, and how M.O.E. and confidence level (that 95%) are related. To learn more if you're a beginner, read *Basic Statistics: A Modern Approach* and *The Cartoon Guide to Statistics*. Otherwise, look at the *more advanced books*.

In terms of the numbers you selected above, the sample size  $n$  and margin of error  $E$  are given by

Fuente: <http://www.raosoft.com/>

La prueba Bebras, que cuenta con un referente internacional, se adaptó al contexto mexicano para evaluar habilidades de pensamiento computacional. La prueba constaba de 13 ítems que fueron revisados para su adaptabilidad tanto en la redacción como en la dificultad, y se

ubicaron en 4 niveles relacionados (de básico a avanzado) que midieron competencias y habilidades como la descomposición, el reconocimiento de patrones, la generalización, la abstracción y el pensamiento algorítmico (ver detalles en la Tabla 1).

La validez del instrumento se realizó mediante la validez de contenido y fue revisada por tres profesionales en el campo del aprendizaje computacional. Además, la fiabilidad se confirmó con un alfa de Cronbach de 0.89, obtenido tras el análisis de consistencia interna con el software SPSS. También se aplicó una prueba piloto a 30 estudiantes para verificar la claridad de las preguntas y el tiempo de respuesta. (ver detalles en la Tabla 1).

**Tabla 1.**

*Preguntas de niveles – Test Bebras adaptado*

| Habilidades del Pensamiento Computacional |                             |            |                     |                              |                  |                     |                            |                                             |
|-------------------------------------------|-----------------------------|------------|---------------------|------------------------------|------------------|---------------------|----------------------------|---------------------------------------------|
| No.                                       | Ejercicio                   | Dificultad | Descomposi-<br>ción | Reconocimie-<br>nto Patrones | Abstracci-<br>ón | Generalizaci-<br>ón | Pensamiento<br>algorítmico | Dominio<br>Computaci-<br>onal               |
| NIVEL 1                                   |                             |            |                     |                              |                  |                     |                            |                                             |
| 1                                         | Huellas                     | A          |                     | x                            | x                |                     |                            | Conceptos<br>programaci-<br>ón              |
| 2                                         | Insectos en las<br>ramas    | A          |                     | x                            |                  | x                   | x                          | Conceptos<br>básicos -<br>programaci-<br>ón |
| 3                                         | Regalos                     | A          | x                   | x                            |                  |                     | x                          | Conceptos<br>básicos -<br>programaci-<br>ón |
| NIVEL 2                                   |                             |            |                     |                              |                  |                     |                            |                                             |
| 4                                         | Vías del tren               | B          |                     | x                            | x                |                     |                            | Algoritmos<br>-<br>Expresione-<br>s         |
| 5                                         | Caza del oso<br>de peluche  | B          |                     | x                            | x                | x                   |                            | Algoritmos<br>y<br>programaci-<br>ón        |
| 6                                         | Pintando las<br>casas       | B          |                     | x                            |                  |                     | x                          | Algoritmos<br>y<br>programaci-<br>ón        |
| NIVEL 3                                   |                             |            |                     |                              |                  |                     |                            |                                             |
| 7                                         | Representació-<br>n teatral | B          | x                   |                              | x                |                     |                            | Algoritmos<br>y<br>programaci-<br>ón        |
| 8                                         | Aspersores                  | B          |                     | x                            | x                |                     |                            | Algoritmos<br>y<br>Diagramas<br>de flujo    |
| 9                                         | Caja más<br>pesada          | B          |                     | x                            | x                |                     |                            | Algoritmos<br>y                             |

|         |                  |   |   |   |   |   | programación              |
|---------|------------------|---|---|---|---|---|---------------------------|
| NIVEL 4 |                  |   |   |   |   |   |                           |
| 10      | Abejas voladoras | C | x | X | x | x | Algoritmos y programación |
| 11      | Árbol Sudoku     | C |   | X |   | x | Algoritmos y programación |
| 12      | Visita al Museo  | C |   |   |   | x | Algoritmos y programación |
| 13      | Plantando flores | C | x | X |   | x | Algoritmos y programación |

La intervención pedagógica se llevó a cabo, durante el cuatrimestre de enero - abril de 2024, utilizando la plataforma web PiensaCODE creada para esta investigación. La herramienta integró el aprendizaje práctico en Scratch, Figura 3, con desafíos basados en el Test Bebras para mejorar y ejercitar habilidades de colaboración y resolución de problemas en un contexto de la vida real. Las actividades, organizadas en siete módulos temáticos (por ejemplo, "Descomposición" y "Reconocimiento de Patrones"), se reforzaron con un sistema de logros y retroalimentación inmediata para guiar la motivación intrínseca. Al final de la octava semana, los estudiantes habían trabajado sincrónicamente 3 horas (trabajando con docentes) y asincrónicamente 2 horas por semana en la plataforma, resultando en 80 horas de intervención.

**Figura 3.**

*Interfaz de lección: Descomposición – Plataforma PIENSACODE*





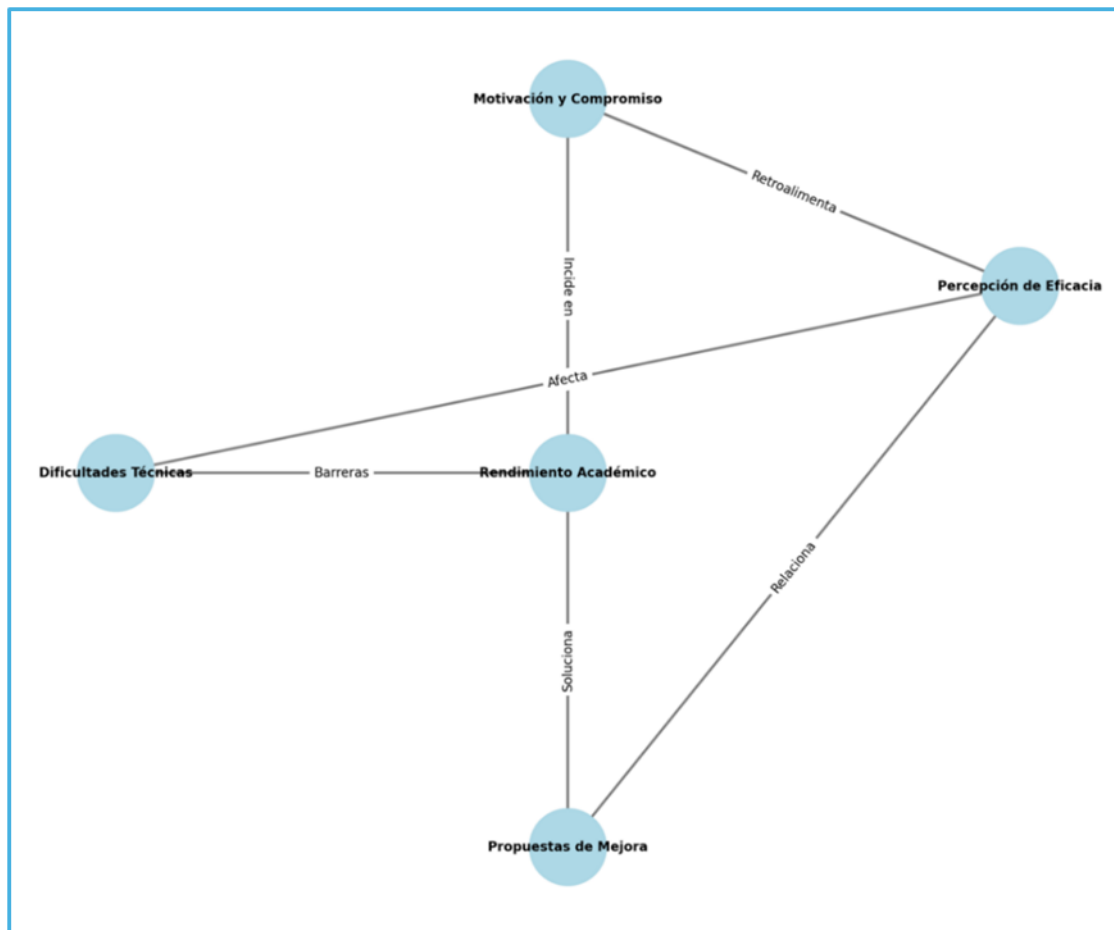
En la fase cuantitativa, se llevaron a cabo evaluaciones iniciales y finales mediante una prueba adaptada del test Bebras, complementadas con el análisis de los registros académicos oficiales, como calificaciones parciales y finales, índices de reprobación y deserción. Para examinar los resultados, se aplicaron diversas técnicas estadísticas: la prueba t de Student para muestras relacionadas permitió identificar mejoras individuales significativas; el análisis de varianza de dos vías (ANOVA) ayudó a explorar posibles diferencias en función del nivel académico y el género; y se emplearon correlaciones de Pearson para estudiar las relaciones entre las habilidades de pensamiento computacional, la motivación y el rendimiento académico. Los datos marginales se trataron omitiendo observaciones con <80% de respuestas completas, garantizando así la solidez de nuestros resultados.

Simultáneamente, la fase cualitativa se basó en entrevistas semiestructuradas y grupos focales con 25 participantes seleccionados intencionadamente y con diversidad en logros académicos. Todas las entrevistas fueron grabadas en audio y se basaron en un protocolo con temas como "motivación por gamificación" y "percepción de la utilidad", transcritos y codificados en ATLAS.ti mediante codificación abierta y axial.

Este proceso llevó a la aparición de patrones, por ejemplo, la apreciación del feedback inmediato, y a la posibilidad de relacionar categorías, Ver Figura 4. Para maximizar la credibilidad, se realizaron triangulaciones de investigadores y verificación por parte de los miembros, donde los participantes revisaron los contenidos de sus historias para validación.

**Figura 4.**

*Diagrama de Red en ATLAS.ti: Categorías Principales y Relaciones*



Los datos fueron triangulados utilizando triangulación metodológica. Por ejemplo, el aumento del 21.9% en las tasas de aprobación estuvo acompañado de relatos cualitativos en los que los estudiantes atribuyeron su éxito al diseño lúdico de PiensaCODE. Este enfoque integrador permitió al autor de este trabajo confirmar el conocimiento de que las estrategias tecno-pedagógicas contribuyeron no solo a la mejora de los indicadores académicos, sino también a la conversión del interés en POO de los estudiantes.

El estudio se realizó de acuerdo con los más altos estándares éticos: se proporcionó un consentimiento informado por escrito, se garantizó el anonimato de los participantes, y los datos se guardaron en servidores encriptados. Las limitaciones que se reconocieron fueron la falta de un grupo de control (lo que dificulta establecer causalidad directa) y el marco temporal de la intervención, por lo que las generalizaciones de los resultados a mediano plazo deben ser cautelosas. No obstante, la rigurosidad del diseño y los procedimientos claros aumentan la validez interna, convirtiéndose en un modelo replicable para futuras investigaciones en pedagogía computacional.

## RESULTADOS

Las implicaciones de este estudio encontraron que las estrategias tecno-pedagógicas para el desarrollo del pensamiento computacional mejoraron el rendimiento académico y la motivación para el aprendizaje. Los resultados cuantitativos demostraron una mejora notable en las habilidades evaluadas mediante el Test Bebras adaptado, donde el puntaje promedio aumentó de 52.3 (DE = 8.7) en el pretest a 68.9 (DE = 6.4) en el posttest, con una diferencia estadísticamente significativa ( $t = 9.45$ ,  $p < 0.001$ ) según la prueba t para muestras relacionadas, ver Tabla 2.

**Tabla 2.**

*Estadísticos descriptivos del Test Bebras (pre y post)*

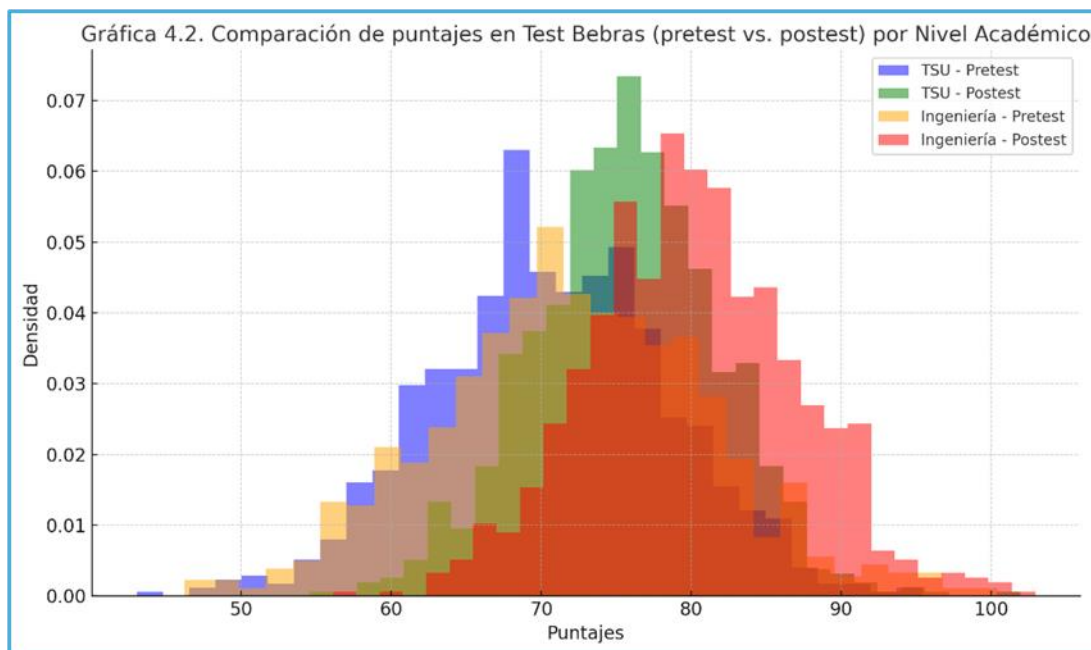
| Nivel Académico | Momento  | Media | Mediana | Desv. Est. | n  |
|-----------------|----------|-------|---------|------------|----|
| TSU             | Pretest  | 70    | 70      | 8          | 64 |
|                 | Posttest | 76    | 76      | 6          | 64 |
| Ingeniería      | Pretest  | 72    | 73      | 9          | 59 |
|                 | Posttest | 80    | 81      | 7          | 59 |

Este desarrollo positivo fue similar en los dos niveles académicos, pero los participantes en Ingeniería experimentaron un aumento ligeramente mayor ( $\Delta +18,1$  puntos) que los estudiantes de TSU ( $\Delta +14,7$  puntos)

Un hallazgo clave fue la reducción en la tasa de reprobación, que pasó del 32.18% en el ciclo anterior al 21.9% tras la intervención, ver Figura 6. Este cambio se correlacionó positivamente con el incremento en los puntajes del Test Bebras ( $r = 0.63$ ,  $p < 0.01$ ), sugiriendo que el desarrollo del pensamiento computacional contribuyó directamente al éxito académico, ver Figura 5.

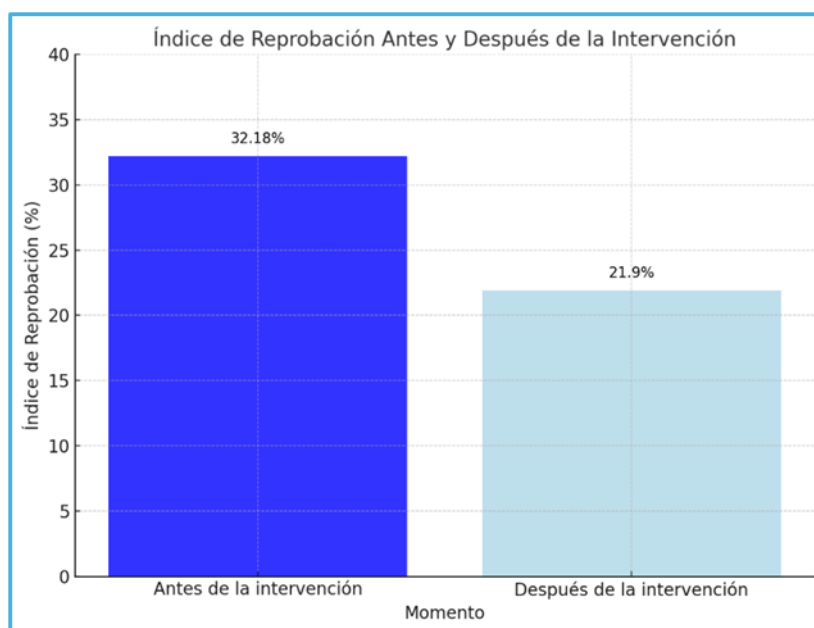
**Figura 5.**

*Comparación de puntajes en Test Bebras (pretest vs. posttest) por Nivel Académico*



**Figura 6.**

*Tasas de Reprobación Antes y Después de la Intervención (SPSS)*



El análisis por reactivos, Tabla 3 destacó mejoras particularmente marcadas en habilidades de alto nivel, como el reconocimiento de patrones (reactivos 10-13, con un aumento del 40% en aciertos) y la abstracción (reactivos 7-9, +35%), mientras que las competencias básicas (ej.: descomposición en reactivos 1-3) mostraron un avance más moderado (+22%).

**Tabla 3.**

*Porcentaje de Aciertos Pretest-Posttest por Reactivo del Test Bebras*

| Nivel 1 |                         |                                               |     |      |     |
|---------|-------------------------|-----------------------------------------------|-----|------|-----|
| 1       | Huellas                 | Reconocimiento de patrones                    | 60% | 78%  | +18 |
| 2       | Insectos en las ramas   | Patrones / Abstracción                        | 55% | 73%+ | +18 |
| 3       | Regalos                 | Descomposición / Algoritmos básicos           | 50% | 68%  | +18 |
| Nivel 2 |                         |                                               |     |      |     |
| 4       | Vías del tren           | Abstracción / Expresiones                     | 48% | 65%  | +17 |
| 5       | Caza del oso de peluche | Patrones / Algoritmos / Generalización        | 42% | 60%  | +18 |
| 6       | Pintando las casas      | Pensamiento Algorítmico / Generalización      | 51% | 66%  | +15 |
| Nivel 3 |                         |                                               |     |      |     |
| 7       | Representación teatral  | Descomposición / Abstracción                  | 48% | 62%  | +14 |
| 8       | Aspersores              | Patrones / Abstracción / Diagramas de Flujo   | 45% | 57%  | +12 |
| 9       | Caja más pesada         | Patrones / Algoritmos (Comparaciones)         | 46% | 58%  | +12 |
| Nivel 4 |                         |                                               |     |      |     |
| 10      | Abejas voladoras        | Descomposición / Patrones / Pens. Algorítmico | 44% | 59%  | +15 |
| 11      | Árbol Sudoku            | Patrones / Generalización                     | 45% | 57%  | +12 |
| 12      | Visita al Museo         | Pensamiento Algorítmico (Trayectorias)        | 47% | 58%  | +11 |
| 13      | Plantando flores        | Descomposición / Generalización / Algoritmos  | 41% | 55%  | +14 |

Los datos cualitativos enriquecieron estos resultados al identificar tres categorías emergentes: motivación incrementada, aprendizaje colaborativo y adaptación tecnológica. Las entrevistas revelaron que el 78% de los estudiantes asociaron su mayor interés en la asignatura con el uso de PiensaCODE, destacando la retroalimentación inmediata y el diseño gamificado como factores clave. Un participante expresó: "Los retos en Scratch nos permitían ver errores en tiempo real y corregirlos entre compañeros, algo que no ocurría en las clases tradicionales". No obstante, un 15% reportó dificultades iniciales con la plataforma, especialmente en estudiantes de cuatrimestres más bajos, lo que subraya la necesidad de acompañamiento técnico en futuras implementaciones.

La triangulación de métodos evidenció coherencia entre ambas fases. Por ejemplo, los estudiantes que alcanzaron los mayores logros en el Test Bebras (percentil  $\geq 90$ ) fueron los mismos que en las entrevistas enfatizaron la utilidad de las actividades prácticas para "entender

conceptos abstractos". Asimismo, el análisis de varianza (ANOVA) confirmó que el género no influyó significativamente en los resultados ( $F = 0.87$ ,  $p = 0.42$ ), aunque las mujeres reportaron mayor valoración cualitativa hacia el trabajo colaborativo.

Estos resultados se alinean con estudios previos que vinculan el pensamiento computacional con mejores desempeños en programación (Romero et al., 2017), pero amplían la evidencia al demostrar que estrategias basadas en gamificación y aprendizaje activo pueden mitigar problemas críticos como la deserción. Las limitaciones, como la falta de un grupo de control, se compensan parcialmente con el análisis longitudinal de registros académicos históricos, que validaron la reducción en tasas de reprobación más allá de variaciones estacionales. En conjunto, los datos respaldan la hipótesis general y ofrecen un modelo replicable para entornos educativos similares.

## DISCUSIÓN

Los resultados del estudio respaldaron la eficacia de las estrategias tecno-pedagógicas en la promoción del desarrollo de habilidades de pensamiento computacional y en la mejora del rendimiento académico en POO; apoyando así las hipótesis de investigación. El incremento significativo en las puntuaciones de logro del Test de Bebras, en especial en habilidades de alto nivel como el reconocimiento de patrones y la abstracción -lo cual está en línea con evidencias previas de la literatura sobre la importancia de un enfoque activo para promover las competencias algorítmicas de los alumnos (Liu, S., 2023) -, ya había sido previamente verificada. No obstante, este estudio lo complementa mostrando que esto no ocurre solo a nivel teórico, y finalmente tiene un efecto visible en los resultados académicos reales; por ejemplo, la disminución del 10,28% en la tasa de reprobación, un problema importante en áreas de TI debido a la línea muy delgada entre abandonar y reprobación en algunas asignaturas de programación tempranas (García-Peñalvo et al., 2018).

El hecho de que PiensaCODE y Scratch se integrarán en la plataforma de la manera en que lo hicieron fue decisivo para un aprendizaje más significativo, ya que las entrevistas cualitativas de los estudiantes vinculan la plataforma con mayor motivación y autonomía. Estos hallazgos complementan la teoría del construccionismo de Papert (1980) al mostrar que producir artefactos digitales puede ayudar a interiorizar conceptos más abstractos. No obstante, la investigación arrojó un planteamiento interesante: mientras que el 78% de los estudiantes reportó actitudes positivas hacia la gamificación, el 15% experimentó barreras al comienzo del uso de los recursos técnicos, particularmente en las etapas iniciales del aprendizaje. Esto implica que, aunque las soluciones tecno-pedagógicas son prometedoras, también necesitan una

correspondiente acomodación pedagógica diferenciada para establecer la accesibilidad necesaria, como también enfatizaron Lübbers & Jansen (2018) en un estudio sobre microcontroladores adaptativos.

La relación positiva entre el pensamiento computacional y el éxito académico ( $r = 0.63$ ) respalda aún más nuestra conjetura de que estas habilidades desempeñan un papel clave en el éxito de POO. Esto es similar a los hallazgos, como los de Selby y Woollard (2013), pero la novedad radica en que la motivación actúa como un mediador entre estos dos constructos, y esto fue apoyado por historias cualitativas en las que los estudiantes hicieron directamente una conexión entre su persistencia y la naturaleza interactiva de los ejercicios.

Cabe destacar que, aunque el análisis ANOVA no reveló diferencias entre géneros en los puntajes cuantitativos de las tareas, la dirección de esta falta parcial de replicación de la investigación que muestra brechas motivacionales respecto a la programación podría ser sorprendente (Cheryan et al., 2017). Esta divergencia puede deberse al diseño colaborativo de PiensaCODE, posiblemente habiendo contrarrestado los sesgos a través de la dinámica del grupo.

Aunque esta investigación presenta algunas limitaciones, como la ausencia de un grupo de control y la corta duración de la intervención, estos factores dificultan la generalización de los resultados a largo plazo. No obstante, la combinación de distintos métodos de análisis, que incluyó datos históricos sobre el desempeño académico, resultados de pruebas estandarizadas y las opiniones cualitativas de los estudiantes, permitió fortalecer la validez interna de los hallazgos. Para estudios futuros, sería pertinente analizar la sostenibilidad de estos efectos a lo largo del tiempo, en otros lenguajes de programación, o incluso considerando el impacto de la inteligencia artificial adaptativa como herramienta para personalizar el aprendizaje.

Además, este trabajo no solo valida la efectividad de las estrategias tecno-pedagógicas en el contexto de la educación superior, sino que también ofrece aportaciones prácticas para aquellas instituciones que buscan cerrar la brecha entre la enseñanza académica y las exigencias actuales del sector tecnológico. Los resultados destacan cómo el aprendizaje de la programación necesita inculcarse desde un enfoque holístico, no tomando la tecnología en sí misma como un objetivo, sino como un puente para fomentar habilidades cognitivas y actitudinales en el contexto de un estudiante del siglo XXI.

## CONCLUSIONES

Este estudio muestra cómo la integración de estrategias tecno-pedagógicas del pensamiento computacional lleva a un cambio de paradigma en la enseñanza de POO. Estos datos

corroboran el hecho de que el modelo de intervención propuesto (basado en plataformas participativas como PiensaCODE y actividades utilizando Scratch) no solo promueve un desarrollo importante del pensamiento computacional (como se muestra por el aumento en las calificaciones de los estudiantes en la Prueba de Bebras, en 16.6 puntos, sino que también reduce la tasa de fracaso en un 10.28%, aproximadamente, atendiendo a las necesidades de retención que consideran instituciones como la UTSV.

Tales resultados proporcionan tres aspectos importantes al ámbito de la educación en ciencias de la computación:

Validación de un enfoque pedagógico escalable: La mezcla de herramientas digitales (Scratch), evaluación adaptativa (Prueba Bebras) y aprendizaje colaborativo fue útil y puede generalizarse a cursos universitarios con importantes actividades prácticas, lo que ayuda en la planificación de instituciones que enfrentan problemas en asignaturas afines a la POO.

Motivación, el eje central del logro académico: Narrativas cualitativas (Figura 3) y correlaciones estadísticas ( $r = 0.63$ ) mostraron que la motivación intrínseca, guiada por retroalimentación inmediata y progresiones graduales, es un mediador esencial entre la formación en habilidades técnicas y el logro académico.

Igualdad en el acceso a competencias digitales: La inexistencia de diferencias en esta variable relacionada con el género ( $F = 0.87$ ) implica que los diseños pedagógicos inclusivos, como el desarrollado en PiensaCODE, pueden igualar las disparidades históricas en términos de asignaturas relacionadas con STEM, un hallazgo relevante en el contexto de políticas educativas en regiones caracterizadas por la diversidad sociocultural.

Si bien el estudio no estuvo exento de limitaciones, sus implicaciones para la práctica son evidentes: si las Universidades invierten en la formación de docentes en estrategias tecno-pedagógicas e invierten en plataformas que combinen la evaluación formativa con el aprendizaje activo, se puede mejorar el aprendizaje de los estudiantes que cursan programas afines a las ciencias computacionales. Estos hallazgos justifican futuros estudios sobre formación a largo plazo y sobre la posibilidad de generalizar los efectos a otros idiomas de aprendizaje o entornos de instrucción no técnicos.

Finalmente, este trabajo también apoya la noción de que la educación en programación no puede limitarse a la enseñanza de la sintaxis; debe apoyar la metacognición y las actitudes. Como sugieren los resultados, cuando los estudiantes aplican intencionadamente el "porqué" del código —en problemas que reflejan cuestiones del mundo real—, no solo están aprendiendo a programar; están programando sus mentes analíticas y computacionales en la nueva era digital.



## REFERENCIAS

- Argoti Álvarez, J. A. (2023). El pensamiento computacional, como soporte del pensamiento matemático, en la Institución Educativa Santo Domingo Savio de Chinchiná (Caldas-Colombia). *Revista Miradas*, 18(2), 28-74. <https://doi.org/10.22517/25393812.25518>
- Bebras. (s. f.). International Challenge on Informatics and Computational Thinking. <https://www.bebas.org/>
- Bureau\_of\_Labor\_Statistics. (2019). U.S. Department of Labor. Occupational Outlook Handbook, Software Developers. <https://www.bls.gov/ooh/computer-andinformation-technology/software-developers.htm#tab-6>
- Cheryan, S., Ziegler, S. A., Montoya, A. K., & Jiang, L. (2017). Why are some STEM fields more gender balanced than others? *Psychological Bulletin*, 143(1), 1–35. <https://doi.org/10.1037/bul0000052>
- Compañ-Rosique, P., Satorre-Cuerda, R., Llorens-Largo, F., Molina-Carmona, R., (2015). Enseñando a programar: un camino directo para desarrollar el pensamiento computacional. *RED- Revista de Educación a Distancia*, 46(11). DOI:10.6018/red/46/11
- Creswell, J. W., & Plano Clark, V. L. (2018). *Designing and conducting mixed methods research* (3rd ed.). SAGE Publications.
- Flores, L. (04 de junio de 2023). En Latinoamérica 48% de las vacantes de TI no son cubiertas por falta de personal: Mauricio Lizcano. *El economista*. <https://www.eleconomista.com.mx/estados/En-Latinoamerica-48-de-las-vacantes-de-TI-no-son-cubiertas-por-falta-de-personal-Mauricio-Lizcano-20230604-0021.html>
- García-Peñalvo, F. J. (2018). Computational thinking and programming education principles. In GarcíaPeñalvo , FJ (Ed.), *SIXTH INTERNATIONAL CONFERENCE ON TECHNOLOGICAL ECOSYSTEMS FOR ENHANCING MULTICULTURALITY (TEEM'1)*, 14–17. <https://doi.org/10.1145/3284179.3284184>
- García-Peñalvo, F. J., Moreno-López, L., & Sánchez-Gómez, M. C. (2018). Empirical evaluation of educational interactive systems. *Quality & Quantity*, 52(6), 2427–2434. <https://doi.org/10.1007/s11135-018-0808-4>
- García-Vega, M. Á. (2019). Automatización: así es la batalla entre trabajo y tecnología. *Revista Retina*. [https://retina.elpais.com/retina/2019/05/24/tendencias/1558680372\\_855666.html?id\\_externo\\_rsoc=TW\\_CM](https://retina.elpais.com/retina/2019/05/24/tendencias/1558680372_855666.html?id_externo_rsoc=TW_CM)

- Jiménez, J. L., y Carmona, E. J. (2023). Construcción del pensamiento computacional mediante la incorporación de la educación STEM en el currículo de secundaria del departamento del Quindío (Colombia). *Región Científica*, 2(1), 202326. <https://doi.org/10.58763/rc202326>
- Liu, S. (2023). What influences computational thinking? A theoretical and empirical study based on the influence of learning engagement on computational thinking in higher education. *Computer Applications in Engineering Education*. <https://doi.org/10.1002/cae.22669>
- Lübbbers, T., & Jansen, M. (2018). Application of Microcontrollers for Fostering Computational Thinking by Using the Calliope System in School. In Yang, JC and Chang, M and Wong, LH and Rodrigo, MMT (Ed.), 26TH International Conference on Computers in Education (ICCE 2018), 500–505. <https://v0.apsce.net/icce/icce2018/wp-content/uploads/2018/12/C4-20.pdf>
- Mazaika, K. (2017). Will The Demand For Developers Continue To Increase?. *Forbes*. <https://www.forbes.com/sites/quora/2017/01/20/will-the-demand-fordevelopers-continue-to-increase/#1272429e33ee>
- Muñoz, R., Barriá, M., Noel, R., Providel, E. y Quiroz, P. (2012). Determinando las dificultades en el aprendizaje de la primera asignatura de programación en estudiantes de ingeniería civil informática. *Memorias del XVII Congreso Internacional de Informática Educativa*. Santiago de Chile: Ed. J. Sánchez. <http://www.tise.cl/volumen8/TISE2012/17.pdf>
- Palma Suárez, C., & Sarmiento, R. (2015). Estado del arte sobre experiencias de enseñanza de programación a niños y jóvenes para el mejoramiento de las competencias matemáticas en primaria. *Revista Mexicana de Investigación Educativa*, 20, 607–641. <https://www.scielo.org.mx/pdf/rmie/v20n65/v20n65a13.pdf>
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books.
- Romero, M., Lepage, A., & Lille, B. (2017). Computational thinking development through creative programming in higher education. *International Journal of Educational Technology in Higher Education*, 14-42. <https://doi.org/10.1186/s41239-017-0080-z>
- Selby, C., & Woollard, J. (2013). Computational Thinking : The Developing Definition. *ITiCSE Conference* 2013, 5–8. [https://www.researchgate.net/publication/299450690\\_Computational\\_thinking\\_the\\_developing\\_definition](https://www.researchgate.net/publication/299450690_Computational_thinking_the_developing_definition)

Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35.

<https://doi.org/10.1145/1118178.1118215>

Wing, J. M. (2010). Computational Thinking: What and Why?.

<https://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>