

PROGRAMACIÓN VISUAL Y ESTRATEGIAS LÚDICAS, UNA VÍA PARA EL DESARROLLO DEL PENSAMIENTO COMPUTACIONAL EN LA INFANCIA

Visual Programming and Playful Strategies, A Pathway to the Development of Computational Thinking in Childhood

Programação visual e estratégias lúdicas, um caminho para o desenvolvimento do pensamento computacional na infância

MS.c. Walfrido Camué Ortiz, <https://orcid.org/0000-0002-4820-9423>

Dr. C. Oscar García Fernández, <https://orcid.org/0000-0002-4820-9423>

Dr. C. José Manuel Izquierdo Pardo, <https://orcid.org/0000-0003-1159-5233>

Universidad de Oriente, Cuba

*Autor para correspondencia. email walfrido.camue@uo.edu.cu

Para citar este artículo: Camué Ortiz, W., García Fernández, O. e Izquierdo Pardo, J. M. (2026). Aplicación educativa para la introducción a la asignatura Enfermería Ginecobstétrica. *Maestro y Sociedad*, 23(2), 1879-1886. <https://maestroysociedad.uo.edu.cu>

RESUMEN

Introducción: La enseñanza de la programación en edades tempranas constituye una prioridad en el Tercer Perfeccionamiento de la Educación Cubana, lo que exige metodologías que integren el pensamiento computacional con estrategias lúdicas. Objetivo: Evaluar el impacto de una metodología pedagógica basada en el aprendizaje basado en juegos y la modificación de mecánicas de juego mediante programación visual, para enseñar fundamentos de programación y desarrollar el pensamiento computacional en niños de 9 a 11 años. Materiales y métodos: Se implementó un estudio de caso cualitativo con 30 niños (15 niñas y 15 niños) sin experiencia previa en programación, quienes participaron en un curso intensivo de dos semanas en la Universidad de Oriente, Cuba. La intervención, estructurada en tres fases (análisis guiado, reproducción y modificación, y creación libre), utilizó Scratch 3.0 como herramienta. Los datos se recolectaron mediante observación participante no estructurada, lista de verificación, portafolio digital y registro diario; se procesaron mediante análisis temático. Resultados: Se evidenció un alto nivel de motivación sostenida, comprensión aplicada de conceptos de programación (secuencias, bucles, condicionales), desarrollo del pensamiento computacional (descomposición, reconocimiento de patrones, abstracción), equidad de género en el desempeño y fortalecimiento del trabajo colaborativo. Conclusiones: La metodología propuesta constituye una innovación pedagógica efectiva, replicable y recomendable para su integración en el currículo escolar cubano, siempre que se acompañe de formación docente específica.

Palabras clave: aprendizaje basado en juegos; programación; gamificación; pensamiento computacional; Scratch.

ABSTRACT

Introduction: Teaching programming at an early age is a priority in the Third Improvement of Cuban Education, which requires methodologies that integrate computational thinking with playful strategies. Objective: To evaluate the impact of a pedagogical methodology based on game-based learning and the modification of game mechanics through visual programming, to teach programming fundamentals and develop computational thinking in children aged 9 to 11 years. Materials and methods: A qualitative case study was implemented with 30 children (15 girls and 15 boys) with no prior programming experience, who participated in a two-week intensive course at the University of Oriente, Cuba. The intervention, structured in three phases (guided analysis, reproduction and modification, and free creation), used Scratch 3.0 as a tool. Data were collected through unstructured participant observation, a checklist, a digital portfolio, and a daily log; they were processed through thematic analysis. Results: A high level of sustained motivation, applied understanding of programming concepts (sequences, loops, conditionals), development of computational thinking (decomposition, pattern recognition, abstraction), gender equity in performance, and strengthening of collaborative work were evidenced.

Conclusions: The proposed methodology constitutes an effective, replicable, and recommended pedagogical innovation for integration into the Cuban school curriculum, provided it is accompanied by specific teacher training.

Keywords: game-based learning; programming; gamification; computational thinking; Scratch.

RESUMO

Introdução: O ensino de programação na primeira infância é uma prioridade no Terceiro Aprimoramento da Educação Cubana, o que exige metodologias que integrem o pensamento computacional com estratégias lúdicas. Objetivo: Avaliar o impacto de uma metodologia pedagógica baseada na aprendizagem baseada em jogos e na modificação de mecânicas de jogo por meio da programação visual, para ensinar fundamentos de programação e desenvolver o pensamento computacional em crianças de 9 a 11 anos. Materiais e métodos: Foi implementado um estudo de caso qualitativo com 30 crianças (15 meninas e 15 meninos) sem experiência prévia em programação, que participaram de um curso intensivo de duas semanas na Universidade de Oriente, Cuba. A intervenção, estruturada em três fases (análise orientada, reprodução e modificação, e criação livre), utilizou o Scratch 3.0 como ferramenta. Os dados foram coletados por meio de observação participante não estruturada, lista de verificação, portfólio digital e registro diário; foram processados por meio de análise temática. Resultados: Evidenciou-se alto nível de motivação sustentada, compreensão aplicada de conceitos de programação (sequências, loops, condicionais), desenvolvimento do pensamento computacional (decomposição, reconhecimento de padrões, abstração), equidade de gênero no desempenho e fortalecimento do trabalho colaborativo. Conclusões: A metodologia proposta constitui uma inovação pedagógica efetiva, replicável e recomendável para integração no currículo escolar cubano, desde que acompanhada de formação docente específica.

Palavras-chave: aprendizagem baseada em jogos; programação; gamificação; pensamento computacional; Scratch.

Recibido: 5/2/2026 Aprobado: 28/3/2026

INTRODUCCIÓN

La educación en el siglo XXI enfrenta el desafío de formar a las nuevas generaciones en competencias que trasciendan los contenidos tradicionales. En un entorno cada vez más digitalizado, habilidades como la resolución de problemas, el pensamiento lógico y la creatividad se han convertido en ejes fundamentales para el desarrollo integral de los estudiantes.

En este contexto, la enseñanza de la programación desde edades tempranas ha dejado de ser una especialidad técnica para convertirse en una necesidad pedagógica, no solo para formar futuros profesionales en ciencias de la computación, sino para dotar a todos los niños de herramientas cognitivas que les permitan comprender y transformar la realidad que los rodea.

El concepto de pensamiento computacional, acuñado por Wing (2006), ha sido clave en este giro educativo. Este término hace referencia a un conjunto de habilidades que incluyen la descomposición de problemas complejos, el reconocimiento de patrones, la abstracción y el diseño de algoritmos.


Lejos de limitarse al ámbito de la informática, el pensamiento computacional se concibe como una competencia transversal aplicable a múltiples disciplinas y situaciones cotidianas. Su desarrollo en la infancia no solo prepara a los niños para un futuro tecnológico incierto, sino que también potencia su capacidad para organizar ideas, tomar decisiones fundamentadas y trabajar de manera colaborativa.

En Cuba, esta visión ha encontrado eco en las políticas educativas actuales. El Tercer Perfeccionamiento de la Educación, proceso que busca actualizar y mejorar el sistema nacional de enseñanza, ha incorporado contenidos de programación y robótica desde los primeros niveles escolares.

Esta decisión responde a la necesidad de alinear la formación de los estudiantes con las demandas del desarrollo científico-técnico del país y con las tendencias internacionales en materia educativa. No obstante, su implementación enfrenta un reto significativo: ¿cómo enseñar conceptos abstractos de programación a niños de primaria de manera accesible, motivadora y efectiva?

La respuesta a este interrogante ha generado un creciente interés en metodologías activas que sitúen al estudiante en el centro del proceso de aprendizaje. Entre ellas, el aprendizaje basado en juegos (game-based learning) y la gamificación han demostrado un alto potencial para fomentar la motivación intrínseca, la participación activa y la perseverancia ante tareas complejas (Deterding et al., 2011).

Estas estrategias se basan en el reconocimiento de que el juego constituye una actividad natural en la infancia, a través de la cual los niños exploran, experimentan y construyen significados. Cuando se integra con



la programación, el juego deja de ser solo un vehículo recreativo para convertirse en un auténtico laboratorio de aprendizaje.

Paralelamente, el desarrollo de entornos de programación visual ha abierto nuevas posibilidades didácticas. Herramientas como Scratch (Resnick et al., 2009) han revolucionado la enseñanza de la programación al eliminar la barrera sintáctica de los lenguajes textuales.

Su interfaz basada en bloques permite a los estudiantes concentrarse en la lógica y la estructura algorítmica, sin la frustración que a menudo acompaña al aprendizaje de lenguajes como C++ o Java. Scratch ha sido ampliamente utilizado en contextos formales e informales para introducir conceptos como secuencias, bucles, condicionales, variables y eventos, facilitando al mismo tiempo el desarrollo de la creatividad y la expresión personal.

Sin embargo, la sola presencia de una herramienta tecnológica no garantiza el aprendizaje efectivo. La investigación educativa ha señalado que el éxito de estas iniciativas depende en gran medida del diseño pedagógico que las acompañe (Brennan & Resnick, 2012).

En este sentido, la innovación no reside únicamente en el uso de Scratch, sino en la manera en que se estructura la experiencia de aprendizaje. Una estrategia particularmente prometedora consiste en involucrar a los estudiantes en la modificación deliberada de mecánicas de juegos existentes, un enfoque que combina el análisis crítico, la experimentación y la creación.

Este enfoque, que los autores denominan “innovación deliberada en mecánicas de juego”, propone que los niños no aprenden a programar a partir de ejercicios abstractos, sino mediante la exploración de juegos funcionales y la introducción de cambios intencionales en su comportamiento.

Al hacerlo, los estudiantes se ven obligados a comprender la lógica subyacente, identificar las relaciones causa-efecto entre los bloques de código y prever las consecuencias de sus modificaciones. Este proceso activo de descomposición y recombinación constituye un entorno privilegiado para el desarrollo del pensamiento computacional.

A pesar de las evidencias disponibles sobre el potencial de Scratch y las metodologías lúdicas, persisten interrogantes sobre su aplicabilidad en contextos educativos concretos, especialmente en países con recursos limitados o con sistemas educativos centralizados como el cubano.

La literatura internacional ofrece numerosos estudios que avalan la eficacia de estas herramientas en entornos experimentales o de investigación (Grover & Pea, 2013; Maloney et al., 2010; Lozano & García-Peñalvo, 2021), pero son menos frecuentes las investigaciones que documentan experiencias sistemáticas en el contexto latinoamericano y, particularmente, en Cuba.

Por otra parte, un aspecto crucial que ha sido escasamente abordado en la investigación sobre enseñanza de la programación en la infancia es la equidad de género. Diversos estudios han señalado que, desde edades tempranas, pueden generarse sesgos que desalientan la participación de las niñas en actividades tecnológicas (Stewart, 2023).

Sin embargo, cuando se utilizan metodologías centradas en la colaboración, la creatividad y el juego, estos estereotipos tienden a diluirse. La presente investigación presta especial atención a esta dimensión, analizando si el enfoque propuesto favorece una participación equitativa entre niños y niñas.

El presente artículo se inscribe en esta línea de trabajo. Su objetivo principal es evaluar el impacto de una metodología pedagógica basada en el aprendizaje basado en juegos y la modificación de mecánicas de juego mediante programación visual, para enseñar fundamentos de programación y desarrollar el pensamiento computacional en niños de 9 a 11 años, en el marco del Tercer Perfeccionamiento de la Educación Cubana.

La investigación se sustenta en una experiencia práctica concreta: un curso intensivo de verano diseñado y ejecutado por los autores en la Universidad de Oriente, Cuba. Este curso sirvió como laboratorio pedagógico para validar la propuesta metodológica en condiciones reales, con niños sin experiencia previa en programación.

Con esta investigación se pretende demostrar que la enseñanza de la programación no es privativa de carreras con perfil informático ni requiere necesariamente de docentes especializados en ingeniería de software.

Por el contrario, cuando se emplean métodos de enseñanza adecuados, niños sin formación previa pueden aprender a programar con éxito, desarrollando simultáneamente habilidades de pensamiento lógico, creatividad y colaboración. Scratch, en este contexto, se convierte en una herramienta facilitadora que permite

construir nuevos conocimientos de manera divertida y significativa.

MATERIALES Y MÉTODOS

Se adoptó un diseño cualitativo de estudio de caso (Yin, 2018), centrado en un grupo específico de estudiantes en un contexto real, lo que permitió un análisis profundo y contextualizado de las dinámicas de aprendizaje.

Este diseño resultó pertinente para captar la complejidad de los procesos de enseñanza y aprendizaje en un entorno natural, así como para comprender las percepciones y experiencias de los participantes.

Participantes: La muestra estuvo conformada por 30 niños (15 niñas y 15 niños) de 9 a 11 años de edad, sin experiencia previa en programación, seleccionados de forma voluntaria a través de una convocatoria abierta. Se agruparon en seis equipos de cinco integrantes cada uno, procurando un equilibrio de género en cada grupo. Esta conformación respondió al propósito de fomentar la colaboración y observar las dinámicas de trabajo mixto.

Contexto: La intervención se desarrolló en el Laboratorio de Informática de la Sede "Julio Antonio Mella" de la Universidad de Oriente, Cuba, durante un curso intensivo de verano con una duración de dos semanas, con sesiones diarias de cuatro horas.

Este entorno universitario ofreció las condiciones tecnológicas adecuadas (computadoras con acceso a Scratch 3.0 y conectividad básica) y un espacio ajeno al aula tradicional, lo que contribuyó a generar un ambiente de aprendizaje menos formal y más propicio para la experimentación.

Herramienta: Se utilizó la plataforma Scratch 3.0, seleccionada por su idoneidad para la enseñanza infantil, su interfaz gráfica intuitiva basada en bloques y su alineamiento con las directrices del Ministerio de Educación cubano para la introducción de contenidos de programación en la enseñanza primaria.

Procedimiento: La intervención se estructuró en tres fases secuenciales, con una progresión gradual en complejidad y autonomía. En la primera fase, análisis guiado (días 1 a 3), los estudiantes analizaron en conjunto videojuegos simples creados previamente por el investigador principal.

Se les guió mediante preguntas orientadoras para identificar las mecánicas de juego (puntuación, vidas, colisiones, niveles) y los bloques de código que las implementaban. Se promovió la exploración libre de los proyectos y la formulación de hipótesis sobre el funcionamiento de cada bloque.

En la segunda fase, reproducción y modificación (días 4 a 7), se asignó a cada equipo la tarea de reproducir el juego analizado y luego modificar al menos una de sus mecánicas. Las modificaciones podían consistir en cambiar la forma de ganar puntos, añadir un nuevo obstáculo, modificar la velocidad de los objetos o incorporar nuevos personajes. Esta fase requirió que los estudiantes aplicaran los conceptos comprendidos y resolvieran problemas emergentes.

En la tercera fase, creación libre (días 8 a 10), los estudiantes, organizados por equipos, crearon un juego original, aplicando las mecánicas y conceptos aprendidos. Contaron con asesoría puntual del docente, pero la toma de decisiones sobre el tema, las reglas y la estructura del juego fue completamente autónoma. Culminaron con una presentación de sus proyectos ante el resto de los grupos.

Instrumentos de evaluación: Se emplearon múltiples instrumentos para asegurar la triangulación de los datos. La lista de verificación permitió verificar de manera eficiente el desempeño individual y grupal en ítems como participación activa, capacidad de explicar conceptos, colaboración en el equipo y cumplimiento de tareas, aplicándose al final de cada sesión.

El portafolio digital recopiló evidencias tangibles del trabajo y progreso a lo largo de las dos semanas, incluyendo capturas de pantalla de los proyectos en desarrollo, fotografías del trabajo en equipo, videos de las presentaciones y los archivos finales de los juegos creados.

El registro diario complementó el sistema de evaluación con observaciones cualitativas y notas contextuales sobre logros destacados, dificultades observadas, interacciones significativas entre los participantes y planes de acción inmediatos para las sesiones siguientes.

Recolección y análisis de datos: La principal técnica de recolección fue la observación participante no estructurada, en la que los investigadores actuaron como facilitadores y observadores, registrando en los registros diarios las dinámicas de cada sesión.

Se realizó un análisis temático (Braun & Clarke, 2006) de las observaciones, los registros y los productos finales, siguiendo un proceso de codificación abierta, identificación de patrones recurrentes y construcción de categorías temáticas. Este proceso permitió sintetizar los hallazgos en torno a las dimensiones clave del aprendizaje: motivación, comprensión conceptual, desarrollo del pensamiento computacional, trabajo colaborativo y equidad de género.

RESULTADOS

Del análisis temático de las observaciones, los registros diarios y los proyectos finales surgieron cinco categorías principales que sintetizan los hallazgos del estudio. En la Tabla 1 se presentan de forma resumida los indicadores observados en cada categoría.

Tabla 1 Categorías emergentes e indicadores de aprendizaje

Categoría	Indicadores observados
Motivación y compromiso	Asistencia perfecta, participación activa, dedicación en horario extra, entusiasmo sostenido
Comprensión aplicada de conceptos de programación	Uso adecuado de secuencias, bucles, condicionales; explicación oral de la lógica implementada
Desarrollo del pensamiento computacional	Descomposición de tareas, reutilización de bloques, abstracción de comportamientos
Equidad de género	Participación equitativa en tareas técnicas y de liderazgo; ausencia de sesgos en la distribución de roles
Fortalecimiento del trabajo colaborativo	Ayuda mutua, resolución conjunta de problemas, celebración colectiva de logros

Fuente: Elaboración propia a partir de los instrumentos aplicados.

Alto nivel de motivación y compromiso: Todos los participantes mostraron entusiasmo sostenido durante las dos semanas. La asistencia fue perfecta, y en varias ocasiones los niños solicitaron permanecer en el laboratorio durante el horario de descanso para continuar trabajando en sus proyectos.

La posibilidad de interactuar con juegos, modificar sus reglas y crear sus propios proyectos fue percibida como una actividad lúdica y significativa, alejada de la rutina escolar tradicional. Un comentario recurrente entre los participantes fue: “parece que estamos jugando, pero estamos aprendiendo”.

Comprensión aplicada de conceptos de programación: Los estudiantes no se limitaron a reproducir los juegos modelo, sino que demostraron capacidad para descomponer sus mecánicas en estructuras lógicas y recombinarlas de forma original.

Por ejemplo, un equipo transformó un juego de evasión de obstáculos en uno de recolección de objetos, modificando tanto la lógica de puntuación (de “perder puntos al tocar” a “ganar puntos al recoger”) como la interacción entre sprites. Se observó que, al finalizar la intervención, el 100 % de los estudiantes podía explicar oralmente qué hacían los bloques utilizados y por qué los habían elegido.

Desarrollo del pensamiento computacional: Los estudiantes evidenciaron habilidades claras en tres dimensiones clave. En la descomposición, dividieron sus juegos en componentes funcionales (movimiento, colisiones, puntuación, niveles) y trabajaron en ellos de manera secuencial.

En el reconocimiento de patrones, reutilizaron bloques de código similares en distintas partes del proyecto, reconociendo estructuras repetitivas que podían resolverse con bucles. En la abstracción, comprendieron que ciertos bloques (por ejemplo, “si toca borde, rebota”) representaban acciones independientes del personaje específico, lo que les permitió transferir soluciones entre diferentes contextos.

Equidad de género en el desempeño y la participación: No se observaron diferencias significativas entre niñas y niños en términos de creatividad, dominio técnico, liderazgo en los equipos o resolución de problemas. Ambos géneros participaron activamente en todas las fases del proceso.

En los equipos mixtos, se observó una distribución natural de tareas sin sesgos de género: en algunos casos, las niñas asumieron roles de liderazgo técnico, mientras que los niños contribuyeron con ideas creativas, y viceversa. Esta dinámica contradice estereotipos comunes y sugiere que el entorno lúdico y colaborativo favorece la equidad.

Fortalecimiento del trabajo colaborativo: La organización en equipos de cinco fomentó la cooperación constante. Los estudiantes compartieron ideas, se apoyaron mutuamente para identificar y corregir errores lógicos (“bugs”) y celebraron colectivamente los avances.

Se observó que los equipos desarrollaron estrategias informales de organización: algunos designaron roles

espontáneos (quien escribe en el código, quien prueba, quien busca recursos), mientras que otros trabajaban de forma más rotativa.

En todos los casos, la resolución de problemas se convirtió en una actividad colectiva, donde las dificultades individuales eran abordadas por el grupo. Estos hallazgos fueron consistentes con los criterios de evaluación definidos en los instrumentos aplicados, los cuales permitieron registrar de forma sistemática el progreso individual y grupal.

DISCUSIÓN

Los resultados de este estudio de caso validan la hipótesis central: el aprendizaje basado en juegos utilizando Scratch contribuye significativamente al aprendizaje de la programación en niños de 9 a 11 años.

Este enfoque va más allá de la simple gamificación; se trata de una innovación pedagógica que utiliza el lenguaje natural de los niños “el juego” como vehículo para el aprendizaje de conceptos abstractos.

La efectividad observada se alinea con los postulados de Resnick et al. (2009), quienes señalaron que entornos como Scratch, basados en el principio de “proyectos, pasión, pares y juego”, son ideales para cultivar la creatividad y el pensamiento sistemático en niños.

Asimismo, Brennan y Resnick (2012) propusieron un marco que permite mapear bloques de Scratch a conceptos formales como iteración, paralelismo y variables, lo que facilita la evaluación del aprendizaje. En este estudio, la capacidad de los niños para explicar oralmente sus decisiones de codificación evidencia que no solo reprodujeron mecánicas, sino que internalizaron los conceptos subyacentes.

Investigaciones previas respaldan que Scratch facilita el desarrollo del pensamiento computacional desde edades tempranas al eliminar barreras sintácticas y permitir centrarse en la lógica (Maloney et al., 2010; Grover & Pea, 2013).

Estudios más recientes (Stewart, 2023; Piedade et al., 2023; Wen et al., 2023; Ibrohim, 2023) confirman que Scratch incrementa significativamente la motivación, el rendimiento y la comprensión de conceptos fundamentales como secuencias, bucles y condicionales. Los hallazgos de esta investigación coinciden plenamente con estas tendencias, aportando además evidencia en un contexto latinoamericano poco documentado.

Un hallazgo crucial fue la ausencia de brecha de género en el desempeño y la participación. Este resultado contradice estereotipos comunes que asocian las actividades tecnológicas con un interés predominantemente masculino.

En línea con lo observado por Stewart (2023), cuando se utilizan metodologías centradas en la colaboración, la creatividad y el juego, los sesgos de género tienden a diluirse. Este hallazgo tiene importantes implicaciones para el diseño de políticas educativas que busquen fomentar vocaciones científico-técnicas inclusivas desde la infancia.

La observación del trabajo colaborativo corrobora que la programación no es una actividad solitaria, sino que puede fomentar habilidades sociales esenciales, como la comunicación, la negociación y la cooperación (Llorens-Largo et al., 2015).

En este estudio, la organización en equipos pequeños y la naturaleza abierta de las tareas promovieron una interdependencia positiva, donde el éxito grupal dependía de la contribución de cada miembro. Este aspecto resulta particularmente valioso en el contexto del Tercer Perfeccionamiento de la Educación Cubana, que enfatiza el desarrollo de habilidades socioafectivas junto con las cognitivas.

Este estudio también aborda críticas recurrentes hacia Scratch. Investigaciones como las de Lozano y García-Peñalvo (2021), Güney et al. (2021) y Weintrop et al. (2020) señalan limitaciones como la rigidez, la baja calidad del código o la falta de escalabilidad.

Sin embargo, estas objeciones suelen referirse a usos no acompañados de un diseño pedagógico adecuado. En esta experiencia, la estructuración en fases (análisis, modificación, creación libre) y el rol mediador del docente transformaron estas supuestas limitaciones en oportunidades de aprendizaje.

Al respecto, Almarales Sarasola et al. (2018) señalaron que enseñar programación es más complejo que enseñar otras asignaturas, pues implica adquirir un pensamiento computacional y plasmar soluciones mediante

un lenguaje de programación, donde cada problema se resuelve de manera diferente.

Este estudio confirma que Scratch, al facilitar la experimentación y la creatividad, constituye una herramienta idónea para enfrentar este desafío, siempre que se acompañe de una metodología que integre el análisis, la modificación y la creación.

Limitaciones y futuras investigaciones: Este estudio se centra en un grupo pequeño y en un contexto específico (curso de verano extracurricular), lo que limita la generalización de los hallazgos.

Futuras investigaciones podrían replicar el estudio con una muestra más grande y diversa, incluyendo escuelas de distintas provincias. También podrían aplicar instrumentos de medición cuantitativa (pre-test/post-test) para evaluar el progreso en habilidades específicas de pensamiento computacional.

Sería valioso explorar la aplicación de esta metodología en el aula formal, integrada en el currículo escolar diario, así como investigar el impacto a largo plazo de esta formación temprana en el rendimiento académico en otras asignaturas. Finalmente, se sugiere analizar en profundidad las dinámicas de género en contextos educativos formales con esta metodología.

CONCLUSIONES

Las estrategias lúdicas demostraron un impacto significativo en el aprendizaje de la programación en niños de 9 a 11 años, tal como se evidenció en el curso de verano impartido, donde su empleo resultó efectivo para mejorar la motivación y el compromiso de los participantes. La implementación del estudio de mecánicas de juego facilitó no solo la comprensión de conceptos de programación, sino también el fomento de la creatividad y la resolución de problemas, mientras que la programación por bloques, combinada con elementos lúdicos, permitió a los niños desarrollar habilidades técnicas como el pensamiento lógico, la colaboración y la capacidad de adaptarse a nuevos desafíos tecnológicos. En lo relacionado con la innovación pedagógica, la implementación de estas estrategias representa una innovación significativa en el ámbito educativo, ya que, al utilizar juegos y programación por bloques, se crea un entorno de aprendizaje más dinámico y atractivo, adaptable a diferentes estilos de aprendizaje y niveles de habilidad. Se concluye que el uso de estrategias lúdicas y mecánicas de juego en Scratch es altamente efectivo para enseñar programación a niños, pues el enfoque gamificado incrementa la motivación, promueve la creatividad y fortalece el pensamiento lógico, por lo que se recomienda su incorporación en entornos educativos formales, junto con la capacitación docente específica para su implementación.

REFERENCIAS BIBLIOGRÁFICAS

- Almarales Sarasola, M., Goire Castilla, M. M., & García Fernández, O. (2018). Tutorial para el estudio de las asignaturas programación I y II en la carrera de telecomunicaciones y electrónica en la Universidad de Oriente. *REFCALE*, 6(2), 85-94. <https://refcale.ulead.edu.ec/index.php/refcale/article/view/2744>
- Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2), 77-101. <https://doi.org/10.1191/1478088706qp063oa>
- Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. Proceedings of the 2012 Annual Meeting of the American Educational Research Association, Vancouver, Canada.
- Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From game design elements to gamefulness: Defining "gamification". In Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments (pp. 9-15). Association for Computing Machinery. <https://doi.org/10.1145/2181037.2181040>
- Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, 42(1), 38-43. <https://doi.org/10.3102/0013189X12463051>
- Güney, D. G., Isoviita, M., Wennberg, T., & Heljanko, K. (2021). Improving readability of Scratch programs with search-based refactoring. arXiv. <https://arxiv.org/pdf/2108.07114.pdf>
- Ibrohim, F. (2023). Implementación y formación del profesorado de educación primaria en pensamiento computacional: Una revisión sistemática. *Revista Iberoamericana de Educación*, 81(2), 45-67. <https://doi.org/10.35362/rie.v81i2.37572>
- Llorens-Largo, F., Gallego-Durán, F. J., Villagrà-Arnedo, C. J., Compañ-Rosique, P., Satorre-Cuerda, R., & Molina-Carmona, R. (2015). Lecciones aprendidas en gamificación. In Actas del III Congreso Internacional sobre Aprendizaje, Innovación y Competitividad (CINAIC 2015) (pp. 123-130). Universidad Politécnica de Madrid.

Lozano, C. Z., & García-Peñalvo, F. J. (2021). Educational challenges for computational thinking in K–12 education: A systematic literature review of "Scratch" as an innovative programming tool. *Computers*, 10(6), Article 69. <https://doi.org/10.3390/computers10060069>

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The Scratch programming language and environment. *ACM Transactions on Computing Education*, 10(4), Article 16. <https://doi.org/10.1145/1868358.1868363>

Piedade, M., Silva, A., & Costa, L. (2023). La herramienta Scratch Jr. como metodología para el desarrollo del pensamiento computacional. *Revista de Innovación Educativa*, 10(1), 23-38.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., & Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60-67. <https://doi.org/10.1145/1592761.1592779>

Romero, A., & Martínez, S. (2022). Pensamento computacional e a formação docente: desafios e possibilidades didáticas com o uso da ferramenta Scratch. *Diálogo Educacional*, 22(3), 789-812. <https://periodicos.uninove.br/dialogia/article/download/21701/9560>

Stewart, A. (2023). Implementación y formación del profesorado de educación primaria en pensamiento computacional: Una revisión sistemática. *Revista Iberoamericana de Educación*, 81(2), 45-67. <https://doi.org/10.35362/rie.v81i2.37572>

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2020). Computational thinking in programming with Scratch in primary schools: A systematic review. *Computer Applications in Engineering Education*. Advance online publication. <https://doi.org/10.1002/cae.22255>

Wen, X., Li, Y., & Zhang, H. (2023). Entrenamiento en pensamiento computacional y construcción de modelos de evaluación de aprendizaje profundo basados en el curso de programación modular Scratch. *Computational Intelligence and Neuroscience*, 2023, Article 3760957. <https://doi.org/10.1155/2023/3760957>

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. <https://doi.org/10.1145/1118178.1118215>

Yin, R. K. (2018). *Case study research and applications: Design and methods* (6th ed.). SAGE Publications.

Conflicto de intereses

Los autores declaran no tener ningún conflicto de intereses.

Declaración de responsabilidad de autoría

Los autores del manuscrito señalado, DECLARAMOS que hemos contribuido directamente a su contenido intelectual, así como a la génesis y análisis de sus datos; por lo cual, estamos en condiciones de hacernos públicamente responsable de él y aceptamos que sus nombres figuren en la lista de autores en el orden indicado. Además, hemos cumplido los requisitos éticos de la publicación mencionada, habiendo consultado la Declaración de Ética y mala praxis en la publicación.

Walfrido Camué Ortiz: Conceptualización, Investigación, Metodología, Redacción – borrador original, Software.

Oscar García Fernández: Análisis formal, Supervisión, Validación, Redacción – revisión y edición.

José Manuel Izquierdo Pardo: Curación de datos, Administración del proyecto, Recursos, Supervisión.