

Diseño de actividades de pensamiento computacional y aprendizaje automático para niñas y niños

Design of Computational Thinking and Machine Learning Activities for Girls and Boys

Luis Eduardo Guzmán-García¹, Katia Itzel Alcocer-Aguilar¹, María Susana Ávila-García¹, Erick Franco-Gaona¹, Marco Bianchetti¹

¹ Departamento de Estudios Multidisciplinarios, División de Ingenierías, Campus Irapuato-Salamanca, Universidad de Guanajuato. le.guzmangarcia@ugto.mx, ki.alcoceraguilar@ugto.mx, susana.avila@ugto.mx, e.francoagaona@ugto.mx, mb@ugto.mx

Resumen

Las habilidades relacionadas con el pensamiento computacional y el aprendizaje automático son fundamentales en la actualidad, por lo que es necesario trabajarlas desde edades tempranas. Para ello, existen diversas herramientas diseñadas para fomentar dichas habilidades en niñas y niños de manera didáctica, visual, y entretenida. Estas herramientas pueden clasificarse en gratuitas y de pago. En este trabajo nos enfocamos en las plataformas gratuitas más reconocidas en el área, como Scratch, Code.org, Teachable Machine, y Machine Learning for Kids, con el propósito de desarrollar actividades que promuevan el pensamiento computacional y el aprendizaje automático en estudiantes de primaria. El objetivo principal fue crear actividades digitales interactivas utilizando estas plataformas, que permitan a niñas y niños aprender conceptos clave de aprendizaje automático y pensamiento computacional. Para la planeación, diseño, y ejecución de las actividades, se utilizó la metodología SCRUM, organizando el trabajo en sprints y asignando tareas específicas a lo largo de cada iteración. Esta metodología permitió una mejor gestión del tiempo, priorización de actividades, revisión continua y adaptación según la retroalimentación recibida. Como resultado, se seleccionaron herramientas de aprendizaje de programación y de inteligencia artificial para niños y se diseñaron un conjunto de actividades de programación por bloques y actividades de clasificación automática enfocadas en temas de ciencia. Estas actividades están listas para ser implementadas en sesiones de trabajo con niñas y niños.

Palabras clave: Pensamiento computacional; Aprendizaje automático; Actividades programación; Niñas y Niños; Plataformas educativas; Educación temprana.

Introducción

Habilidades de Pensamiento Computacional

El pensamiento computacional es una habilidad cognitiva que permite resolver problemas de forma lógica y estructurada, similar a como lo haría una computadora. Implica descomponer problemas complejos, identificar patrones, formular algoritmos y aplicar conceptos de programación para generar soluciones (Robles et al., 2024).

En la actualidad, donde la tecnología es parte esencial de la vida cotidiana, es crucial que los niños no solo consuman tecnología, sino que aprendan a comprenderla y crearla. Fomentar el pensamiento computacional desde la infancia permite desarrollar una mentalidad crítica y creativa frente a los desafíos del entorno digital (Barrera & Montaña, 2015), (Aricapa López et al., 2024).

El pensamiento computacional se puede dividir en varias dimensiones clave, como, por ejemplo:

- **Descomposición:** Separar un problema complejo en partes más simples.
- **Reconocimiento de patrones:** Identificar similitudes y tendencias en datos.
- **Algoritmos:** Crear una secuencia de pasos ordenados para resolver un problema.

De acuerdo con (Robles et al., 2024), (Palop et al., 2025), algunos de los beneficios de desarrollar habilidades de pensamiento computacional incluyen:

- Mejora la capacidad para resolver problemas.
- Fomenta el razonamiento lógico.
- Desarrollar habilidades de planificación y análisis.
- Fortalece la creatividad y la innovación.
- Potencia el trabajo colaborativo y la comunicación en equipo.

Por otro lado, como se indica en (Mitchell, 1997) el desarrollo de estas habilidades tiene varias limitantes, como, por ejemplo:

- Requiere capacitación docente específica.
- Escasa presencia en planos de estudio de nivel básico.
- Falta de acceso a herramientas tecnológicas en algunas escuelas.
- Poca cultura digital en niveles iniciales de educación.

Aprendizaje Automático

El aprendizaje automático (machine learning en inglés) es una rama de la inteligencia artificial que permite a las computadoras aprender de los datos, identificar patrones, y tomar decisiones sin ser programadas de manera específica para cada tarea (Scratch, n.d.).

Incorporar conceptos de aprendizaje automático desde edades tempranas prepara a los niños para entender el funcionamiento de tecnologías emergentes. Además, promueve la alfabetización digital avanzada y el pensamiento crítico frente a los algoritmos que influyen en su vida cotidiana. Introducir estas nociones con entornos accesibles y amigables, como la programación por bloques, facilita su comprensión y estimula el interés por la ciencia y la tecnología (Mitchell, 1997).

De acuerdo con (Palop et al., 2025), algunos de los beneficios de desarrollar habilidades de aprendizaje automático incluyen:

- Mejora la capacidad para resolver problemas.
- Fortalece la creatividad e innovación.
- Potencia el trabajo colaborativo y la comunicación.

Por otro lado, como se indica en (Palop et al., 2025) el desarrollo de estas habilidades tiene varias limitantes, entre las que encontramos:

- Requiere capacitación docente especializada.
- Limitado acceso a tecnología y recursos.
- Poca cultura digital y conocimientos previos en etapas tempranas.

Herramientas de Programación en Bloques y Aprendizaje Automático

Para facilitar el desarrollo del pensamiento computacional y la comprensión del aprendizaje automático en niñas y niños, se han empleado herramientas digitales accesibles y diseñadas para la educación básica. Estos permiten que los estudiantes interactúen con conceptos complejos de forma visual, intuitiva y lúdica.

Code.org

Es una plataforma gratuita que ofrece lecciones interactivas de programación. Sus actividades están pensadas para principiantes y permiten introducir conceptos básicos de lógica y estructuras computacionales de forma progresiva (Code.org, n.d.). La Figura 1 muestra una impresión de pantalla de la interfaz de la plataforma.

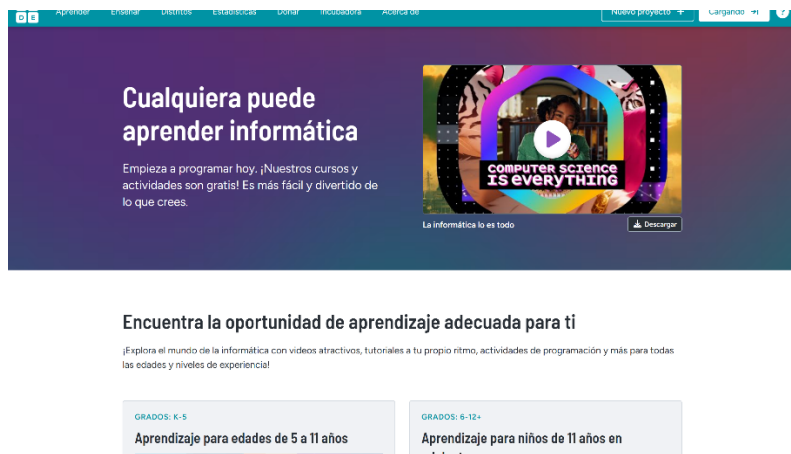


Figura 1. Interfaz principal del sitio web Code.org para la enseñanza de programación. Nota. Captura de pantalla tomada de <https://code.org>

Scratch

Es un entorno de programación por bloques ideal para estudiantes de nivel primario. Con esta herramienta, los niños pueden crear animaciones, historias, y juegos, al mismo tiempo que aprenden sobre secuencias, condicionales, y ciclos (Scratch, n.d.). La Figura 2 muestra una interfaz de la plataforma.

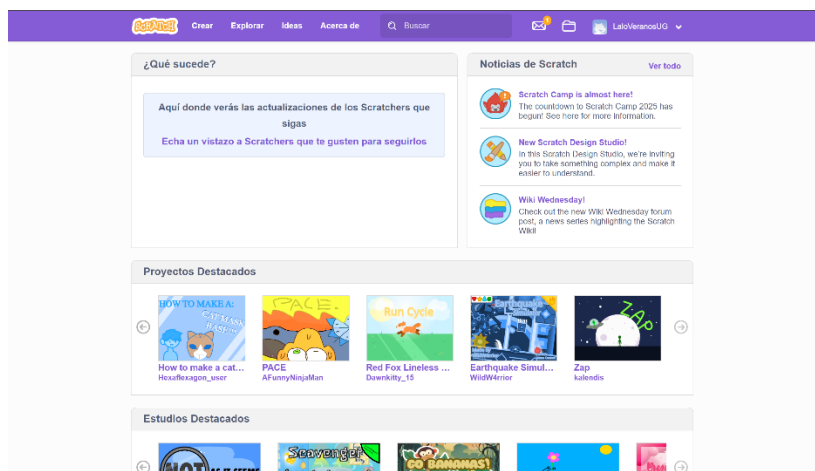


Figura 2. Interfaz principal del sitio web Scratch. Nota. Captura de pantalla tomada de <https://scratch.mit.edu>

Teachable Machine

Es una plataforma de Google que permite crear modelos simples de aprendizaje automático mediante ejemplos, sin necesidad de programar. Los estudiantes pueden entrenar modelos de reconocimiento de imágenes, sonidos o poses, lo que les permite comprender de forma visual cómo “aprende” una máquina (Google, n.d.). La Figura 3 muestra una impresión de pantalla de la interfaz de la plataforma.

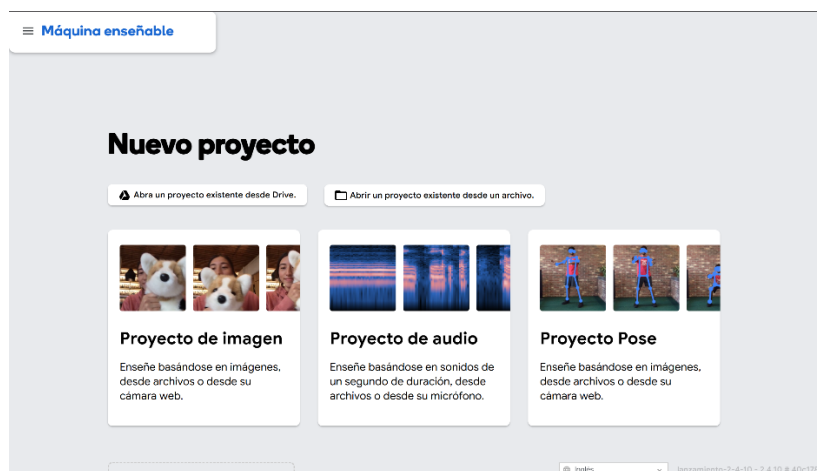


Figura 3. Interfaz principal del sitio web Teachable Machine. Nota. Captura de pantalla tomada de <https://teachablemachine.withgoogle.com/train>

Machine Learning for Kids

Esta plataforma, desarrollada como una extensión de Scratch, permite a los estudiantes usar modelos de clasificación (texto, imágenes, etc.) dentro de sus proyectos de programación, facilitando la conexión entre IA y programación creativa (Lane, n.d.). La Figura 4 muestra una impresión de pantalla de la plataforma.



Figura 4. Interfaz principal del sitio web Machine Learning for Kids. Nota. Captura de pantalla tomada de <https://machinelearningforkids.co.uk>

Materiales y Métodos

El proyecto se desarrolló utilizando un enfoque de desarrollo ágil, basado en la metodología Scrum. Esta metodología permitió dividir el trabajo en ciclos iterativos llamados sprints, lo cual facilitó la planeación, el seguimiento y la evaluación de avances parciales. Cada sprint se enfocó en objetivos específicos, como el diseño, validación, mejora y documentación de actividades educativas.

El enfoque metodológico fue orientado al diseño y prueba de actividades para promover el pensamiento computacional y el aprendizaje automático en estudiantes de nivel primaria. La Figura 5 muestra en diagrama de la metodología utilizada.

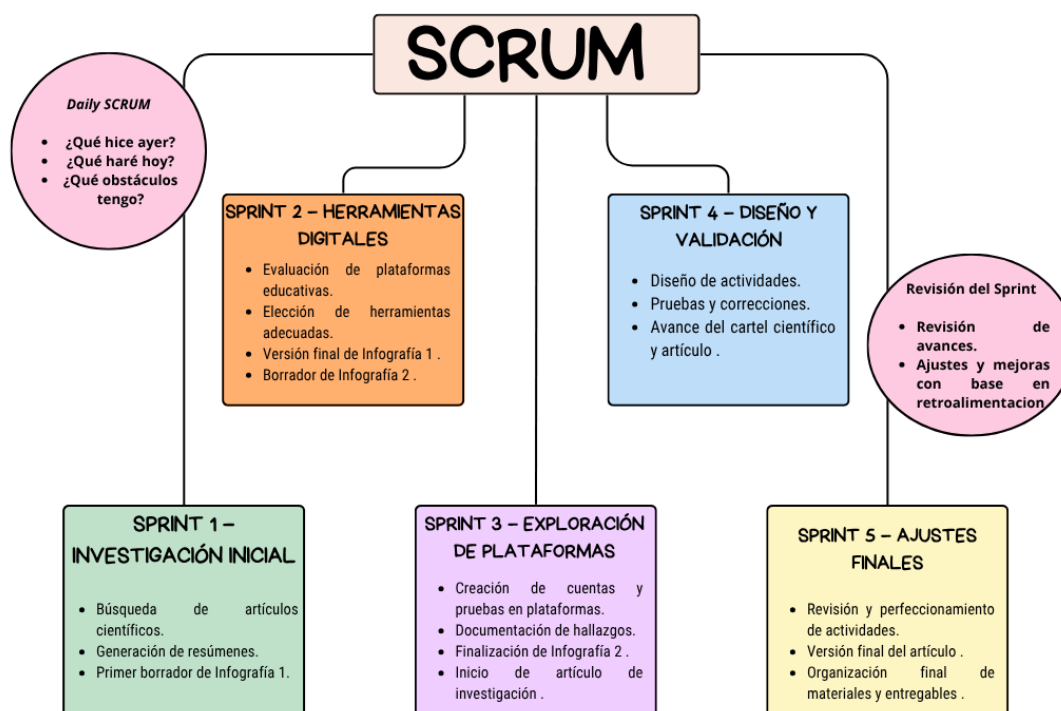


Figura 5. Metodología aplicada.
Fuente: Elaboración propia.

El desarrollo del proyecto se llevó a cabo en las siguientes fases:

1. **Planeación inicial:** definición de objetivos, herramientas y población objetivo.
2. **Selección de herramientas digitales:** se eligieron plataformas amigables como Scratch, Teachable Machine, Code.org y Machine Learning for Kids.
3. **Diseño de actividades:** se desarrollaron actividades basadas en temas científicos y tecnológicos adecuados para 4º y 5º grado de primaria.
4. **Evaluación interna:** revisión y validación por parte del equipo, considerando criterios de claridad, viabilidad y enfoque pedagógico.
5. **Mejora de actividades:** ajustes con base en retroalimentación recibida.
6. **Documentación y cierre:** organización de actividades por herramienta, redacción de justificaciones y estructuración final.

El proyecto se desarrolló en 5 sprints con duración de una semana y para cada sprint como se muestra en la Figura 5 se integraron entre 4 y 6 historias de usuario. Cada historia de usuario se desglosó en varias actividades, las cuales debían completarse diariamente. Como parte del seguimiento diario, se elaboraron informes de avance a través de publicaciones en las que todo el equipo respondía a tres preguntas clave:

- ¿Qué hice ayer?
- ¿Qué haré hoy?
- ¿Qué obstáculos tuve o puedo tener?

Estas dinámicas facilitaron la comunicación del equipo, la identificación de bloqueos y la alineación de objetivos diarios. Además, cada sprint tenía un objetivo específico y entregables claramente definidos:

- **Sprint 1:** Se investigó en artículos académicos, revistas científicas y plataformas oficiales relacionadas con el pensamiento computacional y el aprendizaje automático. Se generaron resúmenes e informes que fueron recopilados como evidencia en una carpeta compartida. El entregable fue el primer borrador de la infografía 1.
- **Sprint 2:** Se realizó la búsqueda y evaluación de herramientas digitales adecuadas para el diseño de actividades. Se eligieron las más pertinentes al proyecto y se elaboró la versión final de la infografía 1, además del primer borrador de la infografía 2.
- **Sprint 3:** Se procedió con la creación de cuentas en las herramientas seleccionadas, se exploraron sus funcionalidades y se documentaron los hallazgos. Los informes y análisis fueron recopilados como evidencia. Se finalizó la infografía 2 y se comenzó la redacción del artículo de investigación.
- **Sprint 4:** Se diseñaron y probaron las actividades educativas dentro de las herramientas seleccionadas. Se realizaron correcciones a partir de estas pruebas y se continuó trabajando tanto en el artículo como en el cartel científico del proyecto.
- **Sprint 5:** Se enfocó en la versión final del artículo de investigación, se corrigieron y perfeccionaron las actividades diseñadas, y se prepararon los materiales finales del proyecto.

Cada sprint concluyó con la revisión de avances, ajustes necesarios y la integración al repositorio de entregables.

Resultados y Discusión

Las actividades fueron diseñadas específicamente para ser implementadas con niñas y niños de nivel primaria, especialmente de 4.º y 5.º grado. Con el propósito de cumplir con los objetivos del proyecto, se procuró que dichas actividades tuvieran un enfoque científico o estuvieran relacionadas con temas de ciencia. Los contenidos se propusieron de manera sencilla, tomando en cuenta la naturaleza del grupo al que están dirigidos. Asimismo, se seleccionan temáticas que puedan resultar atractivas y despertar el interés de los estudiantes, con el fin de generar un impacto positivo en ellos.

Sin embargo, durante el proceso de diseño se identificaron ciertos desafíos. Algunos temas seleccionados presentaban un grado de complejidad mayor del esperado, y en otros casos, surgieron dificultades al momento de traducir conceptos científicos o tecnológicos en actividades comprensibles para niños pequeños. Como resultado, se decidió seleccionar finalmente doce temas, a partir de los cuales se diseñaron actividades específicas. Estas se distribuyeron entre las cuatro plataformas digitales seleccionadas (Scratch, Code.org, Teachable Machine y Machine Learning for Kids), asignando tres actividades a cada una. Cada actividad fue adaptada para que funcione correctamente dentro de los entornos y recursos de dichas plataformas.

En colaboración se plantearon diversas preguntas orientadas a verificar y evaluar la planeación de las actividades. De entre todas las preguntas formuladas, se seleccionan las más pertinentes y útiles como instrumentos de evaluación:

- ¿Tiene un tema científico claro?
- ¿Se eligió la plataforma adecuada?
- ¿Está definido el objetivo de aprendizaje?
- ¿Es clara la actividad que se va a realizar?
- ¿Se indicaron los conceptos de pensamiento computacional a aplicar?
- ¿Se plantearon las instrucciones principales paso a paso?

- ¿Considera mecanismos de interacción como animaciones, juego de preguntas, u otros?
- ¿Se han considerado posibles recursos necesarios?
- ¿Se definió cómo se va a evaluar o probar la actividad cuando esté lista?

Una vez consolidado el instrumento de evaluación (Las preguntas), se aplicó para revisar las actividades desarrolladas hasta ese momento. Los resultados fueron reveladores, ya que permitieron identificar deficiencias importantes. Entre las observaciones más relevantes, se destacó la falta de un enfoque científico claro en algunas actividades, lo cual era fundamental para los objetivos del proyecto. Asimismo, se identificó la ausencia de material de apoyo visual o auditivo necesario para facilitar el desarrollo de las actividades dentro de las plataformas digitales. Otro hallazgo clave fue la inexistencia de un método de evaluación estructurado, lo que impedía medir con precisión el impacto real de las actividades en los estudiantes.

Las actividades identificadas fueron:

- Reciclaje
- Tipos de frutas
- Alimentos saludables y no saludables

El objetivo de estas correcciones fue proporcionarles un enfoque científico más claro y alinearlas con los propósitos pedagógicos del proyecto. Además, se desarrolló un método de evaluación claro que permitirá valorar los resultados de cada actividad de manera objetiva. También se revisaron las plataformas utilizadas, con el fin de asegurar que contaran con los recursos visuales, audiovisuales suficientes. En los casos en que estos recursos no estuvieran disponibles de forma nativa, se indicó la posibilidad de crear materiales complementarios para enriquecer las actividades.

Actividades para el desarrollo del pensamiento computacional:

- Ciclo del agua
- Sistema solar
- Magnetismo
- Luz y colores
- Espacio y asteroides

Actividades para el aprendizaje automático:

- Clasificación de animales
- Estados del clima
- Clasificación de plantas y flores.
- Reciclaje: orgánico vs. inorgánico

Aunque estas actividades podrían requerir algunos ajustes menores, en general, se consideran lo suficientemente completos y funcionales, por lo que representan una base sólida sobre la cual continuar el desarrollo del proyecto.

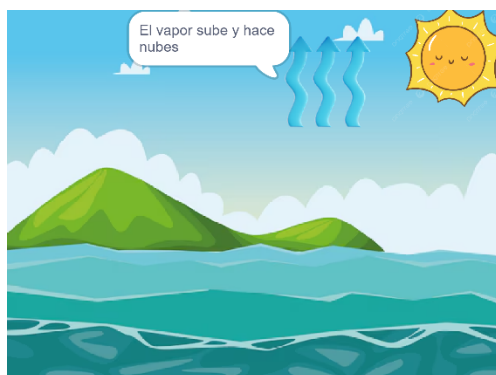
Las actividades que no aprobaron la evaluación anterior fueron revisadas con detenimiento. Las principales observaciones que motivaron su ajuste fueron:

- Ausencia de un enfoque científico claro o tratamiento superficial del tema.
- Falta de recursos visuales, auditivos o complementarios necesarios para el desarrollo de la actividad en la plataforma seleccionada.
- Inexistencia de un método de evaluación formal para medir resultados y validar el impacto de la actividad.

En esta fase, se procedió a realizar las mejoras necesarias en dichas actividades. En lugar de rediseñarlas completamente, se reformuló su enfoque desde una perspectiva científica, manteniendo su estructura original. Las ilustraciones 6-9 muestran ejemplos de las actividades diseñadas.



*Figura 6. Sistema Solar – Pensamiento Computacional.
Fuente: Elaboración propia.*



*Figura 7. Ciclo Del Agua – Pensamiento Computacional.
Fuente: Elaboración propia.*



Figura 8. Clasificación de frutas – aprendizaje automático.
Fuente: Elaboración propia.

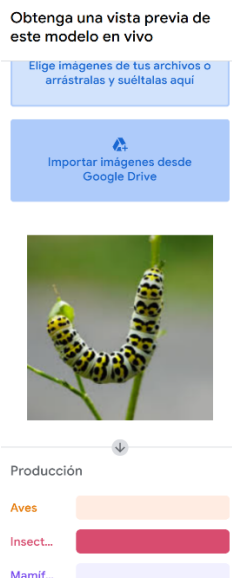


Figura 9. Clasificación de Animales – Aprendizaje automático.
Fuente: Elaboración propia.

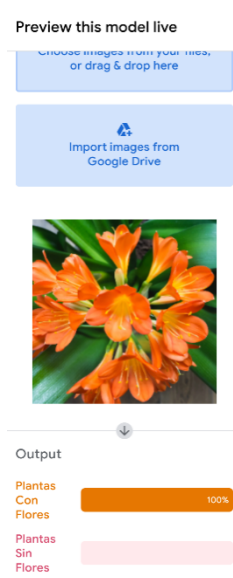


Figura 10. Clasificación de plantas – Aprendizaje automático.
Fuente: Elaboración propia.

Conclusiones

El desarrollo de actividades educativas orientadas al pensamiento computacional y al aprendizaje automático para niñas y niños de nivel primaria representa un reto importante, pero también una oportunidad valiosa para introducir desde edades tempranas habilidades. A lo largo de este proyecto, se comprobó que diseñar actividades efectivas en estas áreas requiere más que creatividad; Demanda planificación estructurada, enfoque pedagógico claro, y una comprensión profunda de las herramientas tecnológicas utilizadas.

La implementación de plataformas como Scratch, Code.org, Teachable Machine y Machine Learning for Kids permitió explorar distintas formas de aprendizaje activo y visual, logrando que los estudiantes se involucren en procesos de programación por bloques, análisis de datos, y clasificación automatizada. Estas experiencias no solo pueden fomentar la lógica y la resolución de problemas, sino que también desarrollan competencias científicas, trabajo colaborativo y pensamiento crítico.

En definitiva, este trabajo reafirma la necesidad de integrar el pensamiento computacional y el aprendizaje automático en la educación básica como parte de una formación integral que prepara a los estudiantes para enfrentar los desafíos de un mundo cada vez más digital.

Como parte de este trabajo se analizaron 4 herramientas de programación y de inteligencia artificial, sin embargo, es necesario extender este análisis para incluir otras herramientas disponibles. Además, se diseñó un número reducido de actividades, por lo que es necesario extender el número de actividades para proporcionar un portafolio que pueda ser considerado en sesiones prácticas.

Como trabajo futuro, se propone la realización de sesiones prácticas de programación con niñas y niños en entornos reales de aula, a fin de evaluar el impacto de las actividades diseñadas. Esta fase permitirá obtener datos sobre la comprensión, el interés y el nivel de apropiación de las actividades por parte del estudiante, así como identificar oportunidades de mejora en la implementación de estas actividades.

Bibliografía/Referencias

- Barrera, J., & Montaña, J. (2015). Pensamiento computacional con Scratch. TISE 2015.
- Palop, C., Redondo, M. A., Ortega, M., & Bravo, C. (2025). Redefinición del pensamiento computacional. Educación y Tecnologías de la Información.
- Mitchell, T. M. (1997). ¿Realmente funciona el aprendizaje automático? *AI Magazine*, 18 (3), 11–20.
- Robles, M., Ramírez, J., & Sánchez, L. (2024). Enfoque STEM+ y gamificación. *Acta Scientiæ Informaticæ*, 5(2), 34–45.
- Aricapa López, A. F., González López, N. P., & Jaramillo Gil, T. (2024). Secuencia didáctica basada en pensamiento computacional. *Revista Ideales*, (18), 102–108.
- Code.org. (sf). Code.org. <https://code.org/>
- Fundación Scratch (sf). Scratch: Imagina, Programa, Comparte. <https://scratch.mit.edu/>
- Google. (sf). Máquina de aprendizaje. <https://teachablemachine.withgoogle.com/>
- Lane, D. (sf). Aprendizaje automático para niños. <https://machinelearningforkids.co.uk/>