https://doi.org/10.69639/arandu.v13i2.2251

# Desarrollo del Pensamiento Computacional en Adolescentes Bachilleres

*Development of Computational Thinking in High School Adolescents*

**Carlos Andrés Parrales Villamar**
cparralesv4@unemi.edu.ec
https://orcid.org/0009-0002-7353-8680
Universidad Estatal de Milagro – Posgrado
Milagro – Ecuador

**Kevin Gabriel Castillo Villegas**
kcastillov@unemi.edu.ec
https://orcid.org/0009-0004-1688-5613
Universidad Estatal de Milagro
Milagro – Ecuador

**Adriana Margarita Sánchez Caicedo**
asanchezc4@unemi.edu.ec
https://orcid.org/0000-0002-1054-9756
Universidad Estatal de Milagro
Milagro – Ecuador

**Mirella Azucena Correa Peralta**
asanchezc4@unemi.edu.ec
https://orcid.org/0000-0003-1426-0244
Universidad Estatal de Milagro
Milagro – Ecuador

*Artículo recibido:18 marzo 2026- Aceptado para publicación:20 abril 2026*
*Conflictos de intereses: Ninguno que declarar.*

## RESUMEN

El presente artículo formó parte de un proyecto de vinculación con la sociedad y tuvo como objetivo analizar el nivel de pensamiento computacional de adolescentes bachilleres en el entorno social de los estudiantes de la carrera de Tecnología de la Información en línea. La investigación tuvo un enfoque cuantitativo de tipo experimental y se aplicó el Test de Pensamiento Computacional (TPC), que evalúa seis conceptos computacionales mediante 28 ítems distribuidos en cinco dimensiones. El instrumento fue aplicado a una muestra de 96 participantes con edades comprendidas entre los 14 y 18 años, pertenecientes al nivel de bachillerato en Ecuador, antes y después de un proceso de capacitación en programación con Scratch. Los resultados evidencian una mejora promedio del 14,17% en el nivel de pensamiento computacional de los participantes, con avances notables en los conceptos de Direcciones Básicas (17,45%), Condicional compuesto if/else (18,77%) y Funciones Simples (14,35%). Los hallazgos mostraron que la inclusión sistemática de contenidos de programación en los pénsum de bachillerato contribuye

significativamente al fortalecimiento del razonamiento lógico-matemático, independientemente del género, entorno socioeconómico o nivel académico previo de los estudiantes.

*Palabras clave:* pensamiento computacional, adolescentes bachilleres, Scratch, programación

## ABSTRACT

This article is part of a community outreach project and aims to analyze the level of computational thinking among high school adolescents in the social environment of students enrolled in the online Information Technology program. The research follows a quantitative-experimental design based on the Computational Thinking Test (CTT), which assesses six computational concepts through 28 items distributed across five dimensions. The instrument was administered to a sample of 96 participants aged 14 to 18, from secondary education institutions in Ecuador, before and after a programming training process using Scratch. Results show an average improvement of 14.17% in participants' computational thinking levels, with notable gains in Basic Directions (17.45%), Compound Conditional if/else (18.77%), and Simple Functions (14.35%). The findings suggest that the systematic integration of programming content into the high school curriculum significantly strengthens logical-mathematical reasoning, regardless of gender, socioeconomic background, or prior academic level.

*Keywords:* computational thinking; high school adolescents; Scratch; programming

## INTRODUCCIÓN

El pensamiento computacional se ha consolidado como una competencia fundamental en la educación del siglo XXI. Wing (2006) lo definió originalmente como el proceso de formular problemas y sus soluciones de manera que puedan ser ejecutadas por una computadora, abarcando habilidades como la abstracción, la descomposición de problemas, el reconocimiento de patrones y el diseño de algoritmos. Autores como Barr y Stephenson (2011), Aho (2012) y The Royal Society (2012) coinciden en que estas capacidades trascienden el ámbito de la informática y son esenciales para enfrentar los desafíos de una sociedad en constante transformación tecnológica.

En el contexto ecuatoriano, la educación técnica y tecnológica enfrenta el reto de preparar a los jóvenes no solo con conocimientos disciplinares, sino con habilidades de razonamiento aplicables a situaciones reales. Las carreras de Tecnología de la Información, tanto presenciales como en línea, tienen un rol estratégico en este proceso: sus estudiantes, al participar en proyectos de vinculación con la sociedad, pueden convertirse en agentes de cambio educativo en sus comunidades.

El modelo de inteligencias múltiples de Gardner (1995) reconoce la inteligencia lógico-matemática como la capacidad de utilizar eficazmente los números y razonar acertadamente, desarrollando sensibilidad hacia patrones y relaciones lógicas. Piaget (1969), por su parte, describió el desarrollo de esta inteligencia en etapas que van desde el estadio sensoriomotor hasta el pensamiento hipotético-deductivo, el cual se consolida precisamente en la adolescencia —etapa en la que se encuentran los participantes de esta investigación—, haciendo de este periodo un momento especialmente propicio para la intervención pedagógica en pensamiento computacional.

Diversos estudios han demostrado la efectividad de herramientas como Scratch para desarrollar competencias computacionales en población juvenil. Mendoza (2018) reportó una mejora del 17% en el nivel lógico-matemático de estudiantes de primaria en Perú tras la introducción de Scratch. Molina Chalacán et al. (2018) documentaron un mejoramiento global del 30% en rendimiento académico en estudiantes ecuatorianos de básica media al incorporar programación informática. A nivel de bachillerato, sin embargo, la evidencia empírica sobre intervenciones sistemáticas en entornos de vinculación universitaria es aún escasa.

El presente trabajo se enmarca en un proyecto de vinculación con la sociedad. A través de este proyecto, estudiantes universitarios lideraron talleres de programación básica con Scratch dirigidos a adolescentes bachilleres de sus entornos sociales inmediatos, contribuyendo al desarrollo de competencias computacionales en una población que normalmente no accede a este tipo de formación. El objetivo central es analizar el impacto de este proceso de capacitación en el nivel de pensamiento computacional de los participantes, utilizando como instrumento de medición el Test de Pensamiento Computacional (TPC) de Román-González et al.(2015)

El pensamiento computacional puede definirse como una actividad mental que involucra al computador como herramienta para la solución de problemas de la vida real, utilizando habilidades como la abstracción, la descomposición del problema, el uso de algoritmos, la evaluación de soluciones y el pensamiento en generalizaciones (Cuny, Snyder y Wing, 2010; Selby y Wollard, 2013). Wing (2010) amplió esta definición al enfatizar que el pensamiento computacional no consiste en programar computadoras, sino en pensar como un científico computacional, lo que incluye dimensiones cognitivas de alto nivel que son aplicables en todas las disciplinas.

Desde una perspectiva educativa, Barr y Stephenson (2011) señalaron que llevar el pensamiento computacional a los niveles K-12 requiere integrar conceptos de ciencias de la computación en el currículo de manera transversal, no como una asignatura aislada. Esta visión ha sido adoptada por países como Estados Unidos (desde kindergarten hasta K-12), el Reino Unido (desde septiembre de 2014) y varios países latinoamericanos que están incorporando el pensamiento computacional en sus mallas curriculares.

## MATERIALES Y MÉTODOS

La presente investigación tuvo un enfoque cuantitativo de tipo experimental con diseño pre-test / post-test en un solo grupo. Se enmarca dentro del proyecto de vinculación con la sociedad, cuyo objetivo es transferir conocimientos de programación a la comunidad a través de los propios estudiantes universitarios como mediadores del aprendizaje.

La muestra estuvo compuesta por 96 adolescentes bachilleres con edades comprendidas entre los 14 y 18 años, residentes en diferentes cantones del Ecuador. Los participantes pertenecen al entorno social inmediato de los estudiantes de la carrera de Tecnología de la Información en línea, quienes facilitaron el acceso a esta población. No se realizó ningún tipo de estratificación previa por género, nivel socioeconómico o rendimiento académico, con el propósito de evaluar el impacto de la intervención en condiciones representativas de diversidad.

Se utilizó el Test de Pensamiento Computacional (TPC) de Román-González et al. (2015), adaptado al contexto ecuatoriano. El instrumento consta de 28 ítems de opción múltiple que evalúan seis conceptos computacionales de complejidad creciente: direcciones básicas, bucles (repetir veces y repetir hasta), condicionales (simple y compuesto) y funciones simples. Cada ítem presenta cuatro opciones de respuesta de las cuales solo una es correcta. Los resultados se expresan como porcentaje de respuestas correctas por concepto.

La intervención se desarrolló en tres fases:

- **Fase 1 – Pre-test**: aplicación del TPC a todos los participantes antes de iniciar el proceso de capacitación, con el fin de establecer la línea base del nivel de pensamiento computacional.

- **Fase 2 – Capacitación:** talleres de programación con Scratch organizados en niveles básico e intermedio. Los contenidos abarcaron: revisión de la interfaz de Scratch, código de programación, control, objetos, videojuegos, historias, características de algoritmos, partes de pantalla, procedimientos, agregar sprites, operadores, ciclos, condiciones y bloques completos. La capacitación fue facilitada por estudiantes universitarios de la carrera de Tecnología de la Información en línea.
- **Fase 3 – Post-test:** reaplicación del mismo instrumento TPC al concluir la capacitación, para medir el impacto de la intervención en el nivel de pensamiento computacional.

**Scratch como Herramienta Pedagógica**

Scratch es un lenguaje de programación visual de bloques desarrollado por el MIT Media Lab, diseñado específicamente para introducir a niños y jóvenes en los conceptos fundamentales de la programación de manera lúdica e intuitiva (Balch, Chung y Brennan, 2011). Su interfaz permite a los usuarios crear proyectos interactivos mediante la combinación de bloques de instrucciones, evitando la complejidad sintáctica de los lenguajes de programación convencionales.

La evidencia empírica respalda la efectividad de Scratch para el desarrollo del pensamiento computacional. Pérez Narváez, Roig Vila y Jaramillo Naranjo (2020) demostraron su utilidad incluso en educación superior. Para el nivel de bachillerato, Scratch representa una herramienta de acceso inmediato que permite a estudiantes sin experiencia previa en programación alcanzar una comprensión funcional de conceptos como secuencias, bucles, condicionales y funciones en un tiempo reducido.

**El Test de Pensamiento Computacional (TPC)**

El instrumento utilizado en esta investigación es el Test de Pensamiento Computacional: diseño y psicometría general, desarrollado por Marcos Román-González, Pérez González y Jiménez Fernández (2015). Este test está compuesto por 28 ítems de opción múltiple con cuatro alternativas de respuesta, de las cuales solo una es correcta. Los ítems están alineados con los estándares de la CSTA (Computer Science Teachers Association) para la educación en ciencias de la computación y evalúan los siguientes conceptos computacionales organizados en cinco dimensiones:

**Tabla 1**

*Dimensiones y conceptos evaluados por el Test de Pensamiento Computacional (TPC)*

| Dimensión | Conceptos computacionales (dificultad creciente) | Cantidad de ítems |
|---|---|---|
| | Direcciones básicas | 4 |
| | Bucles—'repetir veces' | 4 |
| | Bucles—'repetir hasta' | 4 |
| | Condicional simple—'if' | 4 |

| | | |
|---|---|---|
| **Concepto computacional abordado** | Condicional compuesto—'if/else' | 4 |
| | Mientras que—'while' | 4 |
| | Funciones Simples | 4 |
| **Dimensión** | **Entorno gráfico o interfaz** | **Cantidad de ítems** |
| Entorno-Interfaz del ítem | El laberinto | 23 |
| | El lienzo | 5 |
| **Dimensión** | **Estilo de las alternativas** | **Cantidad de ítems** |
| Estilo de las alternativas de respuesta | Visual por flechas | 8 |
| | Visual por bloques | 20 |
| **Dimensión** | **Existencia / Inexistencia de anidamiento** | **Cantidad de ítems** |
| Existencia o inexistencia de anidamiento | Existencia de anidamiento | 19 |
| | Inexistencia de anidamiento | 9 |
| **Dimensión** | **Tareas cognitivas** | **Cantidad de ítems** |
| **Tarea requerida** | Secuenciación | 14 |
| | Completamiento | 9 |
| | Depuración | 5 |

*Nota.* Test de Pensamiento Computacional aplicado a adolescentes, elaborado por los autores

El TPC ha sido validado psicométricamente en diversas poblaciones y contextos, demostrando consistencia interna y validez de constructo satisfactorias (Román González, 2015). Su aplicabilidad ha sido confirmada independientemente de la ubicación geográfica, cultura, género, edad y tipo de institución educativa.

## RESULTADOS Y DISCUSIÓN

El análisis se realizó mediante estadística descriptiva. Para cada concepto computacional se calculó mediante Excel el porcentaje promedio de respuestas correctas en el pre-test y en el post-test, y se determinó la mejora como la diferencia porcentual entre ambos momentos. Se reportan los resultados por pregunta individual y por concepto computacional, así como el promedio global de mejora.

A continuación, se presentan los resultados obtenidos por concepto computacional, comparando los niveles de desempeño antes y después de la capacitación.

**Concepto Computacional: Direcciones Básicas (Preguntas 1-4)**

Las preguntas 1 a 4 evaluaron el concepto computacional de Direcciones Básicas. Se observó una mejora global del 17,45% entre el pre-test y el post-test, siendo este uno de los conceptos con mayor ganancia porcentual en la muestra.

**Tabla 2**

*Resultados del concepto computacional: Direcciones Básicas*

| Direcciones Básicas | Pre-Test (%) | Post-Test (%) | Mejora (%) |
|---|---|---|---|
| **Pregunta 1** | 85,4 | 93,8 | +8,4 |
| **Pregunta 2** | 71,9 | 81,3 | +9,4 |
| **Pregunta 3** | 49,0 | 72,9 | +23,9 |
| **Pregunta 4** | 22,9 | 51,0 | +28,1 |
| **Promedio** | 57,3 | 74,75 | +17,45 |

Los resultados muestran una tendencia de mejora creciente conforme aumenta la complejidad de la pregunta, con ganancias más notables en las preguntas 3 y 4. Las preguntas 1 y 2, con ayuda visual de tipo Pacman, ya mostraban niveles elevados de respuesta correcta en el pre-test (85,4% y 71,9% respectivamente), lo que explica sus menores márgenes de mejora.

**Concepto Computacional: Bucles – 'Repetir Veces' (Preguntas 5-8)**

Las preguntas 5 a 8 evaluaron el concepto de bucles con la estructura 'repetir veces'. La mejora global fue del 12,80%, con variaciones significativas entre preguntas según su nivel de dificultad y representación visual.

**Tabla 3**

*Resultados del concepto computacional: Bucles – 'Repetir Veces'*

| Bucles – 'Repetir Veces' | Pre-Test (%) | Post-Test (%) | Mejora (%) |
|---|---|---|---|
| **Pregunta 5** | 74,0 | 86,5 | +12,5 |
| **Pregunta 6** | 65,6 | 82,3 | +16,7 |
| **Pregunta 7** | 42,7 | 54,2 | +11,5 |
| **Pregunta 8** | 33,3 | 43,8 | +10,5 |
| **Promedio** | 53,9 | 66,7 | +12,80 |

La mejora fue consistente en las cuatro preguntas de este bloque, con valores superiores en las preguntas con apoyo visual directo. Este comportamiento refleja la influencia positiva del tipo de representación gráfica en la comprensión de los bucles por parte de los adolescentes.

**Concepto Computacional: Bucles – 'Repetir Hasta' (Preguntas 9-12)**

Las preguntas 9 a 12 abordaron el concepto de bucles condicionales con la estructura 'repetir hasta'. La mejora global fue de 10,375%, con resultados heterogéneos que reflejan la mayor complejidad abstracta de este tipo de bucle en comparación con el anterior.

**Tabla 4**

*Resultados del concepto computacional: Bucles – 'Repetir Hasta'*

| Bucles – 'Repetir Hasta' | Pre-Test (%) | Post-Test (%) | Mejora (%) |
|---|---|---|---|
| **Pregunta 9** | 80,2 | 85,4 | +5,2 |
| **Pregunta 10** | 16,7 | 17,7 | +1,0 |
| **Pregunta 11** | 43,8 | 69,8 | +26,0 |
| **Pregunta 12** | 41,7 | 51,0 | +9,3 |
| **Promedio** | 45,6 | 55,975 | +10,375 |

La pregunta 11 registró la mejora individual más alta de este bloque (26,0%), mientras que la pregunta 10 mostró una variación mínima, posiblemente atribuible a su alto nivel de abstracción y la ausencia de referentes visuales directos. Destaca que la pregunta 9 ya presentaba un nivel alto de aciertos en el pre-test (80,2%), lo que limita el margen de mejora observable.

**Concepto Computacional: Condicional Simple – 'if' (Preguntas 13-16)**

Las preguntas 13 a 16 evaluaron el condicional simple. La mejora global fue de 15,375%, constituyéndose en uno de los conceptos con avance más sostenido. Todas las preguntas mostraron incremento positivo.

**Tabla 5**

*Resultados del concepto computacional: Condicional Simple – 'if'*

| Condicional Simple – 'if' | Pre-Test (%) | Post-Test (%) | Mejora (%) |
|---|---|---|---|
| **Pregunta 13** | 34,4 | 58,3 | +23,9 |
| **Pregunta 14** | 28,1 | 47,9 | +19,8 |
| **Pregunta 15** | 30,2 | 43,8 | +13,6 |
| **Pregunta 16** | 32,3 | 36,5 | +4,2 |
| **Promedio** | 31,25 | 46,625 | +15,375 |

Es importante destacar que este bloque partió de niveles bajos en el pre-test (promedio 31,25%), lo que indica que los conceptos condicionales representaban un área de oportunidad significativa para la muestra. La capacitación permitió elevar estos niveles de manera consistente, con la mayor ganancia registrada en la pregunta 13 (+23,9%).

**Concepto Computacional: Condicional Compuesto – 'if/else' (Preguntas 17-20)**

Las preguntas 17 a 20 evaluaron el condicional compuesto 'if/else'. Este concepto presentó la mejora global más alta del estudio, con un incremento de 18,775%. Todas las preguntas del bloque mostraron avances significativos.

**Tabla 6**

*Resultados del concepto computacional: Condicional Compuesto – 'if/else'*

| Condicional Compuesto – 'if/else' | Pre-Test (%) | Post-Test (%) | Mejora (%) |
|---|---|---|---|
| **Pregunta 17** | 34,4 | 54,2 | +19,8 |
| **Pregunta 18** | 41,7 | 59,4 | +17,7 |
| **Pregunta 19** | 38,5 | 62,5 | +24,0 |
| **Pregunta 20** | 42,7 | 56,3 | +13,6 |
| **Promedio** | 39,325 | 58,1 | +18,775 |

La pregunta 19 alcanzó el mayor avance individual del bloque (+24,0%). Los niveles de partida del pre-test en este concepto (promedio 39,325%) fueron superiores a los del condicional simple, sugiriendo que la presentación de alternativas compuestas puede resultar más intuitiva para los estudiantes. La capacitación potenció esta disposición inicial de manera notable.

**Concepto Computacional: Mientras Que – 'while' (Preguntas 21-24)**

Las preguntas 21 a 24 evaluaron el concepto del bucle 'while'. La mejora global fue de 14,05%, con comportamientos heterogéneos entre las preguntas, reflejando la complejidad inherente a este tipo de estructura iterativa.

**Tabla 7**

*Resultados del concepto computacional: Mientras Que – 'while'*

| Mientras Que – 'while' | Pre-Test (%) | Post-Test (%) | Mejora (%) |
|---|---|---|---|
| **Pregunta 21** | 30,2 | 40,6 | +10,4 |
| **Pregunta 22** | 29,2 | 32,3 | +3,1 |
| **Pregunta 23** | 26,0 | 38,5 | +12,5 |
| **Pregunta 24** | 42,7 | 72,9 | +30,2 |
| **Promedio** | 32,025 | 46,075 | +14,05 |

La pregunta 24 registró la mejora individual más elevada de este bloque (+30,2%), mientras que la pregunta 22 mostró el avance más discreto (+3,1%). Esta variabilidad sugiere que algunos aspectos del bucle 'while' requieren un mayor tiempo de exposición o estrategias pedagógicas complementarias para ser internalizados de forma eficiente.

**Concepto Computacional: Funciones Simples (Preguntas 25-28)**

Las preguntas 25 a 28 evaluaron el concepto de funciones simples. La mejora global fue de 14,35%, con todos los ítems del bloque mostrando avance positivo.

**Tabla 8**

*Resultados del concepto computacional: Funciones Simples*

| Funciones Simples | Pre-Test (%) | Post-Test (%) | Mejora (%) |
|---|---|---|---|
| **Pregunta 25** | 26,0 | 44,8 | +18,8 |

| | | | |
|---|---|---|---|
| **Pregunta 26** | 55,2 | 61,5 | +6,3 |
| **Pregunta 27** | 44,8 | 59,4 | +14,6 |
| **Pregunta 28** | 47,9 | 65,6 | +17,7 |
| **Promedio** | 43,475 | 57,825 | +14,35 |

La pregunta 25 partió del nivel más bajo del bloque en el pre-test (26,0%) pero logró la mayor mejora relativa (+18,8%). La pregunta 26 ya contaba con un nivel de aciertos relativamente alto en el pre-test (55,2%), lo que limitó su margen de crecimiento posterior.

**Resumen Global de Resultados**

El análisis global de los siete conceptos computacionales evaluados evidencia una mejora promedio del 14,17% entre el pre-test y el post-test, lo que confirma el impacto positivo de la capacitación con Scratch en el desarrollo del pensamiento computacional de los adolescentes bachilleres participantes.

**Tabla 9**

*Resumen global de resultados: mejora por concepto computacional*

| Concepto Computacional | Pre-Test (%) | Post-Test (%) | Mejora (%) |
|---|---|---|---|
| **Direcciones Básicas** | 57,30 | 74,75 | +17,45 |
| **Bucles – 'Repetir Veces'** | 53,90 | 66,70 | +12,80 |
| **Bucles – 'Repetir Hasta'** | 45,60 | 55,98 | +10,38 |
| **Condicional Simple – 'if'** | 31,25 | 46,63 | +15,38 |
| **Condicional Compuesto – 'if/else'** | 39,33 | 58,10 | +18,78 |
| **Mientras Que – 'while'** | 32,03 | 46,08 | +14,05 |
| **Funciones Simples** | 43,48 | 57,83 | +14,35 |
| **PROMEDIO GENERAL** | 43,27 | 57,44 | +14,17 |

El concepto con mayor mejora fue el Condicional Compuesto – 'if/else' (+18,78%), seguido de Direcciones Básicas (+17,45%) y Condicional Simple – 'if' (+15,38%). El concepto con menor avance fue Bucles – 'Repetir Hasta' (+10,38%), posiblemente debido a su mayor nivel

de abstracción y la disponibilidad limitada de tiempo en la capacitación para abordar sus variaciones más complejas.

Los resultados obtenidos en este estudio son coherentes con los hallazgos reportados en investigaciones previas sobre el impacto de la programación con Scratch en el desarrollo del pensamiento computacional. La mejora promedio del 14,17% es comparable a la registrada por Barahona-Rojas et al. (2022) en una intervención similar con niños de primaria en Loja (15,96%), y al 17% reportado por Mendoza (2018) para el mismo nivel educativo en Perú, aunque en nuestro caso la población corresponde al nivel de bachillerato.

Un hallazgo relevante es el comportamiento diferenciado según el tipo de representación visual de los ítems. Las preguntas con apoyo gráfico de tipo Pacman (laberinto) o con representaciones intuitivas del entorno Scratch mostraron consistentemente mejores porcentajes de respuesta en el post-test, en comparación con aquellas que presentaban el código en forma de bloques abstractos. Este patrón, también identificado por Barahona-Rojas et al. (2022), confirma la importancia pedagógica de utilizar herramientas con interfaz visual para la enseñanza inicial de la programación.

El caso del condicional simple – 'if' merece una atención especial. En el estudio de referencia de Loja se registró una disminución en este concepto (-7,43%), fenómeno que no se reprodujo en nuestra muestra, donde se observó una mejora de +15,38%. Esta diferencia podría explicarse por la mayor madurez cognitiva de los adolescentes bachilleres (14-18 años) en comparación con los niños de primaria (7-11 años), quienes se encuentran en el estadio de operaciones formales de Piaget y poseen mayor capacidad para el razonamiento hipotético-deductivo que requiere la comprensión de las estructuras condicionales.

Desde la perspectiva del proyecto de vinculación, los resultados sugieren que los estudiantes universitarios de la carrera de Tecnología de la Información en línea pueden desempeñar un papel efectivo como mediadores del aprendizaje computacional en sus comunidades. Esta modalidad de transmisión de conocimiento no solo beneficia a los adolescentes participantes, sino que también refuerza las competencias pedagógicas y de comunicación técnica de los propios estudiantes universitarios, generando un impacto bidireccional positivo.

En cuanto a las limitaciones del estudio, cabe señalar que el diseño pre-test / post-test sin grupo control impide atribuir con certeza absoluta la mejora observada exclusivamente a la intervención educativa. Asimismo, la duración de la capacitación (número de horas académicas) podría haberse ampliado para abordar con mayor profundidad los conceptos de mayor complejidad, como los bucles 'repetir hasta' y el 'while'. Futuras investigaciones deberían considerar el seguimiento longitudinal de los participantes y la incorporación de grupos de control para fortalecer la validez interna de los hallazgos.

Finalmente, la ausencia de diferencias sustanciales en el desempeño por género confirma lo señalado por Barahona-Rojas et al. (2022): el aprendizaje computacional no está determinado por el género, y las intervenciones bien diseñadas son igualmente efectivas para hombres y mujeres. Esto es especialmente relevante en el contexto ecuatoriano, donde la brecha de género en carreras tecnológicas es un desafío pendiente.

## CONCLUSIONES

El presente estudio demuestra que una intervención pedagógica estructurada con Scratch, facilitada por estudiantes universitarios en el marco de un proyecto de vinculación con la sociedad, produce una mejora estadísticamente relevante en el nivel de pensamiento computacional de adolescentes bachilleres ecuatorianos. La capacitación con Scratch generó una mejora promedio del 14,17% en el pensamiento computacional de los 96 participantes, con avances positivos en los siete conceptos computacionales evaluados. El mayor incremento se registró en el condicional compuesto – 'if/else' (+18,78%) y el menor en bucles – 'repetir hasta' (+10,38%).

El modelo de vinculación universitaria estudiante-comunidad demostró ser una vía efectiva y sostenible para extender el acceso a la educación computacional a poblaciones que no cuentan con esta formación en su currículo escolar regular. Se recomienda la incorporación formal del pensamiento computacional en el currículo de bachillerato ecuatoriano, articulada con los proyectos de vinculación de las carreras tecnológicas, como estrategia de equidad educativa y preparación para el mundo del trabajo digital.

Los ítems con representación visual intuitiva (entorno de laberinto tipo Pacman) favorecen una comprensión más rápida y profunda de los conceptos computacionales, validando la elección de Scratch como herramienta pedagógica para poblaciones sin experiencia previa en programación. Se propone ampliar la duración de las capacitaciones para abordar niveles intermedio y avanzado, realizar estudios longitudinales que permitan evaluar la retención de los aprendizajes, y explorar el impacto diferenciado en función de variables como el tipo de institución educativa, el área geográfica y el acceso previo a tecnología.

## REFERENCIAS


Aho, A. V. (2012). Computation and computational thinking. The Computer Journal, 57(7), 832-835. https://doi.org/10.1093/comjnl/bxs074

Balch, C., Chung, M., & Brennan, K. (2011). Computación creativa con Scratch 3.0: Guía curricular. Eduteka. http://eduteka.icesi.edu.co/articulos/hgse-scratch-computacion-creativa

Balladares Burgos, J. A., Avilés Salvador, M. R., & Pérez Narváez, H. O. (2016). Del pensamiento complejo al pensamiento computacional. Dialnet, (2), 143-159. https://dialnet.unirioja.es/servlet/articulo?codigo=5973042

Barahona-Rojas, S. E., Muñoz Pilozo, A. G., Sanmartín-Zhingre, P. N., Yunga-Benítez, A. E., & Torres-Berrú, Y. M. (2022). Fortalecimiento del pensamiento computacional en niños y adolescentes de la ciudad de Loja. Dom. Cien., 8(3), 657-676. http://dx.doi.org/10.23857/dc.v8i3

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? ACM Inroads, 2(1), 48-54. https://doi.org/10.1145/1929887.1929905

CSTA. (2011). K-12 computer science standards (Level 2). Computer Science Teachers Association. http://csta.acm.org/Curriculum/sub/CurrFiles/CSTA_K12_CSS.pdf

Cuny, J., Snyder, L., & Wing, J. M. (2010). Demystifying computational thinking for non-computer scientists. Unpublished manuscript.

Ferrándiz, C., Bermejo, R., Sainz, M., Ferrando, M., & Prieto, M. D. (2008). Estudio del razonamiento lógico-matemático desde el modelo de las inteligencias múltiples. Anales de Psicología, 24(2), 213-222. https://revistas.um.es/analesps/article/view/42731

Gardner, H. (1995). Multiple intelligences as a catalyst. The English Journal, 84(8), 16-18. https://www.jstor.org/stable/821182

Medina Hidalgo, M. I. (2018). Estrategias metodológicas para el desarrollo del pensamiento lógico-matemático. Dialnet, 125-132. https://dialnet.unirioja.es/servlet/articulo?codigo=6595073

Mendoza Aguirre, M. (2018). Software de programación "Scratch" en el desarrollo del pensamiento lógico matemático de estudiantes de una institución educativa primaria, Chincha – 2017 [Tesis de licenciatura]. Universidad César Vallejo. https://hdl.handle.net/20.500.12692/29889

Molina Chalacán, L. J., Jalón Aria, E. J., & Albarracín Zambrano, L. O. (2018). Inclusión de la programación informática como herramienta para el desarrollo del razonamiento lógico y abstracto en el pensamiento de los niños de educación general básica, nivel medio. Dilemas Contemporáneos: Educación, Política y Valores, 28, 1-18.

Moravec, J. W. (2013). Knowmad society: The "new" work and education. On the Horizon, 21(2), 79-83. https://doi.org/10.1108/10748121311322978

Pérez González, J., Román González, M., & Jiménez Fernández, C. (2015). Test de pensamiento computacional: Diseño y psicometría general. III Congreso Internacional sobre Aprendizaje, Innovación y Competitividad (CINAIC), 1-7.

Pérez Narváez, H., Roig Vila, R., & Jaramillo Naranjo, L. (2020). Uso de SCRATCH en el aprendizaje de programación en educación superior. Catedra, 3(1), 28-45. https://doi.org/10.29166/catedra.v3i1.2006

Piaget, J., & Berth, E. W. (2013). Mathematical epistemology and psychology. Springer Science & Business Media.

Román González, M. (2015). Test de pensamiento computacional: Principios de diseño, validación de contenido y análisis de ítems [Comunicación]. CINAIC 2015.

Selby, C., & Woollard, J. (2013). Computational thinking: The developing definition. University of Southampton. https://eprints.soton.ac.uk/356481

Vilanova, G. E. (2018). Tecnología educativa para el desarrollo del pensamiento computacional. Sistemas, Cibernética e Informática, 15(3), 25-32.

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? Carnegie Mellon University. https://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf