

# Hacia una educación equitativa en ciencias de la computación



Revista EIA  
ISSN 1794-1237  
e-ISSN 2463-0950  
Año XIX/ Volumen 21/ Edición N.42  
Julio - diciembre de 2024  
Reia4210 pp. 1-24

Publicación científica semestral  
Universidad EIA, Envigado, Colombia

## PARA CITAR ESTE ARTÍCULO / TO REFERENCE THIS ARTICLE /

Vieira, C.; Espinal, A.; Chiu, J.; Bredder, E.; Harris, P.; Wilkens, K.  
Hacia una educación equitativa en ciencias de la computación  
Revista EIA, 21(42), Reia4210.  
pp. 1-24.  
<https://doi.org/10.24050/reia.v21i42.1734>

### ✉ Autor de correspondencia:

Alejandro Espinal  
Estudiante PhD  
Universidad del Norte, Colombia  
[espinal@uninorte.edu.co](mailto:espinal@uninorte.edu.co)

**Recibido:** 20-10-2023

**Aceptado:** 27-05-2024

**Disponible online:** 01-07-2024

CAMILO VIEIRA<sup>1</sup>

✉ ALEJANDRO ESPINAL<sup>1</sup>

JENNIFER CHIU<sup>2</sup>

ERIC BREDDER<sup>2</sup>

PAOLA HARRIS<sup>1</sup>

KIM WILKENS<sup>2</sup>

1. Universidad del Norte, Colombia
2. University of Virginia, Estados Unidos

## Resumen.

En la actualidad, la equidad de acceso a la educación en ciencias de la computación representa un desafío: la educación en ciencias de la computación es indispensable para la participación activa en un mundo computacional, pero las iniciativas actuales tienden a aumentar las brechas existentes por factores culturales, socioeconómicos y del contexto. Este artículo propone un modelo para la educación equitativa en computación que aporte a la reducción de estas brechas, mediante tres estrategias principales: (I) Investigar cómo aprende los estudiantes en contextos particulares; (II) Generar capacidad en docentes para enseñar computación a través de programas de desarrollo profesional, mientras investigamos cómo aprenden los y las docentes, y cómo logran (o no) llevarlo al aula; y (III) Diseñar dispositivos accesibles para la enseñanza de la computación. Esta accesibilidad incluye dispositivos que sean de bajo costo, pero que también ser versátiles, en cuanto a su apariencia, las formas de interacción, y los conceptos que permitan enseñar. Este artículo describe el modelo para la educación equitativa en computación, y muestra un caso de estudio para cada una de las estrategias.

**Palabras clave:** Educación, Pensamiento Computacional, Ciencias de la Computación, Formación Docente, Equidad

# Towards equitable computer science education

## Abstract.

Currently, equity of access to computer science education represents a challenge: computer science education is indispensable for active participation in a computational world, but current initiatives tend to increase existing gaps due to cultural, socioeconomic and contextual factors. This paper proposes a model for equitable computing education that contributes to the reduction of these gaps through three main strategies: (I) Investigating how students learn in particular contexts; (II) Building teacher capacity to teach computing through professional development programs, while investigating how teachers learn, and how they succeed (or fail) in bringing it to the classroom; and (III) Designing accessible devices for teaching computing. This accessibility includes devices that are low-cost, but that are also versatile, in terms of their appearance, the ways they interact, and the concepts they teach. This article describes the model for equitable computer education, and shows a case study for each of the strategies.

**Keywords:** Education, Computational Thinking, Computer Science, Teacher Education, Equity.

## 1. Introducción

La computación forma parte de nuestras vidas. En el mundo actual, impulsado por la tecnología, una comprensión más profunda de los fundamentos de la computación puede permitir a los estudiantes tomar el control de la forma en que las tecnologías afectarán a sus vidas, su cultura y su futuro. Los estudiantes necesitan oportunidades para aprender sobre computación que les permitan convertirse en creadores, y no sólo en consumidores de la próxima ola de descubrimientos, soluciones e innovaciones (Wing, 2018). Sin embargo, existen grandes brechas entre quienes tienen acceso a la educación en computación, que están marcadas por el género, el grupo étnico, el contexto y el estrato socioeconómico (Chute, 2009), y estas brechas están aumentando, en lugar de hacer más equitativa la educación en (Clinging, 2006; Hill et al., 2010; Rojas-López et al., 2014).

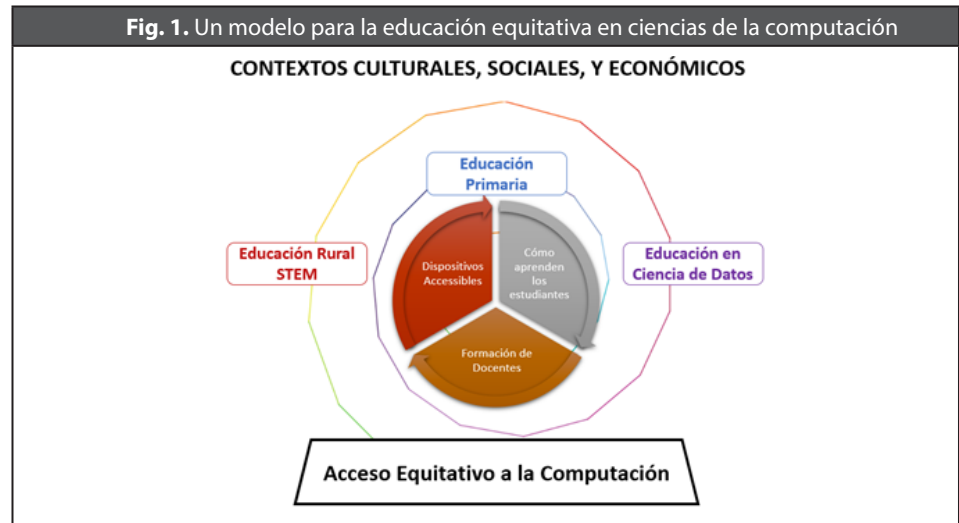
Este problema no es exclusivo de un solo país. Tanto Estados Unidos como Reino Unido (Curzon et al., 2014), Australia (Yadav et al., 2017), los Países Bajos (Yadav, Good, Voogt, & Fisser, 2017)

y Colombia (Espinal et al., 2022) han desarrollado iniciativas para integrar la computación al currículo, como el diseño y publicación de planes de estudio de computación para la educación básica, o la implementación de iniciativas de desarrollo profesional docente. Sin embargo, dado que la enseñanza en computación es bastante reciente, es necesario desarrollar investigación que nos permita comprender cómo aprendemos computación en diferentes niveles y contextos. Además, en muchos de estos lugares hay muy pocos programas formales para que los docentes aprendan a incorporar la computación en su enseñanza, y la infraestructura tecnológica en las escuelas es limitada, lo que a su vez limita la cantidad y calidad de la educación en computación que se ofrece (Angeli et al., 2016; Caeli & Bundsgaard, 2020).

La mayor parte de lo que sabemos sobre cómo aprenden computación y programación los estudiantes procede de países desarrollados y de estudiantes que tienen recursos para dedicarse a la computación (Vieira et al., 2023). Se necesita más investigación para comprender cómo, los estudiantes de diferentes contextos (por ejemplo, cultura, idioma, acceso a la tecnología, estatus socioeconómico) aprenden computación, así como qué es lo que ellos aportan a los ambientes de aprendizaje de computación y los retos potenciales que pueden surgir. Por ejemplo, un estudio reciente identificó que el idioma en el que están diseñados los lenguajes de programación profesionales (inglés) supone un reto adicional para que estudiantes hispanohablantes transfieran su aprendizaje de las primeras experiencias de programación (Espinal et al., 2022).

Integrar la computación en el currículo con docentes que no tienen la preparación adecuada es también un desafío clave para proporcionar un acceso equitativo a la educación en computación. En los países latinoamericanos, por ejemplo, el desarrollo profesional para que los docentes integren las ciencias de la computación en las prácticas de aula es escaso, especialmente en contextos de bajos ingresos, donde la infraestructura tecnológica también es limitada (Trendov, Varas, & Zeng, 2019). Esto hace que pocos docentes aprovechen las tecnologías computacionales para apoyar el aprendizaje de los estudiantes. Por lo tanto, el desafío de hacer que la educación en computación sea equitativa y accesible

también requiere diseñar herramientas computacionales de bajo costo que puedan funcionar en contextos con una infraestructura mínima. Este proyecto aborda estos problemas al explorar cómo proporcionar oportunidades educativas tempranas a todos los niños para promover el interés y las habilidades fundamentales y mitigar las creencias estereotipadas sobre las ciencias de la computación (Varma, 2010; Cheryan, Plout, Handron, & Hudson, 2013). Los investigadores de la Universidad de Virginia (UVA – Estados Unidos) y la Universidad del Norte (UniNorte – Colombia) colaboran en una iniciativa a largo plazo a través de tres estrategias principales: (I) la realización de investigaciones para comprender cómo ocurre el aprendizaje de las ciencias de la computación en contextos de escuelas públicas y privadas; (II) la generación de capacidad en los docentes a través de oportunidades de desarrollo profesional (PD), de la mano de la investigación sobre cómo aprenden y cómo enseñan los docentes, de tal manera que podamos comprender qué apoyos adicionales pueden requerir, y cuáles son las barreras para una enseñanza efectiva; y, (III) el desarrollo y distribución de dispositivos accesibles para la enseñanza de la computación, que sean de bajo costo y que puedan también ser usados por estudiantes de diferentes contextos e incluso, por estudiantes en condición de discapacidad. Estas tres estrategias se pueden trabajar para diferentes niveles educativos y conceptos, pero siempre reconociendo el contexto cultural, socioeconómico y de contexto. Por ejemplo, en la Figura 1, mostramos que esto se podría trabajar para conceptos de computación y pensamiento computacional en educación primaria, pero también podríamos pensar cómo trabajar la educación STEM a través del pensamiento computacional para instituciones rurales usando como contexto la agricultura digital (o de precisión), o explorar cómo los estudiantes aprenden sobre ciencia de datos, cómo generar capacidad en los docentes, y qué tipo de herramientas o dispositivos accesibles deberían ofrecerse para llevar estos conceptos y prácticas al aula.



En el resto del artículo presentamos tres casos de estudio para la educación preescolar en pensamiento computacional. En la siguiente sección describimos nuestros resultados iniciales para el desarrollo de una trayectoria de aprendizajes sobre pensamiento algorítmico con niños y niñas de cinco y seis años. Luego, describimos una experiencia de desarrollo profesional docente que venimos ofreciendo para más de 10 países de Latinoamérica sobre pensamiento computacional para la primera infancia. Finalmente, describimos los avances en el desarrollo de ROVERSA, un dispositivo robótico versátil y accesible, que no solo es de bajo costo, sino que permitirá el aprendizaje de diferentes conceptos y habilidades, y el uso de diferentes “vestidos” que pueda ser usado para diversas actividades.

## 2. ¿Cómo los estudiantes de 5 y 6 años desarrollan habilidades de pensamiento algorítmico?

Diferentes estudiosos sostienen que el aprendizaje de las ciencias de la computación puede ser tan importante como el de las matemáticas o la lectoescritura (Grover y Pea, 2013). La educación en las ciencias de la computación en las aulas de preescolar y primaria implica conceptos como algoritmos, estructuras de control,

representación, procesos de diseño y depuración (Bers, 2018). Si bien ha habido una buena cantidad de investigación sobre el aprendizaje de ciencias de la computación en la educación secundaria y media, existe relativamente poco trabajo en los niveles de primaria y preescolar. Los investigadores no se ponen de acuerdo sobre la mejor manera de apoyar las habilidades de ciencias de la computación en edades tempranas y cómo el aprendizaje de la computación puede estar relacionado con el desarrollo cognitivo de los estudiantes (Relkin, de Ruiter, & Bers, 2021).

Nosotros proponemos que muchas de las habilidades asociadas al pensamiento computacional pueden enseñarse teniendo el pensamiento algorítmico y el diseño de algoritmos como habilidad central, al igual que otros han sugerido que, aunque el pensamiento computacional no sea solo programación, la programación puede ser una forma eficaz de enseñar habilidades del pensamiento computacional (Buitrago-Flórez et al., 2017). Por ejemplo, para enseñar habilidades de depuración, se necesita un algoritmo que depurar. Así mismo, los estudiantes pueden necesitar descomponer un problema cuando es demasiado grande para pensar en soluciones parciales más pequeñas y manejables, lo que se convierte en un algoritmo. Además, cuando identificamos una secuencia de pasos que se debe repetir a través de un ciclo o bucle, estamos realmente generando abstracciones. Por lo tanto, las habilidades de diseño de algoritmos están en el centro del desarrollo de las habilidades de pensamiento computacional en la primera infancia, y resultan un enfoque crítico para dicho nivel.

El pensamiento algorítmico en la primera infancia suele tomar forma a través de actividades que implican programar un robot educativo con movimientos para desplazarse a través de un laberinto y utilizando secuencias de pasos (Bakala et al., 2021; Clarke-Midura, Silvis, et al., 2021). Esto puede hacerse con robots como BeeBot (un robot en forma de abeja) o Code & Go (un ratón robótico), con herramientas en línea, como Code Studio de code.org, o incluso con actividades desconectadas, en las que los estudiantes tienen que utilizar flechas para programar el movimiento de alguna(o) de sus compañeras(os). Sin embargo, la investigación sobre las etapas por las que pasan los estudiantes para aprender a programar es

escasa (Bers, 2019), y hay poca investigación en educación infantil sobre los procesos cognitivos involucrados en el aprendizaje de la programación (Fedorenko et al., 2019) y sobre cómo el aprender a programar puede contribuir (o no) al desarrollo de habilidades cognitivas tempranas (Arfé et al., 2020).

Para contribuir a comprender estos fenómenos, el equipo de investigación diseñó un conjunto de actividades de aprendizaje que incluían actividades desconectadas y el uso del BeeBot, y utilizó la teoría fundamentada para comprender cuáles son las formas en que los estudiantes resuelven retos de pensamiento algorítmico. Estas actividades se presentan en la tabla 1 (Vieira, Velasquez, Chiu, 2023).

**Tabla 1.** Actividades de aprendizaje desconectadas y conectadas

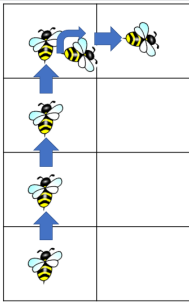
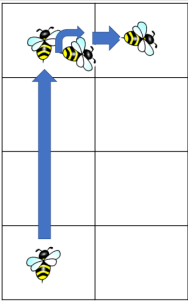
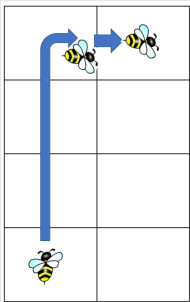
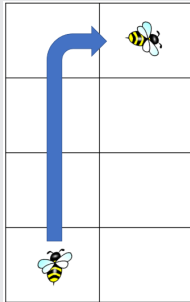
Básico	Intermedio	Avanzado

Los participantes en este estudio fueron 14 estudiantes (nueve niños y cinco niñas) de 5 y 6 años de un jardín infantil privado en Colombia. La edad media de estos niños era de 69,29 meses, con una desviación estándar de 3,45 meses. Estos participantes utilizaron cuatro diferentes aproximaciones para resolver las actividades de pensamiento algorítmico, cada una más sofisticadas que la anterior.

En el extremo menos sofisticado, los estudiantes utilizaron un enfoque de solución Paso a Paso (Tabla 2). Los estudiantes que utilizaron el Paso a Paso no planificaron más de un paso a la vez, por lo que presionaron el botón de comando flecha y luego el botón de “Ir”, y esto lo repitieron varias veces para mover al BeeBot al objetivo deseado. Luego, los estudiantes emplearon una estrategia más sofisticada, que denominamos Descomposición Simple. Los estudiantes que utilizaron la Descomposición Simple lograron programar trayectos rectos, pero no incorporaron giros en sus programas. Estos estudiantes planearon el número de pasos para avanzar hasta que se requería un giro y programaron solo estos movimientos en línea recta. Luego programaron el giro y repitieron esta secuencia según fuera necesario hasta llegar al punto final. La tercera aproximación para resolver estos problemas se caracterizó por estudiantes que realizaron descomposiciones más avanzadas, donde pudieron integrar los giros en la programación de los trayectos en línea recta, pero no se programaron movimientos adicionales después del giro. Esto parece estar asociado a las habilidades de rotación mental de los estudiantes, pues una vez el robot no estaba al frente de ellos, se les dificultaba predecir hacia dónde debían programar el movimiento. Además, entre las dificultades que los estudiantes se encontraban y los errores que cometían, se identificaron diferentes instancias donde la rotación mental resultaba ser el principal obstáculo. Por ejemplo, cuando tenían que mover el robot en forma de L (Reto Intermedio), los estudiantes confundían el botón de girar con el de desplazarse. Es decir, en lugar de presionar una vez girar y tres veces hacia adelante, presionaron tres veces girara. Finalmente, los estudiantes que utilizaron el diseño completo del algoritmo pudieron programar todo el trayecto, incluyendo giros y líneas rectas. La Tabla 2 resume y muestra ejemplos de estos enfoques.



**Tabla 2.** Aproximaciones para resolver las actividades de pensamiento algorítmico

Paso a paso	Descomposición simple	Descomposición avanzada	Diseño completo del algoritmo
Los estudiantes programan el BeeBot para moverse un paso a la vez.	Los estudiantes planean el número de pasos para moverse en línea recta hasta que necesitan girar. Luego programan el giro, y repiten esta secuencia según lo necesiten	Los estudiantes planean el número de pasos para moverse hacia Adelante hasta que necesitan girar, incluyen el giro, pero no logran programar los siguientes pasos.	Los estudiantes logran planear y programar todo el algoritmo para que el BeeBot llegue a su destino.
			

Los estudiantes tuvieron más dificultades para crear el diseño completo del algoritmo para las tareas avanzadas en comparación con las tareas básicas e intermedias. Algunos de los estudiantes lograron avanzar en el nivel de sofisticación a través de la práctica y la retroalimentación dentro de cada una de las seis actividades de diseño de algoritmos, pero esto no fue siempre el caso.

La dificultad para avanzar hacia enfoques más sofisticados puede tener diferentes orígenes. Algunos pueden estar relacionados con la capacidad de memoria de trabajo de los estudiantes, otros con su incapacidad para planificar, otros con sus habilidades de rotación mental, y otros pueden ser una combinación de factores. Nuestra hipótesis es que, en el caso de la Descomposición Simple, los estudiantes podrían haber tenido dificultades con las habilidades de rotación mental, mientras que en el caso de la Descomposición Avanzada, la memoria de trabajo y las habilidades de rotación mental de los estudiantes podrían haber influido en los resultados.

Comprender las formas cada vez más sofisticadas en que los estudiantes entienden y desarrollan habilidades de pensamiento algorítmico y pensamiento computacional es importante, tanto para los educadores, a fin de proporcionar retroalimentación y diseñar actividades de aprendizaje algorítmico, como para los investigadores, a fin de poner a prueba nuevas hipótesis y explorar fenómenos emergentes relacionados con las etapas que atraviesan los estudiantes mientras aprenden a programar (Bers, 2019; Seah & Horne, 2020; Steedle & Shavelson, 2009). Estas progresiones también pueden ser utilizadas para el desarrollo profesional y pueden ofrecer un lenguaje común para educadores, investigadores y comunidades de práctica (Clements & Sarama, 2011; Siemon et al., 2017).

### **3. Desarrollo profesional docente en Ciencias de la Computación**

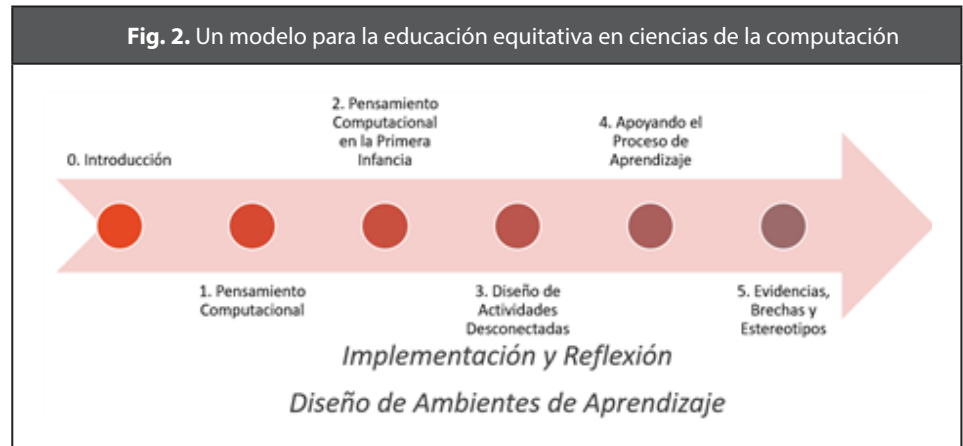
El desarrollo profesional de los docentes es esencial en la educación, ya que implica la adquisición de nuevos conocimientos y habilidades que pueden enriquecer y mejorar su enseñanza. Según Darling-Hammond (2017), este desarrollo se enfoca en mejorar la práctica docente a través del aumento de conocimientos y competencias. Además, Fullan et al. (2013) subrayan que el desarrollo profesional proporciona oportunidades para que los educadores mejoren tanto en su dominio del contenido como en sus técnicas pedagógicas y habilidades tecnológicas. En consecuencia, el desarrollo profesional docente brinda la posibilidad de mejorar las habilidades y el conocimiento de los docentes, lo que, a su vez, puede resultar en una mejora tanto en su enseñanza como en el rendimiento de los estudiantes (Yoon et al., 2007).

Bajo la premisa de fortalecer el conocimiento y habilidades de los docentes en computación, se han desarrollado cinco cohortes de un programa de formación a docentes de primera infancia. La primera cohorte fue realizada en el año 2021, seguida de otra cohorte en el 2022 y las otras tres cohortes durante el 2023. Este programa de desarrollo profesional docente se conceptualizó en respuesta a la necesidad fortalecer las habilidades de pensamiento computacional

desde las etapas iniciales del aprendizaje, considerando el pensamiento computacional como una habilidad analítica fundamental, equivalente en importancia a la lectura, la escritura y la aritmética (Wing, 2008).

Los objetivos de aprendizaje definidos para este programa incluyen: (I) Reconocer la importancia del pensamiento computacional desde los primeros años, (II) Conocer prácticas pedagógicas para enseñar pensamiento computacional como las actividades desconectadas y la progresión pedagógica de Usa-Modifica-Crea, que son utilizadas para fomentar el desarrollo del pensamiento computacional. (III) Comprender la brecha de género y estereotipos sociales, explorando factores que contribuyen a esta brecha y presentar actividades y herramientas para mitigar estas disparidades.

Este programa de formación docente en primera infancia da inicio con una comprensión general del pensamiento computacional y su aplicación específica en el contexto de la primera infancia. A lo largo del programa, los docentes fueron guiados a través de prácticas pedagógicas y herramientas diseñadas para fortalecer habilidades de pensamiento computacional en el aula como el pensamiento algorítmico. Además, cada semana se ofrecían actividades de ejemplo junto con un formato de reflexión, para que las y los participantes intentaran llevar al aula estas actividades e identificaran lo que funcionaba y lo que no. Al final de la formación se propuso el diseño de un ambiente de aprendizaje que integrara de forma orgánica el pensamiento computacional, incorporando las prácticas y herramientas pedagógicas abordadas durante la formación. La secuencia de las actividades se presentan en la figura 2, donde se detallan las acciones propuestas y temáticas abordadas en cada unidad.



Al final de las tres cohortes realizadas en el 2023, fueron recolectados 89 ambientes de aprendizaje diseñados por parte de los docentes participantes en los cuáles se integra el pensamiento computacional en la primera infancia. Para el diseño de estos ambientes de aprendizaje, se propone una estructura que contiene los puntos presentados en la tabla 3.

**Tabla 3.** Estructura de los ambientes de aprendizaje

Componente	Descripción
Título	Un título breve pero descriptivo del proyecto
Contexto	Descripción de las características del contexto, las dificultades y oportunidades que se presentan en el contexto donde se desarrolla la actividad.
Participantes	Descripción del nivel de desarrollo de los/las participantes, edad y los conocimientos previos esperados.
Objetivos de aprendizaje	Definición de los objetivos de aprendizaje que se esperan alcanzar con los/las estudiantes a través del desarrollo de las actividades.
Evaluación	Propuesta de actividades de evaluación para los objetivos de aprendizaje.
Actividades	Descripción de las estrategias pedagógicas o de instrucción que se van a usar para entregar el contenido y ayudar a los estudiantes a alcanzar los objetivos de aprendizaje.
Bibliografía	Lista de referencias utilizadas para el diseño del ambiente de aprendizaje

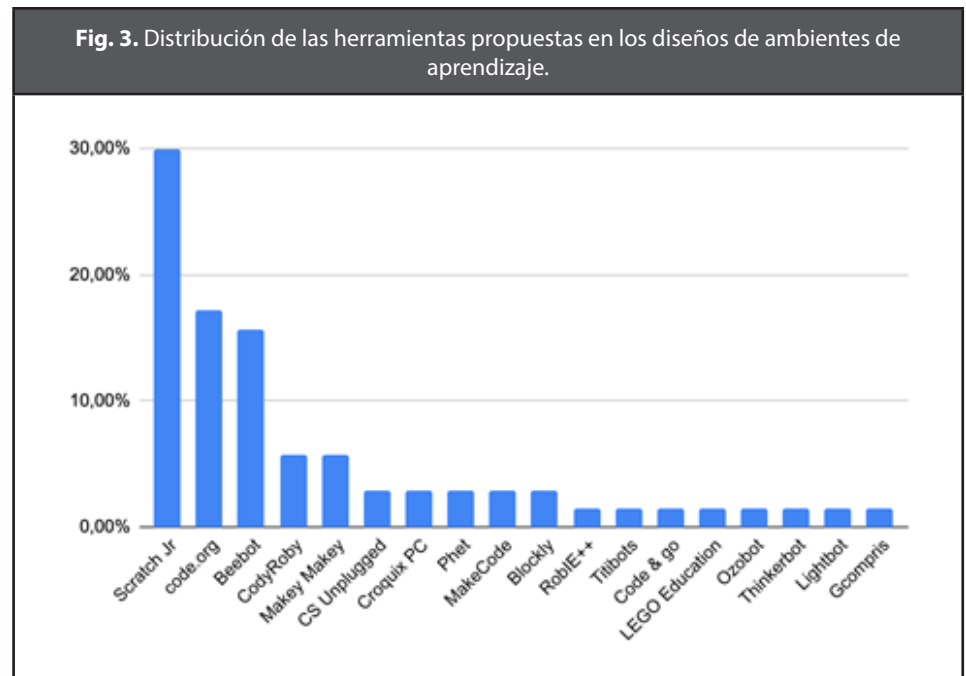
Algunas de las características sobre los docentes que entregaron los ambientes de aprendizaje son que el 81% (72) de las participantes son mujeres y el 19% (17) restante son hombres, y estos se distribuyen entre 58% (52) que pertenecen a colegios de sector oficial o público mientras que el 42% (37) restante pertenecen al sector privado. También se identificó que la zona en las cuales se encuentran distribuidos son, en su mayoría, de la zona urbana con un 90% (80). La distribución se presenta en la siguiente tabla 4.

**Tabla 4.** Distribución de los docentes que entregaron ambientes de aprendizaje

Sector	Género		Zona	
	Femenino	Masculino	Urbana	Rural
Privado	27	10	36	1
Oficial	45	7	44	8
<i>Total</i>	89		89	

La literatura existente respalda la importancia de herramientas tecnológicas intuitivas para el desarrollo temprano como la resolución de problemas y creatividad en los jóvenes (Lavidas et al., 2022). La introducción temprana de habilidades de programación requiere prácticas pedagógicas y tecnologías apropiadas para el desarrollo de los niños (Bers, 2017). Un estudio previo de Leidl y colegas (2017), enfatiza la necesidad de que los maestros seleccionen herramientas tecnológicas apropiadas para el desarrollo del pensamiento computacional y habilidades como el pensamiento algorítmico en sus estudiantes. Estos hallazgos coinciden con la indagación de las herramientas tecnológicas utilizadas en los diseños de ambientes de aprendizaje por parte de los docentes, encontrando que el 30% de los diseños de aprendizaje examinados proponen el uso de ScratchJr, respaldando así la efectividad de esta herramienta para introducir el pensamiento computacional en los primeros niveles educativos (Bers et al., 2014; Moreno-León et al., 2019). Además, el 17% de los docentes utilizan aplicaciones y herramientas de code.org, mientras que el 16% opta por Beebot, tanto el simulador como el dispositivo físico. También se observa que el 5,7% de los

docentes considera el uso de la herramienta CodyRoby y Makey Makey en sus actividades. Estos resultados se detallan en la Figura 3, que presenta la distribución de las herramientas propuestas en los ambientes de aprendizaje diseñados para integrar el Pensamiento Computacional en la primera infancia.



Fueron encontradas diversidad de herramientas utilizadas por los profesores en el proceso de diseño de entornos de aprendizaje. Esto sugiere que los profesores participantes buscaban una herramienta pedagógica adecuada para el proceso de aprendizaje de sus alumnos. No obstante, se puede ver en la figura 3 que las herramientas tecnológicas utilizadas en los entornos de aprendizaje incluyen en su mayoría lenguajes de programación basados en bloques. Las investigaciones existentes sugieren que estos lenguajes de programación pueden reducir la carga cognitiva que implican las actividades de programación introductorias (Jatzlau, 2018). De acuerdo con estudios realizados sobre el uso de estas herramientas, se afirma que la introducción temprana de las habilidades de codificación requiere tecnologías y prácticas apropiadas para el desarrollo de los niños en temprana edad (Bers, 2000).

Posterior a la entrega y revisión de los ambientes de aprendizaje diseñados por los estudiantes, el grupo de investigación realizará visitas a una muestra de docentes certificados con el propósito de observar el uso de los conocimientos y habilidades en el aula. Este proceso, está guiado por el marco de evaluación de Guskey (2000), que es ampliamente utilizado en la evaluación de los programas de desarrollo profesional docente, este marco consiste en cinco niveles a considerar para evaluar el programa: (I) Reacciones, que analiza la satisfacción de los participantes, así como sus reacciones o actitudes hacia la experiencia de aprendizaje, (II) Aprendizaje, que tiene como objetivo responder a la pregunta ¿qué aprendieron los participantes del programa de desarrollo profesional?, (III) Apoyo institucional, que identifica las percepciones de los participantes sobre el apoyo que encuentran su institución, (IV) Uso de conocimientos y habilidades, que determina el impacto del aprendizaje en sus prácticas dentro de su contexto/institución, y (V) Logros de los estudiantes, que explora si la capacitación lleva a un mejoramiento del desempeño estudiantil.

De acuerdo con el nivel IV asociado al uso de los conocimientos y las habilidades, serán realizadas las visitas a los ambientes de aprendizaje de algunos de los docentes certificados y realizar una observación del aula, lo que brindará una mejor comprensión del efecto del programa de desarrollo profesional docente en la educación.

Esta observación del aula se desarrollará en diferentes ciudades de Colombia en colegios privados y colegios públicos, en diversos contextos. Además de observar las prácticas pedagógicas, también se va a evaluar el impacto del uso de robots educativos de primera infancia y analizar su desempeño de los estudiantes en actividades específicas.

#### **4. Dispositivos Accesibles para la Educación en Computación**

El uso del cuerpo y los objetos manipulativos son factores importantes en los procesos de aprendizaje de los niños pequeños, ya que el sistema sensoriomotor afecta tanto al procesamiento de la

información como a la evocación (Kallia & Cutts, 2022). El uso de manipulativos favorece el desarrollo de esquemas sensoriomotores, lo que contribuye al pensamiento abstracto (Kallia & Cutts, 2022). En los primeros años, la comprensión de los conceptos se representa a través de las acciones de los niños. Las representaciones de los conceptos dependen del contexto en el que se aprenden, ya que este contexto informa la recreación de dichos conceptos (Clarke-Midura, Kozlowski, et al., 2021).

Los robots educativos pueden apoyar el desarrollo de las habilidades de las ciencias de la computación con los niños a través de este tipo de enfoques incorporados al aprendizaje, o apoyando el aprendizaje a través de la acción física (Su & Yang, 2023). Estos robots suelen ser atractivos, motivadores, promueven el aprendizaje lúdico y pueden utilizarse para contar historias (Bakala et al., 2021). Los robots educativos pueden servir de alternativa a las pantallas, ya que los colegios suelen evitar que los niños pequeños pasen demasiado tiempo frente a ellas (Clarke-Midura et al., 2021).

A pesar de estas posibilidades, la mayoría de las tecnologías tangibles actualmente disponibles para los maestros de primaria son relativamente costosas (por ejemplo, más de \$ 100 dólares / dispositivo) y normalmente ofrecen poca flexibilidad en términos de: (1) los diferentes tipos de conceptos de pensamiento computacional que el dispositivo puede ayudar a los estudiantes a aprender; (2) la apariencia del robot; y (3) las interfaces de usuario de entrada/salida que ofrecen, especialmente para niños con discapacidades. Por ejemplo, muchos de los dispositivos que se utilizan actualmente en las aulas de primaria ofrecen funcionalidades que permiten a los alumnos comprender la secuenciación y operaciones de control muy sencillas (por ejemplo, BeeBot, Code & Go), pero no ofrecen formas de que los alumnos avancen a través de conceptos de pensamiento computacional más sofisticados, como los ciclos o bucles. Los dispositivos que permiten conceptos de pensamiento computacional más sofisticados suelen tener un costo aún más alto, casi prohibitivo para las escuelas con pocos recursos (por ejemplo, Hummingbird, Finch, Kibo). Además, muy pocos de los robots existentes disponen de herramientas de investigación (por ejemplo, datos de registro de interacciones) que puedan utilizarse para mejorar la enseñanza y el aprendizaje.



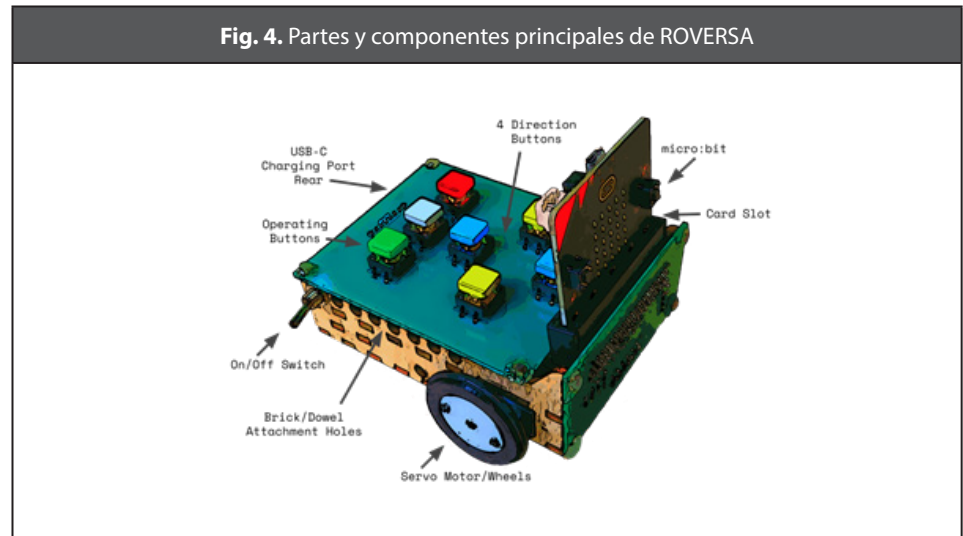
Por lo tanto, para que la enseñanza de las ciencias de la computación sea equitativa y accesible, también es necesario diseñar herramientas computacionales de bajo costo, que puedan funcionar en contextos con una infraestructura mínima. Además, la mayor parte de lo que sabemos sobre cómo aprenden computación los estudiantes procede de estudiantes neurotípicos que disponen de recursos para dedicarse a la computación o que se autoseleccionan en programas de computación en instituciones educativas, para los niveles de secundaria y educación media. Es necesario contar con más investigación para entender cómo los estudiantes neurodiversos y con discapacidades de los grados de preescolar y primaria aprenden ciencias de la computación, las fortalezas que aportan a los entornos de aprendizaje de CS y cualquier desafío potencial que pueda surgir (por ejemplo, Espinal et al., 2022).

#### *4.1. Estrategia de Diseño de ROVERSA*

Para hacer nuestro robot accesible y versátil (ROVERSA), decidimos utilizar la plataforma micro:bit, ya que fue diseñada para entornos educativos y nos permitía aprovechar la funcionalidad extensible y la infraestructura existente. Establecimos como objetivo de diseño un costo inferior a 20 dólares, excluyendo el micro:bit, dada la retroalimentación recibida de escuelas y profesores en Colombia y EE.UU., quienes compartieron que el micro:bit suele comprarse por separado en las escuelas (en el caso de Colombia, el programa Programación para Niños y Niñas ha distribuido miles de tarjetas micro:bit a las instituciones educativas). Nuestro objetivo era construir un dispositivo que fuera fácil y rápido de fabricar con materiales sencillos para que las piezas pudieran ser potencialmente cortadas con láser y ensambladas en las escuelas. Muchos otros kits de robots están hechos de piezas impresas en 3D, contienen muchas tuercas y tornillos y son difíciles de montar. Nuestro objetivo es que nuestros materiales supongan una barrera de entrada baja, ofrezcan formas sencillas de personalizar el exterior y amplíen la funcionalidad en el futuro.

#### *4.2. Prototipo actual de ROVERSA.*

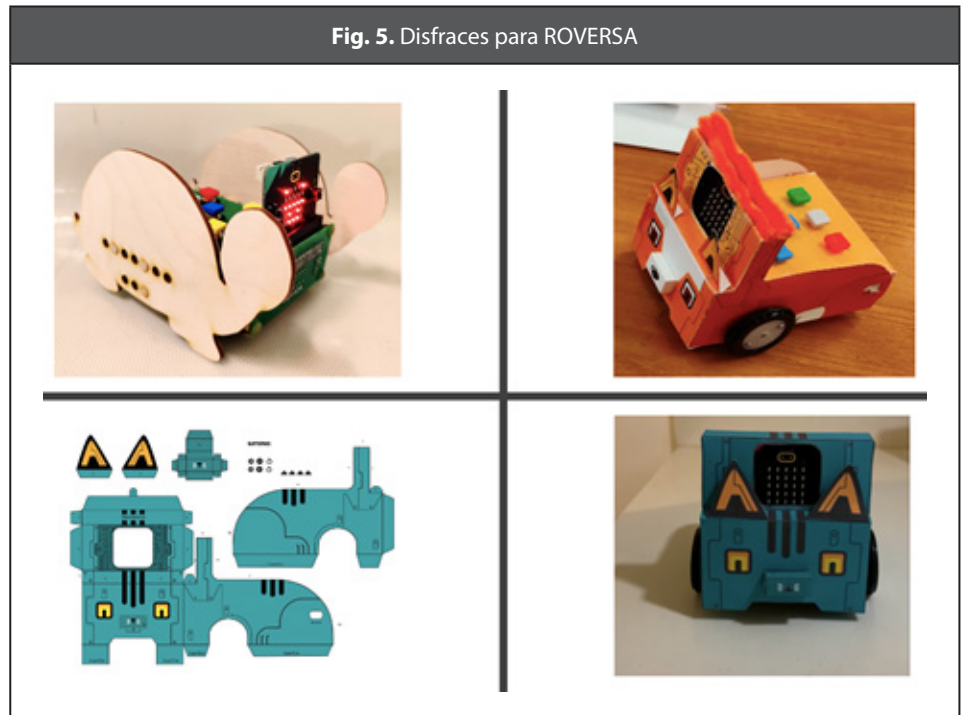
Nuestro prototipo actual es un chasis de madera cortado con láser que contiene agujeros espaciados uniformemente para encajar LEGO y otros conectores como tacos de madera para personalizarlo, incluida una plantilla que se puede dar a los estudiantes para que creen su propio exterior del robot con materiales sencillos como papel y cinta adhesiva. El dispositivo cuenta con pulsadores similares a los de otros robots educativos que permiten a los alumnos crear programas avanzando o retrocediendo y girando 90 grados a derecha o izquierda, con un botón verde para poner en marcha el robot y ejecutar el programa. El chasis de madera está hecho de uniones entrelazadas y seguras, sin tuercas ni tornillos, que encajan entre sí y pueden montarse y volverse a montar fácilmente varias veces. El prototipo actual tiene placas de circuito impreso personalizadas que condensan la electrónica y proporcionan capacidad para futuras ampliaciones. El dispositivo micro:bit se inserta verticalmente para mostrar la matriz de LED incorporada para que los estudiantes puedan programar y ver los mensajes que se desplazan. Hemos incluido un pulsador adicional para una mayor expansión, que actualmente cambia los idiomas de inglés a español y para calibrar algunos parámetros de desplazamiento y giros de ROVERSA. El robot utiliza una batería recargable de litio asegurada dentro de la carcasa y un puerto de carga. La figura 4 muestra el diseño de ROVERSA con la descripción de sus partes y componentes principales.



La restricción de diseño más importante según los comentarios de estudiantes y profesores tenía que ver con la exactitud y precisión de los movimientos del robot. Todos los profesores comentaron que, si los robots no se movían de forma consistente, resultaba muy problemático utilizarlos en las clases. Para solucionar este problema, hemos creado una forma de que el robot calibre los dos servomotores en el dispositivo físico.

ROVERSA puede, además, tomar diferentes formas a través de disfraces flexibles que pueden ser prediseñados, o pueden ser generados en el aula con materiales disponibles en contextos específicos. En la Figura 5, por ejemplo, se pueden ver tres disfraces diferentes diseñados para ROVERSA. El primero, en forma de tortuga, es en madera cortada con una cortadora láser y conectado a través de palillos de madera. Los otros dos son diseños en papel, que solo requieren de una impresión a color, y permite ofrecer la forma de diferentes animales (en este caso, un zorro y un gato), que los estudiantes pueden plegar y armar sobre ROVERSA.

**Fig. 5.** Disfraces para ROVERSA



### 4.3. Trabajo en Desarrollo

Lo que sigue en este proceso nos permitirá ampliar la funcionalidad de ROVERSA para satisfacer las necesidades de los estudiantes de secundaria y probar ROVERSA a una escala mucho mayor en las aulas. Tenemos previsto implementar funciones del diseño universal para el aprendizaje, como el soporte de múltiples idiomas y otras interfaces de entrada/salida con sonido y luz para apoyar a los niños en condición de discapacidad. Por ejemplo, después de que los estudiantes hayan pulsado su programa en los botones (es decir, adelante, adelante, giro a la derecha, adelante, adelante), ROVERSA puede mostrar su programa en la pantalla LED micro:bit (F-F-R-F-F) antes de moverse y tener sonidos de acompañamiento para señalar los diferentes comandos. Tenemos previsto explorar diferentes funcionalidades de los botones (por ejemplo, un botón puede crear un “bucle” o potencialmente “repetir” pasos) para ayudar a ampliar la funcionalidad para los grados 3-5. También ofreceremos la posibilidad de conectarnos a ROVERSA a través de bluetooth gracias a nuestra colaboración con Birdbrain technologies y utilizaremos

FinchBlox, un lenguaje de programación basado en bloques de código abierto para entornos de educación primaria. Planeamos entender cómo capturar y representar eficientemente los datos de registro de las interacciones de los niños con ROVERSA para apoyar el aprendizaje de los estudiantes.

## 5. Conclusiones y próximos pasos

En este trabajo, se presentan las acciones realizadas en búsqueda de una educación equitativa en las ciencias de la computación. Estas acciones propician un mayor entendimiento de la integración de las ciencias de la computación en diferentes contextos culturales, sociales y económicos, a través de tres estrategias: (I) investigar cómo aprenden los estudiantes; (II) generar capacidad en los docentes mientras investigamos cómo aprenden y cómo logran enseñarlo; y (III) desarrollar y distribuir dispositivos versátiles y accesibles.

En este estudio, se detallan las iniciativas clave implementadas para promover una educación en computación más equitativa y accesible. Estas acciones han contribuido significativamente a la comprensión de cómo integrar eficazmente las ciencias de la computación en diversos contextos culturales y socioeconómicos. Como pasos a seguir, se planea fortalecer aún más la investigación para analizar profundamente el impacto de la educación en computación en los entornos educativos. Además, se continuará con el desarrollo de programas de formación para docentes, evaluando continuamente la aplicación de los conocimientos y habilidades adquiridos por los educadores en sus prácticas de aula. Esta evaluación en curso permitirá una comprensión más del efecto de la formación en los docentes participantes del programa. Paralelamente, se van a desarrollar futuras versiones de ROVERSA, buscando optimizar su utilidad en el aula y asegurar su efectividad continua como herramienta educativa. Estos próximos pasos son fundamentales para seguir aportando a la integración de las ciencias de la computación y garantizar un acceso equitativo a las oportunidades educativas en este campo.

## 6. Referencias

- Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J. and Zagami, J. (2016) 'A K-6 computational thinking curriculum framework: Implications for teacher knowledge', *Journal of Educational Technology & Society*, 19(3), pp. 47-57. DOI: 10.2307/jeductechsoci.19.3.47.
- Arfé, B., Vardanega, T. and Ronconi, L. (2020) 'The effects of coding on children's planning and inhibition skills', *Computers & Education*, 148, 103807. DOI: 10.1016/j.compedu.2020.103807.
- Bakala, E., Gerosa, A., Hourcade, J. P. and Tejera, G. (2021) 'Preschool children, robots, and computational thinking: A systematic review', *International Journal of Child-Computer Interaction*, 29, 100337. DOI: 10.1016/j.ijcci.2021.100337.
- Bers, M. U., Flannery, L., Kazakoff, E. R. and Sullivan, A. (2014) 'Computational thinking and tinkering: Exploration of an early childhood robotics curriculum', *Computers & Education*, 72, pp. 145-157. DOI: 10.1016/j.compedu.2013.10.020
- Bers, M. U. (2017) *Coding as a Playground: Programming and Computational Thinking in the Early Childhood Classroom*. Routledge.
- Bers, M. U. (2018) *Coding as a Playground: Programming and Computational Thinking in the Early Childhood Classroom*. Routledge.
- Bers, M. U. (2019) 'Coding as another language: A pedagogical approach for teaching computer science in early childhood', *Journal of Computers in Education*, 6(4), pp. 499-528. DOI: 10.1007/s40692-019-00147-3.
- Caeli, E. N. and Bundsgaard, J. (2020) 'Computational thinking in compulsory education: A survey study on initiatives and conceptions', *Educational Technology Research and Development*, 68(1), pp. 551-573.
- Cheryan, S., Plaut, V. C., Handron, C. and Hudson, L. (2013) 'The stereotypical computer scientist: Gendered media representations as a barrier to inclusion for women', *Sex Roles: A Journal of Research*, 69(1-2), pp. 58-71. DOI: 10.1007/s11199-013-0296-x.
- Clarke-Midura, J., Kozlowski, J. S., Shumway, J. F. and Lee, V. R. (2021) 'How young children engage in and shift between reference frames when playing with coding toys', *International Journal of Child-Computer Interaction*, 28, 100250. DOI: 10.1016/j.ijcci.2021.100250.
- Clinging, D. (2006) 'Fostering computer science success among women and minorities', *Magazine of the Society of Women Engineers*, pp. 38-41.
- Clements, D. H. and Sarama, J. (2011) 'Early childhood mathematics intervention', *Science*, 333(6045), pp. 968-970. DOI: 10.1126/science.1204537.
- Curzon, P., Dorling, M., Ng, T., Selby, C. and Woollard, J. (2014) *Developing computational thinking in the classroom: a framework*. Swindon, GB: Computing at School.
- Darling-Hammond, L., Hyler, M. E. and Gardner, M. (2017) *Effective teacher professional development*.
- Espinal, A., Vieira, C. and Guerrero-Bequis, V. (2022) 'Student ability and difficulties with transfer from a block-based programming language into other programming languages: a case study in Colombia', *Computer Science Education*. DOI: 10.1080/08993408.2022.2140527.
- Fedorenko, E., Ivanova, A., Dhamala, R. and Bers, M. U. (2019) 'The language of programming: A cognitive perspective', *Trends in Cognitive Sciences*, 23(7), pp. 525-528. DOI: 10.1016/j.tics.2019.04.010.
- Flórez, F. B., et al. (2017) 'Changing a generation's way of thinking: Teaching computational thinking through programming', *Review of Educational Research*, 87(4), pp. 834-860. DOI: 10.3102/0034654317710096.

- Fullan, M. and Hargreaves, A. (2012) 'Reviving teaching with 'professional capital'', *Education Week*, 31(33), pp. 30-36.
- Grover, S. and Pea, R. (2013) 'Computational thinking in K-12: A review of the state of the field', *Educational Researcher*, 42(1), pp. 38-43. DOI: 10.3102/0013189X12463051.
- Guskey, T. R. (2000) *Evaluating Professional Development*. Corwin Press.
- Hill, C., Corbett, C. and St Rose, A. (2010) *Why so few? Women in science, technology, engineering, and mathematics*. American Association of University Women.
- Jatzlau, S. (2018) 'Facilitating computational thinking through code-free, block-based programming languages', *Journal of Computer Languages, Systems & Structures*, 52, pp. 1-11.
- Kallia, M. and Cutts, Q. (2022) 'Conceptual development in early-years computing education: A grounded cognition and action based conceptual framework', *Computer Science Education*, pp. 1-27. DOI: 10.1080/08993408.2022.2140527.
- Lavidas, K., Apostolou, Z. and Papadakis, S. (2022) 'Challenges and opportunities of mathematics in digital times: Preschool teachers' views', *Education Sciences*, 12(7), 459. DOI: 10.3390/educsci12070459.
- Leidl, K. D., Bers, M. U. and Mihm, C. (2017) 'Programming with ScratchJr: a review of the first year of user analytics', *Conference Proceedings of International Conference on Computational Thinking Education*, pp. 116-121.
- Moreno-León, J., Robles, G. and Román-González, M. (2019) 'On the Impact of Visual Programming Environments on the Initial Understanding of Programming Concepts: An Exploratory Study', *IEEE Access*, 7, pp. 105255-105269.
- Relkin, E., de Ruiter, L. E. and Bers, M. U. (2021) 'Learning to code and the acquisition of computational thinking by young children', *Computers & Education*, 169, 104222. DOI: 10.1016/j.compedu.2021.104222.
- SChute, E. (2009) 'Lack of diversity part of equation in STEM fields. Colleges try to increase numbers of women, minorities in science and engineering', *Pittsburgh Post-Gazette*.
- Seah, R. and Horne, M. (2020) 'The construction and validation of a geometric reasoning test item to support the development of learning progression', *Mathematics Education Research Journal*, 32(4), pp. 607-628. DOI: 10.1007/s13394-019-00273-2.
- Simon, D., Horne, M., Clements, D., Confrey, J., Maloney, A., Sarama, J., ... and Watson, A. (2017) 'Researching and using learning progressions (trajectories) in mathematics education', *Proceedings of the 41st Conference of the International Group for the Psychology of Mathematics Education*, pp. 109-136.
- Steedle, J. T. and Shavelson, R. J. (2009) 'Supporting valid interpretations of learning progression level diagnoses', *Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching*, 46(6), pp. 699-715. DOI: 10.1002/tea.20308.
- Trendov, M., Varas, S. and Zeng, M. (2019) *Digital technologies in agriculture and rural areas: status report*.
- Varma, R. (2010) 'Why so few women enroll in computing? Gender and ethnic differences in students' perception', *Computer Science Education*, 20(4). DOI: 10.1080/08993408.2010.527697.
- Vieira, C., Chiu, J. and Velasquez, B. (2023) 'Towards a learning progression of sequencing and algorithm design for five- and six-years-old children engaging with an educational robot', *Computer Science Education*. Published online October 12, 2023. DOI: 10.1080/08993408.2023.2255058.

- U, J. and Yang, W. (2023) 'A systematic review of integrating computational thinking in early childhood education', *Computers and Education Open*, 4, 100122. DOI: 10.1016/j.caeo.2023.100122.
- Wing, J. M. (2008) 'Computational thinking and thinking about computing', *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881), pp. 3717-3725. DOI: 10.1098/rsta.2008.0118.
- Yadav, A., Stephenson, C. and Hong, H. (2017) 'Computational thinking for teacher education', *Communications of the ACM*, 60(4), pp. 55-62. DOI: 10.1145/2994591.
- Yadav, A., Good, J., Voogt, J. and Fisser, P. (2017) 'Computational thinking as an emerging competence domain', in *Competence-based vocational and professional education: Bridging the worlds of work and education*, pp. 1051-1067. DOI: 10.1007/978-3-319-41713-4\_49.
- Yoon, K. S., Duncan, T., Lee, S. W. Y., Scarloss, B. and Shapley, K. L. (2007) 'Reviewing the evidence on how teacher professional development affects student achievement', *Issues & Answers. REL 2007-No. 033*. Regional Educational Laboratory Southwest (NJ1).