



Pensamiento computacional a través de STEAM con Sphero: Un ejercicio con estudiantes de primero de secundaria

[en] Computational thinking through STEAM with Sphero: An exercise with secondary school students

Sandra Erika Gutiérrez-Núñez

Universidad Autónoma de Chihuahua, México



Joam Manuel Rincón Zuluaga

Universidad Tecnológica de Ciudad Juárez, México



Recibido: 2023/05/14

Aprobado para publicación: 2024/06/21

Publicado: 2024/06/30

RESUMEN

Este estudio investiga cómo un curso con enfoque STEAM (Science, Technology, Engineering, Arts and Mathematics) con robots Sphero impacta el desarrollo del pensamiento computacional (PC) en preadolescentes de 11 a 13 años de Ciudad Juárez, Chih., México. La hipótesis principal es que el nivel de PC de los participantes en esta investigación mejorará significativamente después de tomar un taller STEAM con Sphero, con miras a responder: las condiciones de impacto, la influencia del diseño instruccional y los aspectos de mayor contribución. Para el desarrollo del estudio, participaron 67 preadolescentes y se caracteriza por ser un diseño experimental con evaluaciones de PC antes y después de la intervención. Los resultados indican que los participantes no mejoraron significativamente en PC, ya que no pudieron resolver los problemas del examen final, con lo cual, se identificó la necesidad de hacer explícitos los elementos del PC durante las clases, como identificación de problemas, abstracción, descomposición, reconocimiento de patrones, creación de algoritmos, depuración e iteración. El estudio concluye que es fundamental introducir explícitamente estos conceptos en las secuencias STEAM y asegurar suficiente práctica con cada concepto de programación, ya que la estructura tradicional de la clase de programación no prepara adecuadamente a los estudiantes, contribuyendo a la deserción escolar.

PALABRAS CLAVE

Robótica educativa, STEAM, pensamiento computacional, competencias del siglo XXI, preadolescentes, educación secundaria.

ABSTRACT

This study investigates how the STEAM workshop with Sphero robots impacts the development of computational thinking (CP) in preadolescents aged 11 to 13 years from Ciudad Juárez, Chih., Mexico. The main hypothesis is that the PC level of the participants in this research will improve significantly after taking a STEAM workshop with Sphero, with a view to answering: the impact conditions, the influence of the instructional design and the aspects of greatest contribution. For the development of the study, 67 preadolescents participated and it is characterized by an experimental design with PC evaluations before and after the intervention. The results indicate that the participants did not improve significantly in PC, since they could not solve the problems of the final exam, therefore, the need to make the elements of PC explicit during classes, such as problem identification, abstraction, decomposition, was identified pattern recognition, algorithm creation, debugging and iteration. The study concludes that it is essential to explicitly introduce these concepts into STEAM sequences and ensure sufficient practice with each programming concept, since the traditional structure of the programming class does not adequately prepare students, contributing to school dropouts.

KEYWORDS

Educational robotics, STEAM, computational thinking, 21st century skills, preteens, secondary education.



Cómo citar (APA 7a edición)

Gutiérrez-Núñez, S. E. y Rincón Zuluaga, J. M. (2024). Pensamiento computacional a través de STEAM con Sphero: Un ejercicio con estudiantes de primero de secundaria. *Revista Estudios de la Información*, 2(1), 4-26. <https://doi.org/10.54167/rei.v2i1.1567>

Introducción

En la era digital, el pensamiento computacional (PC) se ha convertido en una competencia esencial, permitiendo a los estudiantes descomponer problemas complejos y desarrollar soluciones eficaces; mientras que el enfoque educativo STEAM integra disciplinas clave, promoviendo un enfoque holístico en la educación que prepara a los estudiantes para los desafíos del siglo XXI. Durante la última década se ha dado un creciente interés en la incorporación del PC en contextos educativos, entre otras cosas porque se cree que es efectivo para despertar conciencias científicas y ayudar a la mejora y el crecimiento de los países al solventar las demandas laborales relacionadas con las TIC (Bocconi et al., 2016).

El PC tal como se estudia en este artículo, es un pensamiento sistémico transferible, derivado de procesos metodológicos de la ingeniería y las ciencias computacionales conformado por una serie de conocimientos, habilidades, rasgos y prácticas para la resolución de problemas, toma de decisiones y planificación de proyectos en diferentes campos y contextos (Gutiérrez-Núñez et al. 2024). El interés del sector educativo en el PC puede rastrearse con la famosa declaración de Wing (2006) en la que señala que este se trata de un conjunto de “habilidades y destrezas (herramientas mentales) habituales en los profesionales de las ciencias de la computación, pero que todos los seres humanos deberían poseer y utilizar para resolver problemas, diseñar sistemas y, sorprendentemente, comprender el comportamiento humano” (p. 33).

Desde entonces, múltiples investigaciones educativas se han esforzado en demostrar el potencial del PC para sentar las bases de un pensamiento más completo y complejo en los educandos (Huerta Jiménez y Velázquez Albo, 2021). Así, se espera que el PC ayude a las nuevas generaciones a desarrollar las competencias que les serán indispensables para su vida adulta, la cual será caracterizada principalmente por la expansión de la tecnología a todos y cada uno de los aspectos de la cotidianidad (Vázquez Acevedo et al., 2024).

La conceptualización de STEAM y de PC antes presentadas recogen acertadamente las propuestas del pensamiento complejo propuesto por Morin (2007), quien enfatiza la necesidad de integrar conocimientos de diversas disciplinas para entender fenómenos complejos promoviendo una visión holística, también el PC se beneficia del enfoque interdisciplinario del STEAM para resolver problemas. Ambos se centran en la comprensión de las relaciones y conexiones entre las partes de un sistema, reconociendo que el todo es más que la suma sus partes y aceptan la incertidumbre y las paradojas como parte inherente de la realidad y buscan estrategias para gestionarlas, a través de algoritmos probabilísticos, por ejemplo, de heurísticas y de aprendizaje automático.

Dentro de las competencias del siglo XXI se encuentran la resolución de problemas, la creatividad, la colaboración, la comunicación, la alfabetización y la ciudadanía digitales (Segura et al., 2019) y de acuerdo con la Organización de las Naciones Unidas estas competencias deben cultivarse en los estudiantes desde las etapas más tempranas de la educación formal (Scott, 2015).

Por su parte, el enfoque STEM, es un aprendizaje que integra ciencia, tecnología, ingeniería y matemáticas para desarrollar la creatividad en la resolución de problemas. Más tarde se le agregaría la A de artes para incorporar en este enfoque transdisciplinario a todas aquellas disciplinas que no se desprenden de las ciencias exactas. De acuerdo con Yakman (2008), el arte

es una extensión y abarca, además de las bellas artes, el lenguaje y algunas expresiones de las ciencias sociales o hasta de las artes físicas.

La transformación del proceso educativo en la época actual que requiere un alto nivel de tecnología hace del enfoque STEAM una de las claves del progreso y la innovación en el desarrollo de la competencia humana ([Pratama y Widjajanti, 2024](#)). De acuerdo con estos autores, los estudiantes estarán preparados para afrontar la evolución industrial del siglo XXI si desarrollan habilidades comunicativas y de colaboración, pensamiento crítico y habilidades creativas e innovadoras. El papel de la tecnología en la educación es importante porque se incluye como parte del plan de estudios, como un sistema de entrega de instrucción, como herramienta de instrucción y como herramienta para mejorar el proceso de aprendizaje ([Raja y Nagasubramani, 2018](#)) y la inclusión de la robótica en el salón de clases puede ayudar en el desarrollo de habilidades STEAM y favorecer la creatividad del estudiantado ([Scaradozzi et al., 2020](#)).

Este estudio adopta un diseño experimental, implementando evaluaciones de PC antes y después de la intervención para medir el impacto del programa de habilidades STEAM para identificar cambios en sus habilidades de PC, dado que se plantea la hipótesis de que los niños que participen en el taller STEAM con robots Sphero mostrarán una mejora significativa en sus habilidades de PC en comparación con su estado previo al taller.

Antecedentes

La investigación educativa que se ha desarrollado en torno al PC ha permitido llegar a dos conclusiones definitivas. La primera señala que el STEAM facilita la integración del PC en el currículo escolar debido a su interdisciplinariedad que promueve la vinculación entre los contenidos científicos, tecnológicos, ingenieriles, artísticos y matemáticos ([Raposo-Rivas, et al., 2022](#)). Además, al ser el trabajo por proyectos la base del enfoque educativo STEAM, éste favorece en gran medida la colaboración entre los diferentes miembros de un equipo para resolver un problema dado ([Pratama y Widjajanti, 2024](#)).

Las competencias asociadas al aprendizaje STEAM son: la autonomía y emprendimiento, colaboración, comunicación, conocimiento y uso de la tecnología, creatividad e innovación, diseño y fabricación de productos, pensamiento crítico y resolución de problemas ([Sánchez Ludeña, 2019](#)). La segunda conclusión importante en torno al desarrollo del PC reconoce a la programación como el medio idóneo para el desarrollo de las estructuras mentales que surgen en el programador como producto del uso de computadoras para resolver problemas de distinta índole ([García Rodríguez, 2022](#)).

Autores como [González et al. \(2018\)](#) y [González-Fernández \(2021\)](#) advierten que no basta con incluir la programación para desarrollar el PC, sino que dependerá en gran medida de las actividades que proponga el docente en torno a la programación para que este se desarrolle o no. En este sentido, la robótica ha probado ser un excelente aliado para que los niños en diferentes rangos de edad puedan adquirir habilidades de programación y de PC ([Sánchez Tendero et al., 2019](#)), siempre que los docentes cuenten con las habilidades y conocimientos para enseñar a los niños ([Camargo Pérez y Munar Landino, 2021](#)). Por otra parte, [Barrera Lombana \(2015\)](#), resalta el hecho de que al incluir la robótica y el PC en el currículo no se pretende que todos los estudiantes sean programadores profesionales, sino darles herramientas para que adquieran conocimientos más complejos y significativos despertando así el interés y la motivación por su propio proceso de enseñanza-aprendizaje a la par que mejoran sus habilidades de resolución de problemas en equipo.

La robótica presenta la ventaja de brindar retroalimentación inmediata a los niños y jóvenes acerca de los resultados del código que acaban de escribir ([Encalada Cuenca et al., 2024](#)),

lo que les permite valorar sus creaciones, identificar los errores y corregirlos, que son prácticas muy importantes del PC y que no deberían estar limitadas a una sola asignatura dentro del currículo, ya que esta división del conocimiento es contradictoria con el desarrollo de competencias y alfabetizaciones de corte transversal ([Valverde Berrocoso et al., 2015](#)).

Dependiendo de la edad de los niños y jóvenes, se elige el tipo de robot que se usará en la práctica, considerando su desarrollo cognitivo y las prestaciones del robot. En niños de kínder, los robots que se usan generalmente son los Bee Bots o los Blue Bots; para los niños de primaria alta y secundaria baja se utiliza Sphero y para los jóvenes de preparatoria se utilizan kits de Lego Mindstorms. Los niños de 11 a 13 años se encuentran en el estadio de operaciones formales, de acuerdo con la teoría psicogenética de Piaget, en la cual el pensamiento se caracteriza por ser más abstracto y lógico ([Gutiérrez Borda 2021](#)). Los niños de esta edad se encuentran en una etapa crucial de desarrollo cognitivo y emocional, donde la introducción de conceptos avanzados puede moldear significativamente su capacidad de pensamiento crítico y resolución de problemas ([Faroh, 2007](#)). Esta franja de edad es, por tanto, ideal para la implementación de estrategias educativas que fomenten el PC.

El fin último al buscar la integración curricular del PC a través del STEAM es provocar aprendizajes significativos en los estudiantes, de manera interdisciplinar y holística, para tratar de desarrollar habilidades de resolución de problemas en equipo ([Pratama y Widjajanti, 2024](#)), lo que ha demostrado que el uso de robots en el aula puede mejorar significativamente las habilidades computacionales y de resolución de problemas.

Modelos pedagógicos asociados a la robótica educativa

La integración del PC, el enfoque STEAM y la robótica educativa sientan las bases de la filosofía Maker ([Halverson y Sheridan, 2014](#)), la cual tiene su origen en el aprendizaje activo, el constructivismo y el construccionismo derivados de los aportes de [Piaget \(1977\)](#), [Papert \(1980\)](#), [Vygotsky \(1984\)](#) y [Dewey \(1929\)](#). De acuerdo con [Anderson \(2012\)](#) la aplicación del movimiento Maker, versión operativa del construccionismo en la educación, debe realizarse bajo el paradigma de aprender creando conocimiento a través del acto de construir, y debe realizarse en espacios en los que se unen las personas interesadas en la producción creativa e innovadora de todo tipo de artefactos digitales para compartir ideas y creaciones.

Mientras Piaget destaca el rol del aprendiz y cómo este adquiere un papel activo y dinámico con su espíritu crítico y constructivo, Vygotsky destaca la interacción entre estímulos y respuestas y describe que el aprendizaje real es consecuencia de la interacción social y la reconstrucción interna del sujeto, en lo que él enunció como zona de desarrollo próximo ([Huerta Gaytán, 2021](#)). Es así como en las últimas décadas del siglo XX, el proceso de aprendizaje centrado en el estudiante tomó un énfasis especial que dio origen al construccionismo de [Papert \(1980\)](#) como una evolución natural del constructivismo con una perspectiva colectiva y social.

De esta manera los estudiantes adquieren los contenidos curriculares de las distintas disciplinas al verlos materializados en proyectos que requieren un diseño, una investigación, una construcción, diferentes controles de mecanismos, por mencionar los más sobresalientes ([Sánchez Tendero et al., 2019](#)) lo que finalmente permite que desarrollen el tipo de pensamiento sistémico, estructurado, lógico y formal que comprendemos como PC. Sin embargo, es importante señalar que la robótica no es capaz de modificar el aprendizaje de los estudiantes por sí misma, sino que requiere de un modelo pedagógico efectivo, un ambiente de aprendizaje adecuado y de la guía del docente para lograr los objetivos de una enseñanza exitosa ([Cebrián de la Serna et al., 2021](#)).

Los robots Sphero poseen gran versatilidad y capacidad de programación, dado que ofrecen una plataforma interactiva que facilita el aprendizaje práctico. A través de actividades lúdicas y educativas, los robots Sphero permiten a los estudiantes explorar conceptos de programación y robótica, integrándose perfectamente en la metodología de trabajo con enfoque STEAM; sin embargo, hay una escasez de estudios centrados en el uso específico de robots Sphero a través del STEAM para desarrollar el PC en estudiantes de 11 a 13 años.

Objetivos de estudio y preguntas de investigación

El objetivo principal de este estudio es evaluar el impacto de la metodología STEAM utilizando robots Sphero en el desarrollo del PC en niños de 11 a 13 años. Los subobjetivos incluyen la evaluación de un diseño instruccional STEAM a partir de la plataforma del robot Sphero, mediante la aplicación de una prueba de PC antes y otra después de realizar el curso de PC a través del STEAM con robots Sphero.

Las preguntas que se plantearon para este ejercicio son:

- a) ¿Cuál es el impacto de la metodología STEAM utilizando robots Sphero en el desarrollo del pensamiento computacional (PC) en niños de 11 a 13 años?
- b) ¿Cómo influye el diseño instruccional STEAM basado en la plataforma del robot Sphero en el desarrollo de las habilidades de PC en estudiantes de secundaria baja?
- c) ¿Qué aspectos del enfoque STEAM contribuyen más significativamente al desarrollo del pensamiento computacional en los estudiantes?

Los resultados de este estudio proporcionarán valiosa información para educadores y diseñadores de currículos sobre la efectividad del enfoque STEAM para el desarrollo de los elementos del PC. Además, contribuirán al conocimiento sobre cómo promover el PC en los estudiantes primer grado de secundaria, sentando las bases para futuras investigaciones y aplicaciones prácticas.

Metodología

Aplicación del curso STEAM

El Centro de Inteligencia Artificial del estado de Chihuahua (IA.Center), en conjunto con el Instituto Promotor para la Educación del estado de Chihuahua (IPE), realizaron en el 2022 un programa piloto de habilidades STEAM para niños de secundaria de escuelas públicas ubicadas en Ciudad Juárez, con un nivel de dificultad básico.

El diseño instruccional del programa fue tomado del plan de estudios de un curso básico de habilidades STEAM mediante la plataforma de Sphero llamado *Computer Science Foundation Course 1 by Sphero*. El IA.Center adquirió y adaptó las 24 lecciones STEAM con Sphero para ser enseñadas a cuatro grupos de 20 participantes cada uno. Las clases del programa piloto se ejecutaron una vez por semana para cada grupo con un total de 24 semanas de curso. El programa piloto tuvo lugar entre febrero y julio del 2022.

Se respetó la estructura del curso de habilidades STEAM de la plataforma de Sphero. Cada una de las 24 lecciones tuvo una intensidad de dos horas con la siguiente estructura: (1) 5-10 minutos de exploración de la lección; (2) 25-30 minutos para la construcción de habilidades y desarrollo del reto; (3) 5-15 minutos para reflexionar y compartir resultados; y (4) 60 minutos para retos extendidos.

El curso de nivel básico de Sphero se divide en tres temas: (1) "A" en STEAM; (2) formas y números; y (3) naturaleza. Cada tema sigue la misma progresión de aprendizaje, llevando a los estudiantes del tablero Canva (dibujo), a los bloques básicos y por último a los bloques

intermedios, en cada una de las 8 lecciones de las tres temáticas. El formato de enseñanza propuesto por Sphero es de tipo ascensor, en el que los estudiantes suben y bajan de nivel de complejidad al pasar de las lecciones de un tema a las lecciones de otro tema.

Participantes

Los cursos de habilidades STEAM mediante la plataforma de Sphero nivel básico se impartieron a estudiantes de primero grado de secundaria, provenientes de dos instituciones: Adolfo López Mateos y Club de Leones, las cuales se seleccionaron por su anterior participación en este tipo de cursos y su apertura para que los niños asistan al Centro de Inteligencia Artificial en días y horas escolares. Es importante señalar que el Instituto Promotor de la Educación (IPE) se encarga de coordinar dicho traslado. En total fueron 80 estudiantes atendidos en este programa de los cuales 33 eran niñas y 47 niños. Los participantes fueron distribuidos en 4 grupos de 20 miembros cada uno, como se mencionó antes. La selección de los participantes dependió de la autorización de sus padres para el ingresar al programa.

Resultados de investigación

Evaluación del curso de habilidades STEAM y Pensamiento Computacional

Antes de iniciar el curso STEAM mediante la plataforma de Sphero nivel básico, se administró una prueba de 14 preguntas sobre PC a los 80 participantes. Al finalizar las 24 semanas del curso, se aplicó la misma prueba. Este enfoque permitió medir y comparar los conocimientos de los estudiantes antes y después del curso. Los resultados de ambas pruebas fueron analizados estadísticamente para evaluar la efectividad del programa en el desarrollo de habilidades de PC, determinando así si el curso de habilidades STEAM con Sphero es suficiente para ejercitar los elementos clave de este.

La prueba de pensamiento computacional fue adaptada y dosificada por los autores de este trabajo a partir de lo propuesto por [Román-González \(2015\)](#). La adaptación se realizó en virtud de que algunos reactivos de la prueba original presentan una dificultad muy alta para los jóvenes participantes en este estudio dado que algunos conceptos de programación que evalúa la prueba de PC de 28 reactivos no se estudian con tanta profundidad en la guía de Sphero nivel básico. En la Tabla 1 se muestra la prueba de pensamiento computacional aplicada antes y después del curso STEAM y el aspecto a evaluar del PC.

Tabla 1. *Prueba de pensamiento computacional para el curso STEAM*

Pregunta	Concepto computacional estudian						Existencia de anidamiento	Tarea requerida
	Dirección	Bucles (Loops)		Condicionales (conditionals)				
		Repetir veces	Repetir hasta	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)		
1	Sí	No	No	No	No	No	No	Secuenciación
2	Sí	No	No	No	No	No	No	Completamiento
3	Sí	No	No	No	No	No	No	Depuración
4	Sí	No	No	No	No	No	No	Secuenciación
5	Sí	No	Sí	Sí	No	No	Sí	Secuenciación
6	Sí	No	Sí	Sí	No	No	Sí	Secuenciación
7	Sí	Sí	Sí	Sí	No	No	Sí	Completamiento
8	Sí	No	Sí	Sí	No	No	Sí	Depuración

Pensamiento computacional a través de STEAM con Sphero: Un ejercicio con estudiantes de primero de secundaria

Pregunta	Concepto computacional estudian						Existencia de anidamiento	Tarea requerida
	Dirección	Bucles (Loops)		Condicionales (conditionals)				
		Repetir veces	Repetir hasta	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)		
9	Sí	No	Sí	No	Sí	No	Sí	Secuenciación
10	Sí	No	Sí	No	Sí	No	Sí	Secuenciación
11	Sí	No	Sí	No	Sí	No	Sí	Depuración
12	Sí	Sí	No	No	No	Sí	Sí	Secuenciación
13	Sí	Sí	No	No	No	Sí	Sí	Secuenciación
14	Sí	No	No	Sí	No	Sí	Sí	Completamiento

Modelo de pensamiento computacional aplicado

El curso de habilidades STEAM se realizó utilizando la plataforma tecnológica de Sphero dado que permite utilizar un tablero de dibujo, bloques de código y líneas de código para el desarrollo de las lecciones. Los diseños instruccionales de Sphero están alineados a las guías de la Computer Science Foundation (CSF). En este nivel básico se utilizan las herramientas de tablero de dibujo (programación del Sphero mediante dibujos en un tablero Canva) y bloques de programación básicos e intermedios (condicionales simples, ciclos de repetición simples, bloques de sonido, bloques de dibujos, colores y movimiento básico).

El nivel básico del curso de Sphero responde a las normas de aprendizaje de la Computer Science Teachers Association (CSTA) y el K-12 Computer Science Standard. En particular, se responde a los componentes de: P1. Promover una cultura informática inclusiva, P2. Colaborar en torno a la informática, P3. Reconociendo y definiendo problemas computacionales, P5. Creando artefactos computacionales, P6. Prueba y refinamiento de artefactos computacionales, P7. Comunicación acerca de la computación. En cuanto al diseño instruccional propuesto para las lecciones se Sphero, se tiene que cada lección cuenta con una planeación de cinco etapas mostrada en la Tabla 2.

Tabla 2. *Diseño instruccional de las clases de Sphero*

Componente	Descripción	Lecciones
1. Exploración	Estos pasos invitan a los estudiantes a explorar conceptos disciplinares y establecer la conexión con el entorno de programación de Sphero.	"A" en STEAM 1: Pictionary "A" en STEAM 2: Art Bot "A" en STEAM 3: gran poesía "A" en STEAM 4: Mi amigo, Hue "A" en STEAM 5: ¡Luces! ¡Cámara! ¡Acción! "A" en STEAM 6: Sphero Spirograph
2. Desarrollo de habilidades	Estos pasos guían a los estudiantes a través de progresiones de aprendizaje estructuradas sobre nuevos conceptos de programación.	"A" en STEAM 7: Coreografía de danza "A" en STEAM 8: revolución del baile Formas y números 1: Sphero se pone en forma Formas y números 2: mancha la forma Formas y números 3: mi número favorito es...
3. Desafío	Estos pasos les piden a los estudiantes que apliquen creativamente nuevas habilidades y conceptos para superar obstáculos y crear artefactos que demuestren los objetivos de aprendizaje de la lección.	Formas y números 4: conecta los números Formas y números 5: dados animados Formas y números 6: baile cuadrado Formas y números 7: Patinaje artístico Formas y números 8: Adivina mi número Naturaleza 1: La tortuga y la liebre Naturaleza 2: Polinización Naturaleza 3: Viaje al centro de la tierra

Pensamiento computacional a través de STEAM con Sphero: Un ejercicio con estudiantes de primero de secundaria

Componente	Descripción	Lecciones
4. Compartir	Estos pasos proporcionan instrucciones para compartir el trabajo con grupos pequeños o toda la clase	Naturaleza 4: Mimetismo Naturaleza 5: banco de peces Naturaleza 6: Ciclos del sueño Naturaleza 7: lo que necesita una semilla Naturaleza 8: ¿día o noche?
5. Reflexión	Estos pasos les piden a los estudiantes que revisen, sinteticen y procesen el aprendizaje principal de la lección para que puedan aplicarla a nuevas situaciones	
6. Desafío extendido	Estos pasos ofrecen sugerencias abiertas para ampliar el aprendizaje en una lección. Estos desafíos a menudo requerirán un segundo período de aprendizaje.	

Para evaluar si el curso de habilidades STEAM mediante la plataforma de Sphero es suficiente para ejercitar los componentes básicos del PC como lo proponen los autores de este artículo, se comparan, hasta donde los enfoques lo permiten, los elementos del diseño instruccional del curso de Sphero versus los elementos del PC. En la Tabla 3 se muestra el elemento del PC propuesto por los autores, el bloque de planeación según la guía de Sphero y su aplicación práctica en las clases.

Tabla 3. *Aplicación de los componentes del pensamiento computacional en el curso de Sphero*

Elemento del pensamiento computacional	Componente instruccional	Aplicación del componente de pensamiento computacional
Estadios: Identificación del problema, abstracción, descomposición, reconocimiento de patrones.	1. Exploración 2. Desarrollo de habilidades	La aplicación del elemento de pensamiento computacional se materializa en los componentes 1 y 2 debido a que en estas secciones se estudia un problema en concreto, se descompone en las actividades mínimas que se deben realizar y se identifican patrones para implementar soluciones.
Conceptos de programación: eventos, condicionales, bucles, etc.	3. Desafío	La aplicación de <i>conceptos de programación</i> se materializa en el componente 3 debido a que en esta sección se estudian todos los bloques funcionales requeridos en la lección y explica y pone en práctica los conceptos elementales, como lo son los bloques condicionales, ciclos de repetición, interrupciones, variables, funciones, entre otros.
Prácticas: creación de algoritmos, depuración, iteración.	4. Compartir 5. Reflexión 6. Desafío extendido	Las prácticas de pensamiento computacional se materializan en los componentes 4, 5 y 6 debido a que en esta sección los estudiantes crean los algoritmos que requieren para el reto planteado en la lección, deben ejecutar dicho algoritmo e ir depurando e iterando hasta lograr el objetivo de la lección

En el anexo 1 se muestra un ejemplo de una lección de nivel básico de Sphero llamada “Coreografía de danza” para intentar identificar los componentes del PC en el diseño instruccional de Sphero. La lección corresponde al número 7 del bloque temático de “A” en STEAM.

La prueba inicial fue aplicada a 80 niños, sin embargo, de esos 80 sólo 67 presentaron la prueba final porque algunos niños abandonaron el curso en algún momento. De modo que solo se contabilizaron los datos completos para el análisis estadístico. En el anexo 2 se encuentra la

tabla de los resultados de la prueba de PC antes y después del curso de habilidades STEAM con Sphero nivel básico.

Prueba de hipótesis de Wilcoxon

Se realizó una prueba de hipótesis unilateral a la derecha (mayor que) para saber si los estudiantes respondieron un mayor número de preguntas correctas en la prueba de PC después del curso de habilidades STEAM con Sphero, que antes de tomar el curso, con un nivel de significancia del 5%. La hipótesis nula y alternativa son las siguientes:

- H_0 : Los estudiantes obtienen estadísticamente el mismo número de preguntas correctas en una prueba de PC antes y después de tomar el curso de habilidades STEAM con Sphero.
- H_1 : Los estudiantes obtienen estadísticamente un número mayor de preguntas correctas en la prueba de PC después de tomar el curso de habilidades STEAM con Sphero.

La prueba de Wilcoxon es una prueba no paramétrica utilizada para muestras relacionadas cuando no se cumplen los supuestos de la estadística paramétrica. Es adecuada para comparar dos grupos relacionados (un mismo grupo en dos condiciones) y se aplica cuando la variable de resultado es ordinal o cuantitativa, pero no se cumple otro supuesto paramétrico, como la normalidad. A diferencia de las pruebas paramétricas que comparan medias, la prueba de Wilcoxon compara medianas, trabajando con el principio de ordenamiento. Se utiliza principalmente en situaciones donde las muestras son pequeñas o no distribuyen normalmente.

En el anexo 3 se muestra el código de Python utilizado para realizar la prueba de hipótesis de Wilcoxon ([SciPy, 2024](#)), entre las dos columnas de los resultados de las respuestas correctas de la prueba de PC antes y después de haber tomado el curso de habilidades STEAM con la plataforma Sphero. En el Código 1 se muestra el resultado de correr el programa en Python del algoritmo de la prueba de hipótesis de Wilcoxon con los datos mostrados en el anexo 1. Se puede observar en el resultado de la prueba que la hipótesis de Wilcoxon no se puede rechazar la hipótesis nula, así, no se puede presumir que el curso de habilidades STEAM ayuda a aumentar el número de respuestas correctas en las pruebas de PC.

Código 1. Resultados de la prueba de hipótesis de Wilcoxon en Python

```
Prueba de normalidad para respuestas antes del curso:
Estadístico W: 0.9501393437385559
Valor p: 0.009249907918274403

Prueba de normalidad para respuestas después del curso:
Estadístico W: 0.9536346197128296
Valor p: 0.013981513679027557
Las respuestas antes del curso NO siguen una distribución normal (p <= 0.05).
Las respuestas después del curso NO siguen una distribución normal (p <= 0.05).

No todas las columnas pasaron la prueba de normalidad. No es apropiado realizar
la prueba t de Student.
En su lugar, se realizará la prueba de Wilcoxon para muestras relacionadas.

Prueba de Wilcoxon (una cola, alternativa 'greater'):
Estadístico: 445.5
Valor p: 0.9993480167328705

No hay una diferencia estadísticamente significativa a favor de más respuestas
correctas después del curso según la prueba de Wilcoxon (p >= 0.05).
```

Análisis de las preguntas aplicadas

Aunado a la prueba de hipótesis de Wilcoxon en la que no se pudo evidenciar que hay un impacto positivo del curso con respecto al número de preguntas correctas obtenidas por los estudiantes en la prueba final de PC, la media de las respuestas correctas es un dato importante para analizar pues este no alcanza un valor mayor a las siete respuestas correctas (término promedio de respuestas correctas esperadas). Las respuestas obtenidas en las pruebas aplicadas antes y después del curso de habilidades STEAM con Sphero nivel básico se clasificaron en cuatro categorías. En la Tabla 4 se muestra el resultado en porcentaje de las cuatro categorías reportadas en el anexo 1.

Tabla 4. *Análisis de las categorías de las preguntas antes y después del curso STEAM*

No logra contestar bien la pregunta en la prueba ni antes y ni después del curso STEAM ▪ 392 preguntas ▪ 41.79%	Logra contestar bien la pregunta en la prueba antes y después del curso STEAM ▪ 250 preguntas ▪ 26.65%
No logra contestar bien la pregunta en la prueba después del curso STEAM, pero la había contestado bien antes de comenzar el curso ▪ 120 preguntas ▪ 12.79%	Logra contestar bien la pregunta en la prueba después del curso STEAM, pero la había contestado mal antes de comenzar el curso ▪ 176 preguntas ▪ 18.76%

En la Tabla 5 aplicada pueden atribuirse a la ejercitación de alguno de los aspectos del PC en el curso de habilidades STEAM con la plataforma Sphero. El 26.65% de las respuestas correctas de la prueba corresponde a personas que presumiblemente ya había tomado algún curso y/o tenía preparación con plataformas o contenido que trabaja los elementos del pensamiento computacional. El 12.79% de las respuestas de las pruebas la tuvieron presumiblemente personas que no comprenden los conceptos del PC en su totalidad ni antes, ni después del curso STEAM. Finalmente, el porcentaje más grande de respuestas de 41.79% se atribuyen a personas que ni antes, ni después del curso de habilidades STEAM con Sphero, lograron responder correctamente las preguntas de la prueba, lo que a su vez puede indicar que el curso STEAM no trabaja presumiblemente todos los elementos del PC.

En el anexo 2 se puede ver que las ocho preguntas denominadas P7, P13, P14, P8 P6, P4, P9, P11, en ese orden, son las que tienen el mayor número de errores al ser respondidas por los participantes tanto antes como después del curso STEAM, y son las que corresponden a la categoría naranja. Esto implica que más del 57% de las preguntas hechas en la prueba presumiblemente no son trabajadas a profundidad por el curso de habilidades STEAM con Sphero.

Si se contrasta el hecho anterior con la Tabla 1 se puede decir que las tareas requeridas de secuenciación, completamiento y depuración en anidamiento de las preguntas realizadas son las que menos se ejercitan en el curso de habilidades STEAM con Sphero. En particular, se observa que los conceptos de PC que se deben estudiar con mayor preponderancia en el curso de habilidades STEAM son: repetir hasta y mientras que de forma anidada y/o que incluyan condicionales dentro de la instrucción.

Por su parte, las preguntas que se responden muy bien, incluso antes del curso STEAM, son las preguntas P1, P2, P3, P5 y P12, que se encuentran en la categoría de color amarillo. Al

revisar la Tabla, se aprecia que las preguntas con actividades de secuenciación, completamiento y depuración sin anidamiento, que se evalúan en dichas preguntas, son los elementos del PC que menos dificultad presentan para los participantes del curso. Particularmente, los condicionales simples y un ciclo de repetición sin anidamiento son los más sencillos de comprender.

En las preguntas P3, P5, P6, P7, P8, P10 y P12 se observó, presumiblemente, que son las que contienen los elementos del PC mejor trabajado por el curso de habilidades STEAM mediante la plataforma de Sphero. Al revisar la Tabla 1 se observa que las actividades de secuenciación, completamiento y depuración con un anidamiento simple son los elementos de PC que más ejercita el curso STEAM con la metodología propuesta por Sphero. En particular el anidamiento de los condicionales simples con bucles de repetir hasta son los trabajados y comprendidos por parte de los participantes.

Conclusiones

Los cursos disponibles actualmente tanto de programación como de robótica para niños y jóvenes carecen de metodologías probadas para desarrollar el PC, lo que debe ser el objetivo principal de dichos cursos. Como ya se explicó antes, la proliferación de estos cursos se debe al potencial que posee el PC para el desarrollo de habilidades de pensamiento de orden superior y de resolución de problemas, sin embargo, la mayoría de ellos no procura el desarrollo de estas habilidades per se, sino que se concentran en mostrar a los niños y jóvenes cómo usar los bloques de programación y luego los dejan explorar por sí mismos.

Por más de una década se ha venido insistiendo en la necesidad de una definición unificada y operativa de lo que es el PC y de un currículo que indique puntualmente en qué orden se debe introducir cada concepto de programación, en qué profundidad según la edad del estudiantado, y cómo se deben introducir y complejizar las prácticas de PC inherentes al hecho concreto de programar. Muchos de los cursos de PC que se ofertan actualmente enseñan a los niños y jóvenes a utilizar plataformas de programación en bloque y los menús que poseen, pero no prestan atención a los procesos de pensamiento que deben desarrollarse a la par para la resolución de los retos que plantea la programación.

Si bien la programación desarrolla el PC, no es posible afirmar que este se desarrolla en los jóvenes solo por interactuar con las computadoras y los robots. Las dinámicas de clase, las actividades, los retos y la guía del docente juegan un papel determinante en la adquisición de la metacognición de los procesos mentales que se ponen en juego al programar, y el PC es esencialmente la metacognición de los procesos subyacentes al hecho de programar.

Es decir, para que el PC pueda ser transferible de una plataforma a otra y/o de un lenguaje de programación a otro, o de un contexto computacional a otro ámbito de la vida real, o de un campo profesional a otro, debe tener sus bases en la metacognición, la cual solo se desarrolla en los estudiantes con la práctica repetitiva del conjunto de elementos que se identifican como PC.

Los usuarios terminan familiarizándose con las plataformas de programación que manejan los diferentes cursos, pero no con el proceso subyacente a la programación misma ni a la resolución de problemas. Se observa gran cantidad de cursos que se apegan a distintos estándares, desde el DigComp en Europa hasta el currículum de la CSF en Estados Unidos, ambos para el desarrollo de competencias digitales, pero no para PC como tal, y esta constituye la primera falla en el diseño de esos cursos. Lo mismo ocurre con muchas de las intervenciones que buscan desarrollar el PC en las escuelas, dado que o bien, se apegan a alguno de estos marcos de referencia, o no clarifican a qué marco se están apegando para el diseño de las actividades de PC, ni de qué manera lo evaluaron.

En este trabajo se reporta la experiencia que se tuvo con jóvenes de secundaria que llevaron el curso básico de Sphero. Las secuencias STEAM que proponen las guías de Sphero se apegan a los estándares de la CSF, y plantean una serie de ejercicios para que los estudiantes practiquen la integración de bloques con objetivos específicos a través de la interacción con el robot, pero en ningún momento se hace explícita ni la práctica ni el elemento de PC que se está trabajando, o que se aplica para resolver el reto planteado.

En este estudio, los preadolescentes participantes no desarrollaron el PC, pues no fueron capaces de resolver los planteamientos que se les hicieron en el examen. Esto se debe a que si bien, el curso de Sphero tiene el potencial de desarrollar el PC en los estudiantes, esto no se logra porque es necesario hacer explícito en la clase cual es el problema o reto que hay que resolver (identificación del problema), cómo se debe estudiar el reto en virtud de la información que se posee o de la que falta (abstracción), identificar las partes más pequeñas en que puede descomponerse el reto (descomposición), recuperación de información sobre eventos similares y sus soluciones para ver cuales pueden aplicarse al problema actual (reconocimiento de patrones) y planteamiento de posibles soluciones (creación de algoritmos). Además de las prácticas de depuración e iteración.

Todos estos elementos se ponen en práctica durante las sesiones de Sphero, sin embargo, los participantes no los reconocen y por lo tanto no son capaces de utilizarlos a voluntad para resolver diferentes tipos de retos o problemas. Al aplicarles el examen final de PC se hizo evidente que los jóvenes no desarrollaron PC. Por lo tanto, es necesario introducir de manera explícita en las secuencias STEAM los elementos del PC que se desea desarrollar; así como asegurar una buena cantidad de práctica a los estudiantes con cada concepto de programación, porque si bien, estos deben estudiarse de manera progresiva en cuanto a complejidad, el proceso mental de prepararse para dar una respuesta computacional a un problema debe estudiarse en todas las sesiones con la misma rigurosidad.

Para esto, el docente debe procurar a los estudiantes un modelamiento claro del objetivo o producto final de la clase, debe nombrar explícitamente a qué concepto de programación está apelando o está utilizando, debe aplicar el protocolo de pensamiento en voz alta para compartir su razonamiento de resolución de problemas con los estudiantes, debe hacer uso de diagramas de flujo que representen los pasos o etapas necesarias para resolver el problema y por último, debe permitir que el estudiante comparta su propio proceso de razonamiento con la finalidad de identificar lagunas.

De igual modo, es fundamental la ejercitación constante y la práctica suficiente con cada nuevo concepto computacional estudiarse, pues la estructura tradicional de una clase de programación consiste en la presentación de una función, variable, o cualquier otro concepto de programación, seguido de una demostración por parte del docente sobre cómo aplicarlo y en seguida se solicita al estudiante que resuelva un problema para el que no ha sido preparado ni entrenado. A esto se debe en gran parte la deserción de estudiantes de estas carreras. Por lo tanto, son necesarias más lecciones con ejercicios básicos, hasta que el estudiante se sienta cómodo con este concepto y sienta la confianza de reconocerlo y aplicarlo en distintas situaciones.

La frecuencia y duración de las clases impactan fuertemente el desarrollo del PC debido a la ejercitación constante del alumnado. Si bien es cierto que el PC se integra muy bien en el STEAM, no se puede obviar el tiempo que se le debe dedicar en particular al desarrollo del PC. Éste debe poseer su propio espacio curricular en la escuela, tales como Español o Matemáticas, aunque al igual que estos dos, su impacto y aplicación sea transversal al resto de las disciplinas.


Referencias

- Anderson, C. (2012). *Makers. The New Industrial Revolution*. Crown Publishing Group. <https://www.hisour.com/es/educational-robotics-43004/>
- Barrera Lombana, N. (2015). Uso de la robótica educativa como estrategia didáctica en el aula. *Praxis & Saber*, 6(11), 215-234. <https://doi.org/10.19053/22160159.3582>
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A. y Engelhardt, K. (2016). *Developing Computational Thinking in Compulsory Education—Implications for policy and practice*. JRC Publications Repository. <https://doi.org/10.2791/792158>
- Camargo Pérez, A. J. y Munar Ladino, J. A. (2021). Habilidades del pensamiento computacional en docentes en formación de la universidad La Gran Colombia. *Revista Científica UISRAEL*, 8(2), 135-149. <https://doi.org/10.35290/rcui.v8n2.2021.441>
- Cebrián de la Serna, M., Huerta Gaytán, P., Flores Almendárez, J. M., González Fernández, M. O., Gómez Rodríguez, H., Ruíz Rey, F. J., Sánchez Nosea, J. M., González Flores, Y. A., Ornelas Sandoval, M. A., García Fuentes, O. y Cebrián Robles, V. (2021). *Robótica Educativa. Una perspectiva didáctica en el aula*. Universidad de Guadalajara. <http://repositorio.cualtos.udg.mx:8080/jspui/handle/123456789/1157>
- Dewey, J. (1929). *The quest for certainty*. Capricorn.
- Encalada Cuenca, J., Delgado Ramírez, J., y Arboleda Barrezueta, M. (Coordinadores). (2024). *Contextos educativos emergentes: Robótica educativa para estudiantes de educación general básica*. UTMACH. <http://repositorio.utmachala.edu.ec/handle/48000/22412>
- Faroh, A. C. (2007). Cognición en el adolescente según Piaget y Vygotski: ¿Dos caras de la misma moneda? *Boletín Academia Paulista de Psicología*, 27(2), 148-166. <https://www.redalyc.org/articulo.oa?id=94627214>
- García Rodríguez, A. (2022). Enseñanza de la Programación a través de Scratch para el desarrollo del Pensamiento Computacional en Educación Básica Secundaria. *Academia y Virtualidad*, 15(1), 161-182. <https://doi.org/10.18359/ravi.58832>
- González, J., Estebanell, M. y Peracaula, M. (2018). ¿Robots o programación? El concepto de pensamiento computacional y los futuros maestros. *EKS. Education in the Knowledge Society*, 19, 29–45. <https://doi.org/http://dx.doi.org/10.14201/eks20181922945>
- González-Fernández, M. O., González-Flores, Y. A. y Muñoz-López, C. (2021). Panorama de la robótica educativa a favor del aprendizaje STEAM. *Revista Eureka sobre Enseñanza y Divulgación de las Ciencias*, 18(2), 230101-230119. <https://orcid.org/0000-0002-1783-1113>
- Gutiérrez Borda, A. E. (2021). La edad de las operaciones formales de Jean Piaget y el rendimiento académico en matemáticas. *Ciencia Latina Revista Científica Multidisciplinar*, 5(4), 5864-5882. https://doi.org/10.37811/cl_rcm.v5i4.728
- Gutiérrez-Núñez, S. E., Rincón-Zuluaga, J. M., Castillo Ortega, M. F. y Serna Martínez, J. F. (2024). *Pensamiento computacional: Un pensamiento holístico* (Trabajo en proceso).
- Halverson, E. R. y Sheridan, K. (2014). *The maker movement in education*. *Harvard educational Review*, 84(4), 495-504. <https://doi.org/10.17763/haer.84.4.34j1g68140382063>
- Huerta Gaytán, P. (2021). Antecedentes y fundamentos de la robótica educativa. M. O. González Fernández (Coord.), *Robótica Educativa. Una perspectiva didáctica en el aula* (pp. 10-23). Universidad de Guadalajara. http://repositorio.cualtos.udg.mx:8080/jspui/bitstream/123456789/1157/1/Libro_Robotica_interactivo.pdf
- Huerta Jiménez, C. S. y Velázquez Albo, M. (2021). Pensamiento computacional como una habilidad genérica: Una revisión sistemática. *Ciencia Latina Revista Científica Multidisciplinar*, 5(1), 1055-1078. https://doi.org/10.37811/cl_rcm.v5i1.311

- Morin, E. (2007). *La cabeza bien puesta* (Trad. Paula Mahler). Buenos Aires, Argentina: Ediciones Buena Visión.
- Papert, S. (1980). *Mindstorms: Children, Computers and Powerful ideas*. Harvester Press.
- Piaget, J. (1977). *The development of thought: Equilibration of cognitive structures*. Viking.
- Pratama, K. A. y Widjajanti, D. B. (2024). STEM: Its Potential in Developing Students' Computational Thinking. *KnE Social Sciences*, 1074-1083. <https://doi.org/10.18502/kss.v9i13.16033>
- Raja, R. y Nagasubramani, P. C. (2018). Impact of modern technology in education. *Journal of Applied and Advanced Research*, 3, 33-35. <https://doi.org/10.21839/jaar.2018.v3iS1.165>
- Raposo-Rivas, M., García-Fuentes, O. y Martínez-Figueira, M.-E. (2022). La robótica educativa desde las áreas STEAM en educación infantil: Una revisión sistemática de la literatura (2005-2021). *Revista Prisma Social: Revista de Ciencias Sociales*, (38), 94-113. <https://revistaprismasocial.es/article/view/4779>
- Román-González, M. (2015). *Test de Pensamiento Computacional: Principios de diseño, validación de contenido y análisis de ítems*. [https://www.researchgate.net/publication/288341872_Test_de_Pensamiento Computacional principios de diseño validación de contenido y análisis de ítems Computational Thinking Test design guidelines content validation and item analysis](https://www.researchgate.net/publication/288341872_Test_de_Pensamiento_Computacional_principios_de_diseño_validación_de_contenido_y_análisis_de_ítems_Computational_Thinking_Test_design_guidelines_content_validation_and_item_analysis)
- Sánchez Ludeña, E. (2019). La educación STEAM y la cultura «maker». *Padres y Maestros / Journal of Parents and Teachers*, (379), 45-51. <https://doi.org/10.14422/pym.i379.y2019.008>
- Sánchez Tendero, E., Cózar Gutiérrez, R. C., y González-Calero Somoza, J. A. (2019). Robótica en la enseñanza de conocimiento e interacción con el entorno. Una investigación formativa en Educación Infantil. *Revista Interuniversitaria de Formación del Profesorado*, 33(1), 11-28. <https://doi.org/10.47553/rifop.v33i1.72087>
- Scaradozzi, D., Cesaretti, L., Screpanti, L. y Mangina, E. (2020). Identification of the Students Learning Process During Education Robotics Activities. *Frontiers in Robotics and AI*, 7, e21. <https://doi.org/10.3389/frobt.2020.00021>
- SciPy. (2024). *SciPy documentation*. <https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.wilcoxon.html>
- Scott, C. L. (2015). *El futuro del aprendizaje 2: ¿Qué tipo de aprendizaje se necesita en el siglo XXI?* UNESCO. https://unesdoc.unesco.org/ark:/48223/pf0000242996_spa
- Segura, J. A., Llopis Nebot, M. Á., Mon, F. E. y Valdeolivas Novella, M. G. (2019). El debate sobre el pensamiento computacional en educación. *RIED-Revista Iberoamericana de Educación a Distancia*, 22(1), 171-186. <https://doi.org/10.5944/ried.22.1.22303>
- Valverde Berrocoso, J., Fernández Sánchez, M. R. y Garrido Arroyo, M. C. (2015). El pensamiento computacional y las nuevas ecologías del aprendizaje. *Revista de Educación a Distancia (RED)*, (46), 1-18. <https://revistas.um.es/red/article/view/240311>
- Vázquez Acevedo, H. M., Licona Suarez, L. J. y Felizzola Medina, L. D. (2024). Pensamiento Computacional: Una competencia del siglo XXI: Revisión sistemática en Scopus. *Revista Latinoamericana Ogmios*, 4(9), 1-16. <https://doi.org/10.53595/rlo.v4.i9.090>
- Vygotsky, L. S. (1984) Aprendizaje y desarrollo intelectual en la edad escolar. *Infancia y Aprendizaje*, 7(27-28), 105-116. <https://doi.org/10.1080/02103702.1984.10822045>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. <https://doi.org/10.1145/1118178.1118215>
- Yakman, G. (2008). *ST Σ @M Education: An overview of creating a model of integrative education*. [https://www.researchgate.net/publication/327351326_STEAM Education an overview w of creating a model of integrative education](https://www.researchgate.net/publication/327351326_STEAM_Education_an_overview_of_creating_a_model_of_integrative_education)

Anexo 1. Ejemplo de una secuencia STEAM con Sphero

Elemento del pensamiento computacional	Componente instruccional	Ejemplo de una lección de nivel intermedio
<ul style="list-style-type: none"> ▪ Identificación del problema: abstracción, descomposición, reconocimiento de patrones. ▪ Conceptos de programación: eventos, condicionales, bucles, etc. ▪ Practicas: creación de algoritmos, depuración, iteración 	<ol style="list-style-type: none"> 1. Exploración 2. Desarrollo de habilidades 3. Desafío 4. Compartir 5. Reflexión 6. Desafío extendido 	<p>Lección 7: Coreografía de danza En esta lección, utiliza el Bucle X Veces y Bloques de Bucle Hasta para que tu robot Sphero haga la coreografía de cierta canción. Dibuja las Tarjetas de Coreografía para dictar el orden del baile, y ajústalo para coincidir con la música durante la exploración abierta. Finalmente, termina esta lección con tiempo de diseño abierto para agregar saltos y Bloques de Habla a tu rutina de baile.</p> <p>Tu robot Sphero puede bailar a lo largo de cualquier melodía, pero ¿Qué dibujarás de las Tarjetas de Coreografía? Pon tus movimientos de baile en los Controles de Bucle para que coincida con la música.</p> <p>Objetivos de aprendizaje</p> <ul style="list-style-type: none"> • Ajustar mi código para coincidir con la música. • Decir la diferencia entre Bucle X Veces y Bloque de Bucle Hasta. • Experimentar con diferentes Bloques de Luces, Sonidos y Movimientos. • Utilizar un Comparador en un Bloque de Control. <p>Robots compatibles</p> <div style="display: flex; justify-content: space-around; align-items: center; text-align: center;"> <div data-bbox="654 1035 789 1171"></div> <div data-bbox="849 1056 938 1150"></div> <div data-bbox="1003 1035 1138 1171"></div> </div> <p>Sphero mini Bolt</p> <p>Normas de aprendizaje Computer Science Teachers Association (CSTA) K-12 Computer Science Standards, Revised 2018.</p> <ul style="list-style-type: none"> • P2. Colaborando acerca de la computación. • P5. Creando artefactos computacionales. <p>Contenido: Arte.</p> <p>Actividad de estudiantes: No detengas la música Escucha la canción que vas a coreografiar. Piensa en el ritmo y como programarás tu robot Sphero para que la baile.</p> <p>¿Así que, crees que puedes bailar? Tal vez puedas, pero ¿puedes programar un robot Sphero para que se mueva con algo de música con coreografías complejas?</p> <p>Escuchemos una canción juntos como clase, ¡esto es lo que estamos coreografiando! Mientras escucha, piensa en el ritmo.</p> <p>¡Escucha la música!</p>

Elemento del pensamiento computacional	Componente instruccional	Ejemplo de una lección de nivel intermedio
		 <p>Actividad de estudiantes: Comparadores y sensores Construye un comparador usando el sensor de tiempo transcurrido. Ponlo dentro de un bloque de bucle hasta para programar tu robot Sphero y que baile hasta que la canción acabe.</p> <p>¿Cómo podemos decirle a nuestro robot Sphero que haga estos movimientos hasta que termine la canción? ¡Todo está en el tiempo!</p> <p>En la última lección, Espirógrafo, usaste el bloque de bucle infinito. En este programa, usarás un bucle hasta bloquear. Puedes programar un robot Sphero para que repita hasta que haya pasado una cierta cantidad de tiempo, la duración de tu canción!</p> <p>Un bloque de bucle hasta-solo ejecutará los bloques internos hasta que se cumpla una condición específica. Esta condición, algo que debe ser cierto para continuar, la establece un comparador. Los comparadores comparan dos valores: uno recopilado de los sensores (el reloj de un robot) con un valor establecido por el programador, tú.</p> <p>Construirás un comparador utilizando el sensor de tiempo transcurrido.</p> <p>Realiza lo siguiente:</p> <ol style="list-style-type: none"> 1. Pregúntale a tu profesor cuánto dura en segundos nuestra canción de baile. 2. Desde el menú Controles, arrastra el bucle hasta el bloque verdadero en tu lienzo. 3. Desde el menú Comparadores, arrastra el bloque de comparadores al campo hexagonal del bucle hasta el bloque. 4. Desde el menú Sensores, arrastra el sensor de tiempo transcurrido a la burbuja izquierda del bloque de comparadores. 5. Cambia el centro a mayor o igual que ($> =$). Esto le dice al robot Sphero que se detenga en esa cantidad de segundos o más. 6. Cambia la burbuja de comparación de la derecha a la cantidad de segundos que desea pasar antes de que el robot Sphero se detenga. 7. Arrastra los bloques de coreografía para que el bucle hasta que el bloque contenga todas las demás coreografías.

Elemento del pensamiento computacional	Componente instruccional	Ejemplo de una lección de nivel intermedio
		<p>8. Inicia el robot Sphero. 9. Prueba y depura tu código. Recuerda: El sensor de tiempo transcurrido mide el tiempo en segundos desde que se inicia el programa.</p> <p>Comprendiendo lo aprendido Selecciona la respuesta correcta ¿Cuál no es verdad? “Basado en la biblioteca de sensores, Podrías usar un comparador para decirle a tu robot Sphero que haga un bucle hasta...”</p> <p>— “...Viajó más que una distancia dada”. — “... Sintió que la temperatura era menor que una cantidad dada”. — “... Sintió que su rumbo fue de 0°”. — “... Sintió que la luminosidad era menor que una cantidad dada”.</p> <p>Actividad de estudiantes: Competencia de baile Explora los bloques de habla y como usar el motor en bruto y bloques de retraso para hacer que tu rutina sobresalga. ¿Qué ideas tienes para mejorar el programa de dados? Consejo del profesor</p> <div style="display: flex; justify-content: space-around;"> <div data-bbox="641 966 828 1071"> <p>Hablar</p> </div> <div data-bbox="933 966 1396 1113"> <p>Saltar</p> </div> </div> <p>Es casi la hora del baile ... si otros grupos no están listos, ¡perfecciona tu programa hasta que esté listo para ganar! Durante la competencia, buscarás otro robot Sphero cuyo programa se destaque para ti, así que presta atención.</p> <p>Realiza lo siguiente:</p> <ol style="list-style-type: none"> 1. Cuando todos los grupos estén listos, inicia la música y sus programas al mismo tiempo. 2. ¡Observa el baile! 3. Mantén un registro de un robot Sphero cuya rutina te gustaría destacar. <p>Reflexión grupal:</p> <ul style="list-style-type: none"> ▪ ¿De quién es la rutina de baile que merece una victoria (además de la tuya, por supuesto)? ▪ ¿Cómo crees que los programadores utilizan el sensor de tiempo transcurrido en el mundo que nos rodea? ▪ ¿Crees que algún "código" funciona con temporizadores en tu escuela? <p>Escribe una respuesta corta. En 2-3 oraciones, explica qué rutina de baile de robot merece una victoria y por qué.</p>

Anexo 2. Tabla de resultados de la prueba de pensamiento computacional al programa de STEAM con Sphero

SEXO	Preguntas																												Análisis entre prueba inicial y final							
	Movimiento								Mezcla de ciclos de repetición y condicionales																				Pretest	Postest	Diferencia	No logra contestar nunca bien	Contesta bien antes y después	Contesta bien al inicio, mal al final	Contesta mal inicio, bien al final	
	P1		P2		P3		P4		P5		P6		P7		P8		P9		P10		P11		P12		P13		P14									
	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D								
H	1	1	1	1	1	1	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	5	6	1	7	4	1	2	
M	1	1	1	1	1	1	0	1	1	1	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	5	8	3	5	4	1	4
H	1	1	1	1	1	0	1	0	1	1	0	1	1	0	0	1	0	1	0	0	1	0	1	0	0	0	0	0	8	6	-2	3	3	5	3	
M	0	1	1	0	0	0	0	0	1	0	1	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	4	5	1	7	2	2	3	
H	1	1	1	1	1	1	0	0	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	9	10	1	4	9	0	1	
H	1	1	1	1	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	1	1	1	1	1	0	0	0	8	5	-3	4	3	5	2		
H	1	1	1	1	0	1	0	1	1	1	1	1	0	0	0	0	1	0	0	1	0	1	0	1	0	0	0	5	9	4	4	4	1	5		
M	1	1	1	0	0	1	0	0	1	0	0	0	0	1	1	0	0	0	1	1	0	0	0	1	0	0	0	5	5	0	6	2	3	3		
H	1	0	1	0	1	1	0	1	0	1	1	1	0	0	0	1	0	0	0	1	0	1	1	1	0	0	0	5	8	3	4	3	2	5		
M	1	1	1	1	1	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	7	6	-1	7	6	1	0		
M	0	1	1	1	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	4	5	1	7	2	2	3		
H	1	1	1	1	0	0	1	0	0	0	0	0	0	1	1	0	0	0	1	1	1	0	0	0	0	0	7	5	-2	6	4	3	1			
M	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	3	2	-1	11	2	1	0		
H	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	1	0	1	1	0	0	0	10	11	1	2	9	1	2		
H	1	1	0	1	0	0	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	1	1	0	0	0	3	7	4	7	3	0	4			
H	1	1	1	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	6	3	-3	8	3	3	0		
H	1	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	4	2	-2	10	2	2	0			



Pensamiento computacional a través de STEAM con Sphero: Un ejercicio con estudiantes de primero de secundaria

SEXO	Preguntas																												Análisis entre prueba inicial y final							
	Movimiento								Mezcla de ciclos de repetición y condicionales																				Pretest	Posttest	Diferencia	No logra contestar nunca bien	Contesta bien antes y después	Contesta bien al inicio, mal al final	Contesta mal inicio, bien al final	
	P1		P2		P3		P4		P5		P6		P7		P8		P9		P10		P11		P12		P13		P14									
	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D						
H	1	1	1	1	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	3	5	2	8	2	1	3
H	1	1	1	1	1	1	0	0	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	8	7	-1	5	6	2	1
H	1	1	1	1	0	1	0	1	1	1	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	5	7	2	7	5	0	2	
H	1	1	1	1	1	1	1	1	0	1	0	0	0	1	1	0	0	0	0	1	0	1	1	1	1	0	0	1	7	10	3	2	4	2	6	
H	1	1	1	1	1	1	1	0	1	0	0	0	1	0	1	1	0	0	0	0	0	0	1	1	0	1	0	1	7	8	1	3	4	3	4	
H	1	1	1	1	0	0	1	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	0	4	7	3	7	4	0	3	
H	1	1	1	1	1	1	1	1	0	1	1	1	0	1	1	0	1	0	1	1	1	1	0	1	0	0	1	10	9	-1	1	6	4	3		
H	1	1	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	3	6	3	6	1	2	5	
H	1	1	1	1	1	1	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	1	4	9	5	5	4	0	5	
H	1	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	1	1	1	1	0	0	1	1	1	0	0	8	7	-1	5	6	2	1		
M	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	2	4	2	9	1	1	3		
M	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	1	0	0	1	0	1	0	1	0	0	0	0	7	5	-2	6	4	3	1		
H	1	1	1	1	0	1	1	1	1	0	1	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	7	7	0	4	4	3	3	
M	1	1	1	1	1	0	1	1	1	1	0	1	0	0	1	0	0	0	1	1	0	0	1	0	1	0	0	8	7	-1	3	4	4	3		
H	1	1	1	1	1	1	1	0	1	0	1	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0	0	5	9	4	5	5	0	4		
H	1	1	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	6	7	1	5	4	2	3		
M	0	1	0	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	3	7	4	6	2	1	5		
M	1	1	1	0	0	0	1	0	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	0	0	0	7	3	-4	5	1	6	2		
M	1	1	1	1	0	1	1	1	1	0	0	0	0	0	0	1	0	1	1	0	0	0	0	1	0	0	0	5	8	3	5	4	1	4		



Pensamiento computacional a través de STEAM con Sphero: Un ejercicio con estudiantes de primero de secundaria

SEXO	Preguntas																												Análisis entre prueba inicial y final						
	Movimiento								Mezcla de ciclos de repetición y condicionales																				Pretest	Posttest	Diferencia	No logra contestar nunca bien	Contesta bien antes y después	Contesta bien al inicio, mal al final	Contesta mal inicio, bien al final
	P1		P2		P3		P4		P5		P6		P7		P8		P9		P10		P11		P12		P13		P14								
	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D					
M	1	1	1	0	1	1	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	0	0	6	5	-1	6	3	3	2	
M	1	1	1	0	0	1	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	3	5	2	8	2	1	3	
H	1	1	1	1	0	1	0	0	1	1	1	0	0	0	0	0	1	0	1	0	1	0	1	1	1	1	1	8	8	0	3	5	3	3	
M	1	1	1	1	0	1	0	0	1	1	0	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	6	5	-1	7	4	2	1	
H	1	1	0	1	1	1	0	0	0	1	0	0	0	0	1	0	0	0	1	1	0	0	1	1	0	0	0	5	6	1	7	4	1	2	
M	1	1	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	1	0	0	3	5	2	7	1	2	4	
H	1	1	0	0	0	0	1	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	3	5	2	9	3	0	2	
H	1	1	1	1	1	1	0	0	1	0	0	0	0	0	1	0	1	1	0	1	1	0	1	0	0	0	0	5	7	2	5	3	2	4	
M	1	1	0	1	0	1	0	0	0	0	1	0	0	1	0	0	1	1	0	0	0	0	1	1	0	0	1	5	6	1	6	3	2	3	
M	1	1	1	0	0	0	0	0	0	1	0	1	1	1	0	0	1	0	0	0	1	1	1	0	0	0	1	5	7	2	5	3	2	4	
M	1	1	1	1	0	1	0	0	1	1	0	1	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	5	6	1	6	3	3	2	
M	1	1	1	1	1	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	5	4	-1	8	3	2	1	
H	1	1	1	1	1	1	1	0	1	1	1	1	0	1	0	0	1	1	1	1	0	0	0	0	0	0	0	8	8	0	5	7	1	1	
H	1	1	1	1	0	1	0	0	1	1	0	0	0	0	1	0	1	0	0	0	1	1	0	0	0	0	0	6	5	-1	6	3	3	2	
M	1	1	1	1	1	1	1	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	5	5	0	7	3	2	2	
M	1	1	0	1	1	1	0	0	0	1	0	1	0	0	0	0	0	1	1	1	1	0	0	0	0	0	5	8	3	5	4	1	4		
M	1	1	1	1	1	1	0	0	1	1	0	0	0	0	0	0	1	0	1	0	1	1	0	0	0	1	7	6	-1	5	4	3	2		
H	1	1	1	1	0	1	0	0	0	1	0	0	0	0	1	0	1	1	1	1	0	1	1	0	0	0	6	8	2	5	5	1	3		
H	1	1	1	1	0	1	0	0	1	1	1	1	0	0	0	0	1	1	0	1	1	0	1	0	0	0	6	9	3	5	6	0	3		



Pensamiento computacional a través de STEAM con Sphero: Un ejercicio con estudiantes de primero de secundaria

SEXO	Preguntas																												Análisis entre prueba inicial y final						
	Movimiento								Mezcla de ciclos de repetición y condicionales																				Pretest	Postest	Diferencia	No logra contestar nunca bien	Contesta bien antes y después	Contesta bien al inicio, mal al final	Contesta mal inicio, bien al final
	P1		P2		P3		P4		P5		P6		P7		P8		P9		P10		P11		P12		P13		P14								
	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D							
H	1	1	1	1	1	1	0	0	1	1	0	0	0	0	0	1	1	0	1	0	1	1	1	0	1	0	1	6	1	5	3	6	0	5	
H	1	1	1	1	1	1	0	0	1	1	0	1	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	4	8	4	6	4	0	4	
H	1	1	1	1	1	1	0	0	1	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	6	5	-1	7	4	2	1		
M	1	1	1	1	1	1	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	5	5	0	7	3	2	2		
M	1	1	1	1	1	1	0	0	1	1	1	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	6	6	0	7	5	1	1		
H	1	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	3	5	2	9	3	0	2		
M	1	1	1	1	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	6	6	0	6	4	2	2		
M	1	1	1	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	3	5	2	9	3	0	2		
H	1	1	1	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	5	5	0	7	3	2	2		
H	1	1	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	1	5	6	1	5	2	3	4		
H	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	4	5	1	8	3	1	2		
M	1	1	1	1	1	1	0	1	1	1	1	0	0	0	0	1	0	0	0	1	1	0	0	0	1	0	8	7	-1	4	5	3	2		
67	0		2		13		35		8		38		51		44		29		25		27		19		51		50		3	4	5	392	250	120	176
	63		49		29		11		29		9		1		1		12		12		8		22		3		1		5.	6.	0.	5.85	3.73	1.79	2.63
39	1		10		7		13		10		7		3		11		16		9		11		12		4		6		1.	1.	2.	1.96	1.63	1.33	1.40
28	3		6		18		8		20		13		12		11		10		21		21		14		9		10		3.	3.	4.	3.77	2.61	1.75	1.94

Notas para interpretar la tabla:

- AN, DN:** A es la prueba que se aplica antes de comenzar el curso y D es la prueba que se aplica después de terminar el curso. El N es el número de la pregunta, por ejemplo, A1, D1 son la prueba aplicada antes y después del curso STEAM con Sphero para la pregunta 1.



Anexo 3. Código en Python para realizar la prueba de hipótesis de Wilcoxon

```
import numpy as np
import scipy.stats as stats
import pandas as pd
# Datos extraídos de las columnas del anexo 1
pretest = [
    5, 5, 8, 4, 9, 8, 5, 5, 5, 7, 4, 7, 3, 10, 3, 6, 4, 3, 8, 5, 7, 7, 4, 10, 3, 4, 8, 2, 7, 7,
    8, 5, 6, 3, 7, 5, 6, 3, 8, 6, 5, 3, 3, 5, 5, 5, 5, 5, 8, 6, 5, 5, 7, 6, 6, 6, 4, 6, 5, 6, 3, 6,
    3, 5, 5, 4, 8
]
postest = [
    6, 8, 6, 5, 10, 5, 9, 5, 8, 6, 5, 5, 2, 11, 7, 3, 2, 5, 7, 7, 10, 8, 7, 9, 6, 9, 7, 4, 5, 7,
    7, 9, 7, 7, 3, 8, 5, 5, 8, 5, 6, 5, 5, 7, 6, 7, 6, 4, 8, 5, 5, 8, 6, 8, 9, 11, 8, 5, 5, 6, 5, 6,
    5, 5, 6, 5, 7
]
# Paso 2: Prueba de normalidad (Shapiro-Wilk)
stat_antes, p_antes = stats.shapiro(pretest)
stat_despues, p_despues = stats.shapiro(postest)
print("Prueba de normalidad para respuestas antes del curso:")
print("Estadístico W:", stat_antes)
print("Valor p:", p_antes)
print("\nPrueba de normalidad para respuestas después del curso:")
print("Estadístico W:", stat_despues)
print("Valor p:", p_despues)
# Evaluación de normalidad
alpha = 0.05
if p_antes > alpha:
    print("Las respuestas antes del curso siguen una distribución normal (p > 0.05).")
else:
    print("Las respuestas antes del curso NO siguen una distribución normal (p <= 0.05).")
if p_despues > alpha:
    print("Las respuestas después del curso siguen una distribución normal (p > 0.05).")
else:
    print("Las respuestas después del curso NO siguen una distribución normal (p <= 0.05).")
# Evaluación de si se puede proceder con la prueba t de Student
if p_antes > alpha and p_despues > alpha:
    print("\nAmbas columnas pasaron la prueba de normalidad. Procedemos con la prueba t de Student.")
    # Paso 3: Prueba t de Student
    t_stat, p_value = stats.ttest_rel(pretest, postest)
    print("\nPrueba t de Student:")
    print("Estadístico t:", t_stat)
    print("Valor p:", p_value)
    # Conclusión
    if p_value < 0.05:
        print("\nHay una diferencia estadísticamente significativa entre las medias antes y después del curso (p < 0.05).")
    else:
        print("\nNo hay una diferencia estadísticamente significativa entre las medias antes y después del curso (p >= 0.05).")
else:
    print("\nNo todas las columnas pasaron la prueba de normalidad. No es apropiado realizar la prueba t de Student.")
    print("En su lugar, se realizará la prueba de Wilcoxon para muestras relacionadas.")
    # Paso 3: Prueba de Wilcoxon de una cola
    wilcoxon_stat, wilcoxon_p = stats.wilcoxon(pretest, postest, alternative='less')
    print("\nPrueba de Wilcoxon (una cola, alternativa 'less'):")
    print("Estadístico:", wilcoxon_stat)
    print("Valor p:", wilcoxon_p)
    # Conclusión
    if wilcoxon_p < 0.05:
        print("\nHay una diferencia estadísticamente significativa a favor de más respuestas correctas después del curso según la prueba de Wilcoxon (p < 0.05).")
    else:
        print("\nNo hay una diferencia estadísticamente significativa a favor de más respuestas correctas después del curso según la prueba de Wilcoxon (p >= 0.05).")
```