

<https://doi.org/10.23913/ride.v12i23.1079>

Artículos científicos

Desarrollo del pensamiento computacional en niñas y niños usando actividades desconectadas y conectadas de computadora

Development thinking computing to children use of unplugged and plugged computing

Desenvolvimento do pensamento computacional em meninas e meninos usando atividades de computador desconectadas e conectadas

Carlos Enríquez Ramírez

Universidad Politécnica de Tulancingo, México

carlos.enriquez@upt.edu.mx

<https://orcid.org/0000-0003-4963-9828>

Mariza Raluy Herrero

Universidad Politécnica de Tulancingo, México

mariza.raluy@upt.edu.mx

<https://orcid.org/0000-0001-8516-1337>

Luz María Vega Sosa

Universidad Politécnica de Tulancingo, México

luz.vega@upt.edu.mx

<https://orcid.org/0000-0001-6355-4172>

Resumen

Actualmente, una de las competencias que más debe ser desarrollada desde edades tempranas es el pensamiento computacional, el cual procura resolver problemas por medio de la computadora, de ahí que demande la maduración de diversas habilidades cognitivas. Por tanto, el objetivo de este trabajo fue mostrar datos cuantitativos de la aplicación de un taller de programación en bloques con la finalidad de incorporar las habilidades básicas del pensamiento computacional en una muestra de alumnos de la escuela primaria rural Bartolomé Vargas Lugo, del municipio de Acatlán, Hidalgo. La pregunta que guio el presente estudio fue la siguiente: ¿cuál es el grado de aceptación de un curso de programación en bloques combinando computación conectada y desconectada? El diseño de la investigación fue mixto y la metodología cuantitativa, mientras que el enfoque del estudio fue exploratorio y descriptivo. El referente tomado para el desarrollo de las competencias del pensamiento computacional en los participantes del taller fue la filosofía constructivista, mediante el uso de recursos didácticos usados para fortalecer los nuevos conocimientos en el área de la programación de computadoras. Como resultados se identifica la aceptación del uso de la plataforma, de las actividades previas para adoptar conocimientos nuevos en el desarrollo de un programa de computadoras y de la búsqueda de información con respecto al tema. Con base en los resultados de las encuestas aplicadas, se puede indicar que se observaron algunas diferencias en cuanto al género, pues las niñas tienen más curiosidad de aprendizaje sobre la herramienta de programación y visualizan sus beneficios. También se observa en los resultados que los niños y niñas muestran una poca o nula resistencia en la adquisición de los conocimientos experimentados en el taller de programación y establecen mecanismos de confianza en la resolución de problemas basados en computadora, o bien en la interpretación y uso de estructuras de control. Por ello, para futuros trabajos se puede aplicar este tipo de actividades en escuelas distantes de la zona urbana con la finalidad de establecer comparativos.

Palabras clave: educación básica, habilidades, informática educativa, lenguaje de programación, programación informática, técnicas didácticas.

Abstract

The objective of this work was to show quantitative data from the application of a block-based computer programming workshop in order to incorporate the basic skills of computational thinking in a sample of Bartolomé Vargas Lugo rural elementary school students from the municipality of Acatlán, Hidalgo where there is a number to consider of participating students who do not have computer equipment at home to solve this characteristic, the course was developed in the following two hours after their normal departure from classes. The question that guided the present study was the following: what is the degree of acceptance of this course combining connected and disconnected computing? The research design was mixed, and the methodology used was quantitative, while the focus of the study was descriptive and exploratory. The reference that is taken for the development of computational thinking competencies in the workshop participants is the constructivist philosophy through the use of didactic resources in connected and disconnected computer programming. As results, acceptance is identified both in the use of the platform, the activities prior to the use of the computer, the understanding of the activities prior to the development of a computer program and a relevant point the search for information regarding programming. These data can be seen reflected in a questionnaire with a Likert scale. In addition, there are some differences in this workshop course in the stages of development of programs at the gender level where girls are more curious about learning in the programming tool, they visualize the benefits of learning programming among some others. As observed in the results, the boys and girls show little or no resistance in acquiring the knowledge experienced in the developed workshop, such as establishing trust mechanisms in some future that are presented again with concepts such as problem-solving based in computer or in the interpretation and use of control structures. Within the previous works it is intended to carry out this concept of disconnected activities for the acquisition of computational thinking in remote schools from the urban area in order to establish a comparison in the knowledge of the same.

Keywords: Basic education, skills, educational informatics, programming language, programming informatics, didactic techniques.

Resumo

Atualmente, uma das competências que mais se deve desenvolver desde cedo é o pensamento computacional, que busca resolver problemas por meio do computador, portanto exige o amadurecimento de várias habilidades cognitivas. Portanto, o objetivo deste trabalho foi mostrar dados quantitativos da aplicação de uma oficina de programação em blocos para incorporar as habilidades básicas de pensamento computacional em uma amostra de alunos da escola primária rural Bartolomé Vargas Lugo, do município de Acatlán, Hidalgo. A questão que norteou o presente estudo foi a seguinte: qual o grau de aceitação de um curso de programação em bloco combinando computação conectada e desconectada? O desenho da pesquisa foi misto e a metodologia quantitativa, enquanto o foco do estudo foi exploratório e descritivo. A referência assumida para o desenvolvimento das habilidades de pensamento computacional nos participantes das oficinas foi a filosofia construtivista, por meio da utilização de recursos didáticos utilizados para fortalecer os novos conhecimentos na área de programação de computadores. Os resultados são a aceitação do uso da plataforma, as atividades anteriores de adoção de novos conhecimentos no desenvolvimento de um programa de computador e a busca por informações sobre o assunto. Com base nos resultados das pesquisas aplicadas, pode-se indicar que algumas diferenças foram observadas em relação ao gênero, uma vez que as meninas têm mais curiosidade em aprender sobre a ferramenta de programação e visualizar seus benefícios. Observa-se também nos resultados que meninos e meninas apresentam pouca ou nenhuma resistência na aquisição dos conhecimentos vivenciados na oficina de programação e estabelecem mecanismos de confiança na resolução de problemas informatizados, ou na interpretação e utilização de estruturas de controle. . Portanto, para trabalhos futuros, este tipo de atividades pode ser aplicado em escolas distantes da zona urbana para estabelecer comparações.

Palavras-chave: educação básica, habilidades, computação educacional, linguagem de programação, programação de computadores, técnicas de ensino.

Fecha Recepción: Mayo 2021

Fecha Aceptación: Noviembre 2021

Introducción

Es evidente que la sociedad actual se encuentra sumergida en una cambiante era tecnológica, lo cual ha obligado a la creación de esquemas de enseñanza que promuevan el desarrollo de habilidades que se ajusten a las demandas actuales. Por eso, como señalan Espino y González (2015), se debe orientar a los niños y niñas para que cultiven ciertas capacidades digitales esenciales para la vida cotidiana y laboral.

En tal sentido, según un estudio realizado por Salesforce Economy, para el año 2024 se estima un aumento anual de 44 % en puestos especializados de inteligencia artificial y 38 % en los empleos relacionados con el internet de las cosas (Gantz, 2019), conceptos novedosos en el área tecnológica. Por ese motivo, resulta vital incorporar en edades tempranas el uso y el desarrollo de tecnología, como sucede en diversas naciones (p. ej., como EE. UU, Reino Unido, Nueva Zelanda, Corea del Sur e Israel), donde han incluido dentro de sus programas educativos la enseñanza de la programación de computadoras (Mannila *et al.*, 2014).

Entre las competencias que los niños y niñas deben adquirir en la proliferación de las tecnologías de la información y comunicación (TIC) se encuentran el análisis de información, la representación de esta mediante abstracciones (tales como modelos y simulaciones), la automatización de soluciones empleando el pensamiento algorítmico, y la generalización y transferencia del proceso de resolución de problemas mediante una computadora. Estas competencias se engloban dentro de lo que Wing (2006) denomina *pensamiento computacional* (PC). Su desarrollo no solo aminora los costos y sus efectos son más duraderos que los iniciados en etapas tardías, sino que también ayudan a sortear muchos obstáculos al momento de ingresar a un campo de trabajo específico (González-González, 2019).

En el presente trabajo, por ende, se emplean diversas técnicas didácticas en un taller de programación de computadoras para fomentar climas de confianza y creatividad donde los participantes adquieran los contenidos del PC desde un rol constructorista. Al respecto, cabe destacar que el constructorismo entiende a la tecnología y particularmente a la computadora como unas estructuras ideales donde el niño puede genuinamente construir su conocimiento desde “un sentido de dominio sobre un elemento de la tecnología moderna y de las ideas más profundas de la ciencia, la matemática y el arte de construcción de modelos intelectuales” (Papert, 1987, pp. 17-18). Esta referencia a la filosofía

constructivista se hace para abordar el desarrollo del pensamiento computacional utilizando recursos didácticos en la programación de computadora conectada y desconectada en alumnos de educación básica.

Trabajos relacionados

El PC está siendo ampliamente estudiado debido a la tendencia actual que aboga por la inclusión de la ciencia informática en instancias cada vez más tempranas de la educación formal. Para ello, los investigadores y maestros están empleando ejercicios de programación desde una computadora (es decir, actividades conectadas), así como estrategias didácticas que permitan entender conceptos del entorno de las ciencias de la computación sin la necesidad de usar una computadora (es decir, actividades desconectadas). El uso de actividades desconectadas como recurso didáctico para transmitir los conceptos de PC se justifica para tener un aprendizaje significativo a través de experiencias kinestésicas, lo cual tiene un menor costo y beneficia a los niños que carecen de equipos informáticos (Huang y Looi, 2021; Venkatesh, Das y Das, 2021). Esto, según Papert (1980), se puede aplicar mediante el uso de determinados juegos.

Otro de los precursores de la programación desconectada (PD) es Tim Bell (Bell, Alexander, Freeman y Grimley 2009), quien explica que este tipo de aprendizaje no solo simula los procesos que realiza una computadora, sino que también brinda a los estudiantes la posibilidad de explorar las ideas fundamentales de la CS. En concordancia con esta idea, la Universidad de Canterbury (25 de febrero de 2020) promueve un entorno constructivista donde los participantes reciben desafíos basados en reglas sencillas que permiten descubrir por propia cuenta poderosas ideas de CS. En pocas palabras, no solo constituye una forma más de aprendizaje significativo, pues también los empodera para darse cuenta de que son ideas que están a su alcance.

Asimismo, investigadores en el campo de la enseñanza de PC afirman que el uso de PD facilita el aprendizaje de conceptos relacionados con el CS, como lo señalan Hufad, Faturrohman y Rusdiyani (2021), quienes trabajaron con niños de entre 5 y 6 años de edad en la resolución de problemas mediante ejercicios de PD. En su investigación, identificaron que las habilidades de resolución de problemas mejoran si las actividades son lúdicas, pues esto mantiene motivados a los niños. Lógicamente, cada estrategia varía dependiendo de su

naturaleza. Por ejemplo, Threekunprapa y Yasri (2021) se basaron en la realidad aumentada para enseñar diagramas de flujo a un grupo de jóvenes de una escuela secundaria.

Otro de los factores de estudio de la PD es identificar las variables que influyen en el aprendizaje de habilidades del PC, como lo explican Hsu, Chang y Hung (2018), quienes midieron la participación por género en la implementación de un taller de programación, pues —según Dagiene, Pelikis y Stupurienė (2015)— existen diferencias entre hombres y mujeres para desarrollar la habilidad de resolver un problema a través de los referidos recursos tecnológicos. De hecho, en el trabajo por Torres-Torres, Román-González y Pérez-González (2019) se describen algunas actividades no desconectadas usadas con un grupo de mujeres no sólo para acercarlas a temas de ciencia, tecnología, ingeniería y matemáticas, sino también con el fin de que pudieran crear códigos más complejos que los elaborados por los hombres. La motivación, por tanto, es otro factor de estudio para el logro de la adquisición de habilidades del PC cuando se emplean actividades PD (Brackmann *et al.*, 2017; Del Olmo, Cózar-Gutierrez y González-Calero, 2020).

Método

Se realizó un taller de programación por bloques. Para ello, se usó el lenguaje de programación Scratch (Fagerlund, Häkkinen, Vesisenaho y Viiri, 2021) y se emplearon recursos didácticos aplicados a la programación conectada y desconectada con un grupo de 22 estudiantes de sexto grado de educación básica (11 niñas y 11 niños).

Asimismo, se utilizó un diseño experimental mixto para identificar la aceptación y el uso de la programación mediante bloques y para determinar el grado de adquisición de conceptos y habilidades del pensamiento computacional.

Para registrar las actividades finales efectuadas por los estudiantes se utilizaron grabaciones en video (Imler y Eichelberger, 2011), lo cual facilita la observación del desarrollo de las prácticas, especialmente en todo lo relacionado con uso correcto de las estructuras de control, procesos como la definición de variables, elementos de entrada de datos y visualización de los resultados.

Como punto inicial se aplicó una encuesta para identificar el uso de equipos de cómputo en casa. Los resultados demuestran que 73.68 % de los participantes en el taller carecen de equipos de cómputo. Por eso, se decidió efectuar un taller durante 10 semanas

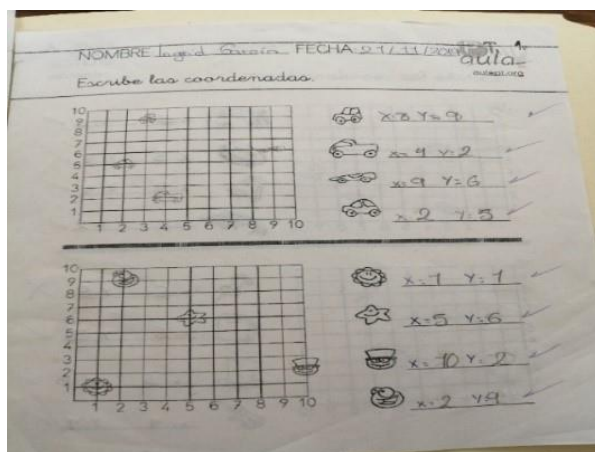
(dos horas por sesión) después de las horas de estudio. Para concretar este taller en dicho horario se solicitó la autorización a los padres.

La planeación de las actividades tuvo la estructura siguiente: objetivo por cumplir en cada sesión, desglose de los temas, materiales y estrategias para el desarrollo de actividades conectadas y desconectadas, las cuales se describen a continuación.

Primera sesión. Se explicó a los participantes los conceptos de algoritmo de computadora, instrucción, pasos o actividades. De este modo, el niño podía identificar la construcción de estos y describir un acontecer cotidiano mediante el uso de esta herramienta. Luego, el equipo de trabajo se encargó de jugar con el algoritmo en conjunto con los niños y niñas.

Segunda sesión. Los alumnos iniciaron con la interacción de la herramienta de programación. Esto permite conocer algunas sentencias elementales para mover objetos en el entorno de programación Scratch. Se trabajó previamente con actividades desconectadas a la computadora para adquirir o reforzar el concepto *coordenada* por medio de elementos para ubicar en un cuadrante (figura 1).

Figura 1. Trabajo de coordenadas para ubicar objetos en un plano



Fuente: Elaboración propia

Posteriormente, se usó el entorno de trabajo Scratch, y por medio de la programación se movió un objeto por diferentes puntos, lo que permitió involucrar el uso de la programación conectada. Se enseñaron instrucciones del entorno para mover imágenes mediante pequeños códigos con la finalidad de explicar elementos básicos que se podían usar durante el curso.

Tercera sesión. Nuevamente se presentaron actividades desconectadas, como el juego de dados, para explicar el proceso de sumatoria. Mediante esta actividad se inició el conocimiento del concepto de *ciclo* en una actividad conectada donde los alumnos identificaron las instrucciones y aprendieron a construir sentencias de repetición.

Cuarta sesión. Se trabajó el concepto *variable* en la actividad desconectada. Para ello, se usaron bolsas de papel para representar las fases de creación de una de ellas, es decir, nombrado e inicialización, dependiendo el contenido de la bolsa (galletas o dulces) (figura 2).

Figura 2. Aprender el concepto *variable*



Fuente: Elaboración propia

Identificadas las fases de la creación de las variables, se pasó a la plataforma de programación para implementar desde la herramienta los pasos siguientes para el uso de variables.

Quinta sesión. Se trabajó con actividades conectadas para dar a conocer el concepto *bloque si-entonces*, mediante un programa propuesto por el instructor donde el alumno replica el mecanismo de la instrucción (figura 3).

Figura 3. Actividad conectada mostrando el uso de sentencia *si-entonces*

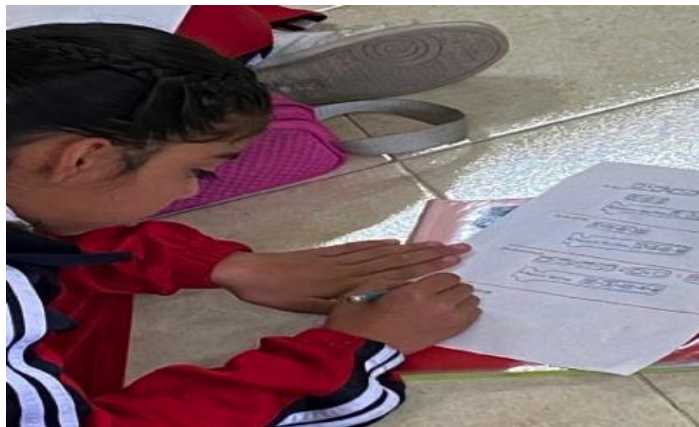


Fuente: Elaboración propia

Sexta sesión. Se utilizaron actividades desconectadas para explicar la composición de expresiones aritméticas básicas, las cuales empezaron a construir los propios niños mediante operadores aritméticos (figura 4).

Por otra parte, se usó la plataforma para identificar la sección donde se ubicaban los operadores aritméticos, y se explicó cómo se debían llevar a cabo la construcción de algunas expresiones aprendidas en la PD.

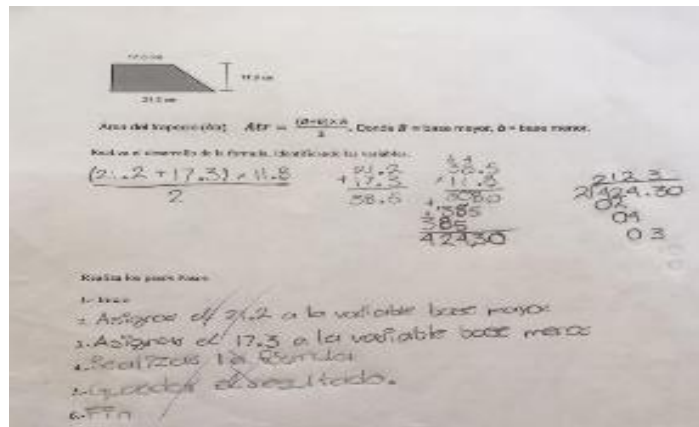
Figura 4. Alumna desarrollando expresiones aritméticas en el leguaje por bloques



Fuente: Elaboración propia

Séptima sesión. En esta se empezaron a conjuntar los conocimientos adquiridos en las actividades vistas para introducirlos al concepto de pseudocódigo y construcción de la solución en el entorno de Scratch. Además, se inició el proceso de verificación y validación como parte del desarrollo de *software* (figura 5).

Figura 5. Desarrollo de primer pseudocódigo en fase de evaluación del taller



Fuente: Elaboración propia

Las actividades anteriores reforzaron los conceptos de pensamiento computacional desde las competencias de abstracción, pensamiento algorítmico y verificación, mediante las estrategias que fueron explicadas por medio de bloques en la plataforma de programación.

En la siguiente sección se describen los hallazgos recabados a lo largo del taller de programación.

Resultados

Como parte del estudio de los resultados, se elaboró un cuestionario para evaluar el grado de aceptación de las actividades conectadas y desconectadas desarrolladas (tabla 1).

Tabla 1. Resultado de aplicación de cuestionario

#	Pregunta	Totalmente en desacuerdo	En desacuerdo	Ni de acuerdo ni en desacuerdo	De acuerdo	Totalmente de acuerdo
1	Aprendo varios conceptos nuevos de programación en cada sesión que tenemos.	4.55 %	0.00	4.55 %	64 %	27.26 %
2	Aprendo en este curso de Scratch por qué me puede servir en el futuro	0.00	4.55 %	0.00	3 %	81.82 %
3	Entiendo los pasos que debemos usar para hacer un programa en Scratch	4.55 %	0 %	9.08 %	54.55 %	31.82 %
4	Si hay algo que no comprendo durante el curso, lo pregunto a mis amigos	9.09 %	4.55 %	22.73 %	18.18 %	40.91 %
5	Prefiero que los profesores me expliquen todo lo relacionado a la actividad	9.09 %	0.00 %	18.18 %	18.18 %	54.55 %
6	Podría realizar yo solo, sin ayuda de los demás, un programa en Scratch	4.55 %	4.55 %	22.73 %	54.55 %	13.64 %
7	Encuentro relación de las actividades previas con lo que hicimos en la computadora	9.09 %	0.00 %	4.55 %	59.09 %	27.27 %
8	Me gustan las actividades previas al uso de la computadora en el curso	0.00 %	0.00 %	0.00 %	18.18 %	59.09 %
9	Se me hacen aburridas las actividades previas al uso de la computadora	36.36 %	0.00 %	9.09 %	31.82 %	18.18 %
10	Se me hace difícil el desarrollar un programa en Scratch	18.18 %	0.00 %	9.09 %	31.82 %	22.73 %
11	Me gusta buscar información de Scratch	0.00 %	0.00 %	4.55 %	18.18 %	54.55 %
12	Me gusta aprender programación con Scratch	0.00 %	0.00 %	0.00 %	9.09 %	86.36 %
13	Me interesa aprender más sobre el tema de Scratch	0.00 %	0.00 %	0.00 %	13.64 %	72.73 %

#	Pregunta	Totalmente en desacuerdo	En desacuerdo	Ni de acuerdo ni en desacuerdo	De acuerdo	Totalmente de acuerdo
14	Quiero seguir aprendiendo más del tema de programación de computadoras	0.00 %	0.00 %	4.55 %	13.64 %	81.82 %
15	Me agradaría estudiar una carrera en sistemas computacionales	13.64 %	0.00 %	9.09 %	31.82 %	45.45 %

Fuente: Elaboración propia

Como elemento de validación del instrumento de medición, se calculó el alfa de Cronbach de 0.717, resultado que se obtuvo con el *software* de SPSS. Para realizar una mejor lectura de los resultados se propusieron los segmentos (ver tabla 2) para agrupar las preguntas.

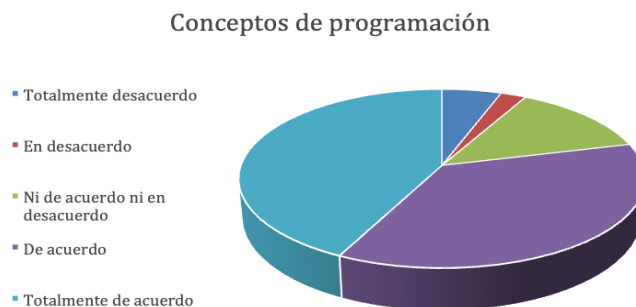
Tabla 2. Segmento de preguntas.

Segmento	Pregunta
I	1, 2, 3, 4, 5 y 6
II	7, 8 y 9
III	10, 11, 12, 13
IV	14 y 15

Fuente: Elaboración propia

En el segmento 1 se agruparon las preguntas 1, 2, 3, 4, 5 y 6, las cuales hacen referencia al aprendizaje y al desarrollo de los conceptos de PC. En la figura 6 se visualiza que la mayoría de los participantes están de acuerdo o totalmente de acuerdo, lo que permite establecer la existencia de conceptos de este tipo de pensamiento en los participantes del curso.

Figura 6. Segmento relacionado al aprendizaje de conceptos de PC



Fuente: Elaboración propia

En las preguntas del segmento 2 —referente a la programación desconectada— solamente un porcentaje pequeño aún está totalmente en desacuerdo con su uso para adquirir los conceptos del PC (figura 7).

Figura 7. Gráfica del segundo segmento referente al uso de actividades previas



Fuente: Elaboración propia

El tercer hace referencia a las actividades que se llevaron a cabo en la fase de programación conectada como puede verse en la figura 8. Donde se observa el interés hacia el uso de la plataforma y además les causa curiosidad lo cual resuelven indagando más sobre la misma.

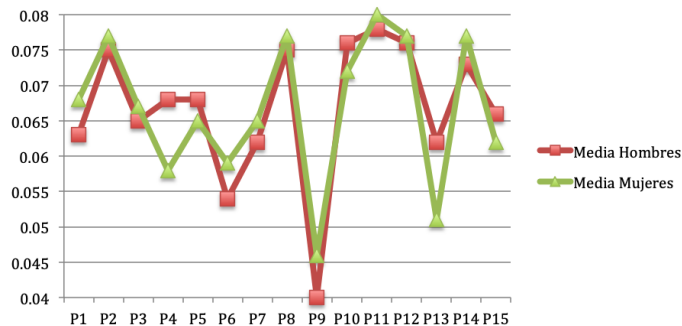
Figura 8. Gráfica del tercer segmento referente a las actividades conectadas.



Fuente: Elaboración propia

Por otra parte, el análisis de la información por género se enseña en la figura 9. Véase la aceptación del taller por parte de las niñas.

Figura 9. Resultados por género



Fuente: Elaboración propia

Las participantes al taller muestran una independencia en el manejo de los conceptos del PC, así como una autonomía en la creación de programas en la plataforma. De hecho, se presentan con mayor confianza elaborando de manera independiente las actividades desconectadas, así como mayor aceptación en comparación con los varones.

En el taller de programación se utilizaron ejercicios de diversos tipos con animaciones (p. ej., movimientos de los objetos de manera aleatoria y otros controlados por parte del usuario). Además, programas que permiten una interacción entre el usuario y la computadora mediante la entrada de datos y la visualización de los resultados de un proceso.

Para evaluar el logro de las competencias esperadas en el taller de programación se usó la taxonomía de SOLO (*Structure of the Observed Learning Outcome*), es decir, estructura del resultado observado de aprendizaje (Biggs y Collis, 1982). La taxonomía

define jerárquicamente las categorías del aprendizaje, esto es, clasifica los resultados de programación de computadoras en preestructural, uniestructural, multiestructural, relacional y abstracto, lo que permite analizar la calidad del aprendizaje desde los niveles más concretos hasta los más abstractos y complejos.

Los niveles de SOLO se adaptan para dar una contextualización de los rubros evaluados en cada una de las actividades realizadas en la fase final del taller de la siguiente manera.

Preestructural (N1). Los programas contienen solamente pocos bloques y el estudiante no entiende su significado.

Uniestructural (N2). Los programas tienen secuencias de acción de una manera simple. No se encuentran relaciones entre las estructuras de control.

Multiestructural (N3). El programa cumple con todos los requisitos dados e incluye una variedad de diferentes tipos de bloques. El código puede haber sido reorganizado para hacer una solución más integrada.

Relacional (N4). Se proporciona un programa bien estructurado que elimina toda redundancia y tiene una estructura clara y lógica.

Abstracto (N5). El programa utiliza conceptos y bloques más allá de los ejercicios para proporcionar mejoras a la solución.

En el taller de programación por bloques se cuenta con un total de 11 Scripts (forma como se conoce a un programa en el entorno de Scratch) con un promedio de 14.09 bloques por proyecto, es decir, sentencias de programación que se usan para el logro de un objetivo. Los 16 bloques con mayor frecuencia de uso por parte de los participantes en la programación conectada se muestran en la tabla 3.

Tabla 3. Bloques usados en el taller de programación

Bloque	Número de ocasiones	Porcentaje de todos los bloques usados
cambiar y en _	20	12.90 %
cambiar x en _	15	9.68 %
decir_ por _ seg.	15	9.68 %
si _ entonces	15	9.68 %
mover _ pasos	13	8.39 %
por siempre	12	7.74 %
ir a x_ y_	10	6.45 %
rebotar si toca un borde	10	6.45 %
Variable	10	6.45 %
al presionar	7	4.52 %
preguntar _ y esperar	5	3.23 %
Respuesta	5	3.23 %
repetir _	5	3.23 %
operadores	5	3.23 %
fijar estilo de rotación	4	2.58 %
al presionar la tecla	4	2.58 %

Fuente: Elaboración propia

Con esta información se puede obtener la frecuencia de ocasiones que son usadas las categorías de instrucciones (tabla 4). La información demuestra que los alumnos centran más su interés en el movimiento de los objetos y aplican de manera esporádica la construcción de expresiones aritméticas.

Tabla 4. Concentrado de uso de categorías

Categoría	Frecuencia	Porcentaje
Movimiento	72	46.45 %
Control	32	20.64 %
Apariencia	15	9.67 %
Eventos	11	7.09 %
Sensores	10	6.45 %
Variables	10	6.45 %
Operadores	5	3.22 %

Fuente: Elaboración propia

Por otra parte, para identificar aún más el comportamiento de cada uno de los participantes se grabaron algunas actividades según estas tres categorías: a) conceptos de

programación, b) operatividad y c) niveles de comprensión (Funke, Geldreich y Hubwieser, 2017).

- a. Conceptos de programación. Todos los estudiantes participaron en el desarrollo de las actividades con diferentes grados de complejidad. Algunos de ellos realizaron actividades diferentes a las esperadas por el instructor, es decir, existía una autonomía y curiosidad por involucrar bloques, principalmente de movimiento. Con respecto al uso de las variables, la mayoría cumplió con el proceso indicado de declarar, inicializar y mostrar el valor de las variables. Así pues, 17 % de los participantes implementó algún concepto visto en el taller en sus actividades finales.
- b. Operatividad. 85 % de los estudiantes cumplieron con actividades donde el usuario debía ingresar datos por medio del teclado. La mayoría identificó el proceso para que su programa permitiera ingresar datos y mostrara un resultado.
- c. Nivel de comprensión. Para evaluar este nivel se empleó la clasificación de la taxonomía de SOLO. En la tabla 5 se identifican los porcentajes de los niveles de aprendizaje de los participantes.

Tabla 5. Concentrado de niveles de aprendizaje SOLO

Nivel	Porcentaje
N1: Preestructural	25 %
N2: Uniestructural	35 %
N3: Multiestructural	21 %
N4: Relacional	8 %
N5: Abstracto	11 %

Fuente: Elaboración propia

En la clasificación se observa que 25 % de los participantes deben integrar más elementos para que entiendan el significado de los bloques y desarrollar un PC de manera que les permita desarrollar mayor capacidades.

En cuanto a los siguientes niveles, 35 % de los alumnos incorporan secuencias de instrucciones simples; esto significa que se debe realizar mayor trabajo en estructuras de control de repetición y decisión.

Sobre los dos últimos niveles, se identifica que los alumnos adquieren habilidades en el PC, como la abstracción y el pensamiento algorítmico.

Discusión

Este primer acercamiento al desarrollo de las habilidades del PC es alentador, aunque falta por detallar algunas actividades que permitan incrementar el nivel de aprendizaje en el taller de programación. Además, debido a que los alumnos prefieren el uso de instrucciones de movimiento, se debe resaltar más la parte de abstracción. Asimismo, a partir de este conocimiento que adquieren los alumnos de programación en bloques, se pueden incorporar elementos de trabajo en el paradigma de construcción de código textual. Es decir, nuevas formas de aplicar las estructuras de control desde perspectivas conocidas para luego efectuar una migración natural a otros entornos de programación

Igualmente, se observa una clara participación y agrado por parte de los niños en este tipo de aprendizajes. Este dato es muy importante y se puede aprovechar para fomentar una cultura de creación tecnológica, y no solo de consumidor de ella.

Sin embargo, para conseguir tal propósito se deben establecer estrategias a nivel global, como las expuestas por la Secretaría de Educación Pública del Gobierno de México, a través de la Coordinación General @prende.mx, y su Programa de Inclusión Digital (PID). De esa forma se puede robustecer el marco referencial de PC para la educación básica, cuyo objetivo principal es cultivar en la comunidad educativa mexicana los saberes asociados al mundo digital.

Conclusión

Como conclusión se puede indicar que los estudiantes pueden participar de manera activa en la implementación de proyectos de pensamiento computacional mediante diversas estrategias pedagógicas en la llamada computación conectada y desconectada. De hecho, se puede asegurar que el uso de ambas variantes aporta diversos aprendizajes a través de una dinámica que se diferencia de los talleres tradicionales.

Además, el desarrollo del PC no se relaciona solamente con programar en un entorno como Scratch, sino que involucra una serie de mecanismos del pensamiento, como el abstracto-matemático y el pragmático-ingenieril, los cuales son aplicados a lo largo de la vida.

Asimismo, se comprueba que el pensamiento computacional se puede desarrollar usando un lápiz y un papel, con didácticas dinámicas como elemento alternativo para la transferencia de las competencias de este tipo de pensamiento. Además, el concepto de programación desconectada se convierte en elemento clave cuando se tienen carencias en equipos de cómputo.

En definitiva, los resultados muestran una poca o nula resistencia en la adquisición de los conocimientos experimentados en el taller. Por tanto, establecen mecanismos de confianza para la resolución de problemas basados en computadora o bien en la interpretación y uso de estructuras de control.

Futuras líneas de investigación

A pesar de que este fue un primer acercamiento, se percibe un avance considerable en el desarrollo de materiales para el seguimiento de impartición de talleres. Sin embargo, estos deben ser adaptados a una nueva realidad que combina lo virtual y lo presencial.

Asimismo, se pueden implementar otras estrategias diferentes a las usadas en este trabajo para incrementar las habilidades del pensamiento computacional y para lograr un mayor porcentaje en los niveles de lo relacional y lo abstracto.

Por otro lado, es necesario desarrollar líneas de acción para que todos los alumnos de educación básica tengan acceso a materias que incluyan la programación de computadoras.

Referencias

- Bell, T., Alexander, J., Freeman, I. and Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology*, 13(1), 20-29.
- Biggs, J. B. and Collis, K. F. (1982). *Evaluating the quality of learning: the SOLO taxonomy (structure of the observed learning outcome)*. Academic Press.
- Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A. and Barone, D. (2017). Development of computational thinking skills through unplugged activities in primary school. *Proceedings of the 12th workshop on primary and secondary computing education - WiPSCE '17* (pp. 65-72). ACM Press.
- Dagiene, V., Pelikis, E. and Stupurienė, G. (2015). Introducing Computational Thinking through a Contest on Informatics: Problem-solving and Gender Issues. *Informacijos Mokslai/Information Sciences*, 73.
- Del Olmo, J., Cózar-Gutierrez, R. and González-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of Primary Education. *Computers & Education*, 150.
- Espino, E. E. y González, C. (2015). Estudio sobre diferencias de género en las competencias y las estrategias educativas para el desarrollo del pensamiento computacional. *Revista de Educación a Distancia*, 46.
- Fagerlund, J., Häkkinen, P., Vesisenaho, M. and Viiri, J. (2021). Computational thinking in programming with scratch in primary schools: A systematic review. *Computer Applications in Engineering Education*, 29(1), 12-28.
- Funke, A., Geldreich, K. and Hubwieser, P. (2017). Analysis of scratch projects of an introductory programming course for primary school students. *2017 IEEE Global Engineering Education Conference (EDUCON)* (pp. 1229-1236). Athens, Greece: IEEE.
- Gantz, J. (2019). *The salesforce economic impact: 4.2 million new jobs, \$1.2 trillion of new business revenues from 2019 to 2024*. Salesforce.
- González-González, C. (2019). State of the art in the teaching of computational thinking and programming in childhood education. *Education in the Knowledge Society*, (20), 1-15.

- Hsu, T. C., Chang, S. C. and Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, 126(1), 296-310.
- Huang, W. and Looi, C. (2021). A critical review of literature on "unplugged" pedagogies in K-12 computer science and computational thinking education. *Computer Science Education*, 31(1), 83-111.
- Hufad, A., Faturrohman, M. and Rusdiyani, I. (2021). Unplugged Coding Activies for Early Childhood Problem-Solving Skills. *Journal Pendidikan Usia Dini*, 15(1), 121-140.
- Imler, B. and Eichelberger, M. (2011). Using screen capture to study user research behavior. *Library Hi Tech*, 29(3), 446-454.
- Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L. and Settle, A. (2014). Computational thinking in K-9 education. In *Proceedings of the working group reports of the 2014 on innovation & technology in computer science education conference*, (pp. 1-29).
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Brighton, Sussex: Basic Books.
- Papert, S. (1987). *Desafío de la mente*. Buenos Aires, Argentina: Ediciones Galapago.
- Threekunprapa, A. and Yasri, P. (2021). The role of augmented reality-based unplugged computer programming in the effectiveness of computational thinking. *International Journal of Mobile Learning and Organisation*, 15(3), 233-250.
- Torres-Torres, Y. D., Román-González, M. and Pérez-González, J. (2019). Implementation of Unplugged Teaching Activities to Foster Computational Thinking Skills in Primary School from a Gender Perspective. In *Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality* (pp. 209-215). ACM.
- Universidad de Canterbury (25 de febrero de 2020). *Informática sin computadora*. Retrieved from <https://csunplugged.org/es/>
- Venkatesh, P., Das, S. and Das, A. (2021). Design and Development of Low-Cost Unplugged Activies for Teaching Computational Thinking at k-5 Level. *Design for Tomorrow*, 3, 523-534.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.

Rol de Contribución	Autor (es)
Conceptualización	Carlos Enríquez Ramírez (principal) Mariza Raluy Herrero (igual) Luz Maríz Vega Sosa (igual)
Metodología	Carlos Enríquez Ramírez (principal) Mariza Raluy Herrero (igual) Luz Maríz Vega Sosa (igual)
Software	No aplica
Validación	Carlos Enríquez Ramírez (principal) Mariza Raluy Herrero (igual) Luz Maríz Vega Sosa (igual)
Análisis Formal	Carlos Enríquez Ramírez (principal) Mariza Raluy Herrero (igual) Luz Maríz Vega Sosa (igual)
Investigación	Carlos Enríquez Ramírez (principal) Mariza Raluy Herrero (igual) Luz Maríz Vega Sosa (igual).
Recursos	Carlos Enríquez Ramírez (principal) Mariza Raluy Herrero (igual) Luz Maríz Vega Sosa (igual)
Curación de datos	Carlos Enríquez Ramírez (principal) Mariza Raluy Herrero (igual) Luz Maríz Vega Sosa (igual)
Escritura - Preparación del borrador original	Carlos Enríquez Ramírez (principal) Mariza Raluy Herrero (igual) Luz Maríz Vega Sosa (igual)
Escritura - Revisión y edición	Carlos Enríquez Ramírez (principal) Mariza Raluy Herrero (igual) Luz Maríz Vega Sosa (igual)
Visualización	Carlos Enríquez Ramírez (principal) Mariza Raluy Herrero (igual) Luz Maríz Vega Sosa (igual)
Supervisión	Carlos Enríquez Ramírez (principal) Mariza Raluy Herrero (igual) Luz Maríz Vega Sosa (igual)
Administración de Proyectos	Carlos Enríquez Ramírez (principal) Mariza Raluy Herrero (igual) Luz Maríz Vega Sosa (igual)
Adquisición de fondos	No aplica