

Robótica Educativa para el desarrollo del Pensamiento Computacional, como una extensión metodológica de ChildProgramming



**Isabel Cristina Mejía Córdoba
Byron Giovanni Salazar España**

Director: PhD. Julio Ariel Hurtado Alegría

Universidad del Cauca

**Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Sistemas
Grupo de Investigación y Desarrollo en Ingeniería de Software (IDIS)
Línea de Investigación en Ingeniería de Software
Popayán, septiembre de 2021**

Robótica Educativa para el desarrollo del Pensamiento Computacional, como una extensión metodológica de ChildProgramming



Monografía de Trabajo de Grado para optar al título de Ingeniero en
Electrónica y Telecomunicaciones

Isabel Cristina Mejía Córdoba
Byron Giovanni Salazar España

Director: PhD. Julio Ariel Hurtado Alegría

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Sistemas
Grupo de Investigación y Desarrollo en Ingeniería de Software (IDIS)
Línea de Investigación en Ingeniería de Software
Popayán, septiembre de 2021

NOTA DE ACEPTACIÓN

[1]

PRESIDENTE DEL JURADO

JURADO

Tabla de contenido

1.	INTRODUCCIÓN	1
1.1	PLANTEAMIENTO DEL PROBLEMA.....	1
1.2	OBJETIVOS	3
1.2.1	Objetivo General	3
1.2.2	Objetivos Específicos	3
1.3	RESULTADOS OBTENIDOS.....	3
1.4	ORGANIZACIÓN DEL DOCUMENTO	3
2	MARCO TEÓRICO Y REVISIÓN DE LA LITERATURA	4
2.1	MARCO TEÓRICO	4
2.1.1	Pensamiento Computacional (CT)	4
2.1.2	Robótica Educativa (ER).....	6
2.1.3	ChildProgramming	7
2.2	REVISIÓN DE LA LITERATURA.....	7
2.2.1	Selección de los documentos	7
2.2.2	Distribución de los documentos seleccionados	9
2.2.3	La Robótica Educativa y el Pensamiento Computacional	10
2.2.3.1	Confluencia entre la Robótica Educativa y el Pensamiento computacional	10
2.2.3.2	Aportes de la Robótica Educativa al Pensamiento Computacional.....	10
2.2.4	Eficacia de la Robótica Educativa en el desarrollo del Pensamiento Computacional	12
2.3	RECOMENDACIONES DE LOS INVESTIGADORES	15
3	CHILDPROGRAMMING-RE: UNA EXTENSIÓN DE CHILDPROGRAMMING PARA ROBÓTICA EDUCATIVA.....	17
3.1	LA EXTENSIÓN CHILDPROGRAMMING-RE	18
3.2	CONCEPTOS BÁSICOS DEL PENSAMIENTO COMPUTACIONAL Y ROBÓTICA	19
3.2.1	Conceptos del Pensamiento Computacional	20
3.2.2	Conceptos de Robótica [59]	21
3.3	PRÁCTICAS DE CHILDPROGRAMMING-RE	21
3.3.1	Prácticas educativas	23
3.3.2	Prácticas para el Desarrollo del Pensamiento Computacional en el Contexto de la Robótica Educativa	25
3.3.3	Prácticas de Evaluación	29
3.4	GUÍA DE CHILDPROGRAMMING CON LA EXTENSIÓN PENSAMIENTO COMPUTACIONAL Y ROBÓTICA EDUCATIVA.	31

3.5 Lecciones aprendidas	37
4. PLANEACIÓN, EJECUCIÓN Y RESULTADOS DEL ESTUDIO	39
4.1 PLANEACIÓN DEL ESTUDIO.....	39
4.1.1 Metodología de Estudio	39
4.1.1.1 Exploración.....	39
4.1.1.2 Construcción de la Extensión.....	40
4.1.1.3 Evaluación	40
4.1.2 Variables.....	40
4.2 DISEÑO DEL ESTUDIO	40
4.2.1 Selección de los participantes del estudio.....	40
4.2.2 Contexto del Estudio	41
4.2.3 Pregunta de investigación	42
4.2.4 Objetivo del estudio:	42
4.3 EJECUCIÓN DEL ESTUDIO.....	42
4.3.1 Estudio Exploratorio	42
4.3.1.1 Selección del estudio:.....	42
4.3.1.2 Los sujetos de investigación	42
4.3.1.3 Ejecución del estudio exploratorio	43
4.3.1.4 Instrumentos de medición del Estudio Exploratorio	48
4.3.1.5 Resultados cuantitativos.	50
4.3.1.6 Análisis de resultados	57
4.3.2 Ejecución del experimento confirmatorio	58
4.3.2.1 Contexto de la investigación.....	58
4.3.2.2 Pregunta de investigación	58
4.3.2.3 Objetivo del estudio:	59
4.3.2.4 Selección del estudio	59
4.3.2.5 El experimento confirmatorio y los sujetos de investigación	59
4.3.2.6 Ejecución del experimento confirmatorio	59
4.3.2.7 Instrumentos y técnicas para recolección de información	68
4.3.2.8 Resultados cuantitativos.	70
4.4 ANÁLISIS	78
5. CONCLUSIONES Y TRABAJO FUTURO	80
5.1 CONCLUSIONES	80

5.2 TRABAJO FUTURO	81
Bibliografía	82

Índice de figuras

Figura 1. Proceso de selección de las publicaciones, de acuerdo con la aplicación sucesiva de los criterios de exclusión.	9
Figura 2. Distribución a) por cantidad y b) por porcentaje de los documentos seleccionados para revisión.....	9
Figura 3. Relación entre las habilidades del PC y el número de veces estudiada en los estudios prácticos.....	12
Figura 4. Tipos de robot empleados en los estudios.	13
Figura 5. Existencia de grupos de control entre los estudios.	13
Figura 6. Participantes programando en mBlock durante el estudio exploratorio.....	17
Figura 7. El mBot Ranger, de Makeblock.....	18
Figura 8. Extensión de Childprogramming y sus paquetes metodológicos (Fuente Propia).	18
Figura 9. Conceptos básicos y sus subpaquetes (fuente propia).	20
Figura 10. Clasificación de las prácticas para Childprogramming con RE. Fuente Propia.	22
Figura 11. Ciclo de vida de ChildProgramming. Tomado de [8].....	31
Figura 12. Requisitos mínimos de una sesión de entrenamiento en robótica educativa. Adaptado de: RobotLab (http://www.bnm.me.gov.ar/).....	32
Figura 13. Estructura por etapas de una sesión de robótica.	33
Figura 14. Participantes del estudio exploratorio.	41
Figura 15. Distribución de las respuestas a la pregunta N.1 a) en el Pre-test y b) en el Pos-test.	51
Figura 16. Distribución de las respuestas a la pregunta N.7 a) en el Pre-test y b) en el Pos-test.	53
Figura 17. Comparativo a) Pre-test y b) Pos-test sobre la apropiación de los conceptos de PC.	55
Figura 18. Resultados del análisis con Dr. Scratch del código de a) la segunda sesión, b) la tercera sesión y c) la cuarta sesión de entrenamiento.	56
Figura 19. Algunos participantes del grupo de control.....	60
Figura 20. A la derecha equipo “Walter”, a la izquierda equipo “Walter 2.0”.....	62
Figura 21. Distribución de las respuestas a la pregunta N.1 a) en el Pre-test y b) en el Pos-test. Comparativo entre el grupo de control y el grupo experimental.	71
Figura 22. Distribución de las respuestas a la pregunta N.7 a) en el Pre-test y b) en el Pos-test Grupo de Control y Experimental.....	73
Figura 23. Comparativo Pos-test sobre la apropiación de los conceptos de PC en a) el grupo de control y b) el grupo experimental.....	74
Figura 24. Resultados del análisis con Dr. Scratch del código de a) la primera sesión, b) la segunda sesión, c) la tercera sesión, d) cuarta sesión y e) quinta sesión de entrenamiento del grupo de control.....	76
Figura 25. Resultados del análisis con Dr. Scratch del código de a) la segunda sesión, b) la tercera sesión, c) cuarta sesión y d) quinta sesión de entrenamiento del grupo experimental.	78

Índice de tablas

Tabla 1. Modelo para el desarrollo de habilidades de CT propuesto por Atmatzidou y Demetriadis, condensado de [56] y [22].	15
Tabla 2. Estructura para la descripción de prácticas. Fuente propia	23
Tabla 3. Organización de Actividades.	24
Tabla 4. Organización del plan de trabajo.	25
Tabla 5. Práctica Fundamental de Incrementalidad.	26
Tabla 6. Práctica Aprendizaje Basado en el Error.	27
Tabla 7. Práctica del desarrollo del pensamiento computacional a través de la solución de problemas de robótica educativa.	28
Tabla 8. Pre-test	29
Tabla 9. Pos-test	30
Tabla 10. Observación.	31
Tabla 11. Actividades para el desarrollo del Pensamiento Computacional en el contexto de la Robótica Educativa.	37
Tabla 12. Curso “Scratch + Arduino: Introducción a la Robótica”	43
Tabla 13. Sesión 1 de entrenamiento	44
Tabla 14. Sesión 2 de entrenamiento	46
Tabla 15. Sesión 3 de entrenamiento	47
Tabla 16. Sesión 4 de entrenamiento	48
Tabla 17. Distribución de las preguntas según las variables a medir en el Pre-test. Fuente Propia	49
Tabla 18. Distribución de las preguntas según las variables a medir en el Pos-test. Fuente Propia	49
Tabla 19. Nivel de Competencia para cada aspecto del pensamiento computacional [65]	50
Tabla 20. Puntuación de los criterios relacionados con la motivación en el Protocolo de Observación	52
Tabla 21. Puntuación de los criterios relacionados con la participación en el Protocolo de observación.	53
Tabla 22. Modelo de conceptos de CT utilizados en este proyecto.	54
Tabla 23. Habilidades de Pensamiento Computacional, según son descritas vistas por la herramienta Dr. Scratch y su aplicación al estudio exploratorio.	57
Tabla 24. Curso “Programación con Scratch para Niños”	61
Tabla 25. Sesión 1 de entrenamiento	61
Tabla 26. Sesión 2 de entrenamiento	61
Tabla 27. Sesión 3 de entrenamiento	61
Tabla 28. Sesión 4 de entrenamiento	62
Tabla 29. Sesión 5 de entrenamiento	62
Tabla 30. Curso “Scratch + Arduino: Introducción a la Robótica”	63
Tabla 31. Sesión 1 de entrenamiento	64
Tabla 32. Sesión 2 de entrenamiento	65
Tabla 33. Sesión 3 de entrenamiento	66
Tabla 34. Sesión 4 de entrenamiento	67
Tabla 35. Sesión 5 de entrenamiento	68
Tabla 36. Distribución de las preguntas según las variables a medir en el pre-test Grupo de control. Fuente Propia	69

Tabla 37. Distribución de las preguntas según las variables a medir en el pos-test Grupo de control. Fuente Propia	69
Tabla 38. Distribución de las preguntas según las variables a medir en el Pos-test – Grupo Experimental. Fuente Propia	70
Tabla 39. Puntuación de los criterios relacionados con la motivación en el Protocolo de Observación para el grupo experimental.	72
Tabla 40. Puntuación de los criterios relacionados con la participación en el Protocolo de observación.....	74

INTRODUCCIÓN

1.1 PLANTEAMIENTO DEL PROBLEMA

El pensamiento computacional (CT¹) es una habilidad del siglo XXI que ha venido ganando importancia rápidamente en el mundo cada vez más tecnológico en el que vivimos y que se ha posicionado como un objetivo de investigación de todas las disciplinas de ciencia e ingeniería [1]. De acuerdo con Wing [1], [2], “CT es el proceso de pensamiento involucrado en la formulación de un problema y la expresión de su(s) solución(es), de tal manera que una computadora - humana o máquina - pueda ejecutarla efectivamente”. Existe el creciente reconocimiento de que las prácticas de CT son fundamentales para que todos los estudiantes aprendan y que, por tanto, deben ser desarrolladas por todos. Ellas forman la piedra angular del lenguaje de la innovación e impulsarán todos los descubrimientos futuros de Ciencia, Tecnología, Ingeniería y Matemáticas (STEM²) [3]. Esa omnipresencia derivada de los conceptos de pensamiento computacional, dicta la importancia de exponer a los estudiantes a tales nociones desde sus primeros años escolares y de ayudarlos a tomar conciencia sobre cuándo y cómo aplicar esta habilidad esencial [4]. Sin embargo, las actividades de CT pueden no resultar muy atractivas para los jóvenes [5] y prácticas tales como “considerar problemas analíticamente” y “usar datos para tomar decisiones” lucen abstractas y difíciles de comprender por parte de los estudiantes [3]. Uno de los desafíos de los cursos de CT es motivar a aquellos estudiantes que se sienten de antemano desanimados, ya que perciben la programación como una tarea complicada; por tanto, esos cursos deberían ser capaces de cautivar la más amplia audiencia posible, llegando incluso a aquellos estudiantes que no se consideran a sí mismos como candidatos para disciplinas STEM.

La construcción de software requiere que se desarrollen habilidades para poder concebir problemas y abstraer estos de manera tal que puedan ser resueltos satisfactoriamente. Este esquema implicaría que desde muy pequeños los desarrolladores de software comiencen a utilizar estrategias adecuadas para formar esas habilidades [6]. Precisamente, ChildProgramming, propuesto por el grupo IDIS³ de la Universidad del Cauca, formaliza un modelo de procesos para soportar el desarrollo del pensamiento computacional a través de la enseñanza del desarrollo de software y la solución de problemas complejos a temprana edad [7] y es el resultado de un proceso de abstracción de conceptos, prácticas y actividades evidenciadas en niños con edades comprendidas entre los 8 y los 10 años [8]. Dentro de las debilidades del desarrollo del CT, está la falta de estrategias que permitan incrementar la participación y la motivación de los estudiantes, destacándose la carencia de material educativo que facilite la comprensión de los conceptos de CT [7], [9]. De hecho, en [10] se señalan, como oportunidades potenciales de investigación, la necesidad de capturar el interés de los estudiantes en el CT y de diseñar cursos de aprendizaje sobre conceptos y habilidades fundamentales de CT. Aunque existen propuestas para subsanar estas deficiencias, algunas de las cuales se exponen en el apartado de “Trabajos relacionados”, aquellas aún muestran limitaciones para desarrollar y/o evaluar el desarrollo de las habilidades computacionales de los niños. ChildProgramming utiliza algunas estrategias orientadas en este camino, pero se limita a la codificación como foco central del

¹ CT: Computational Thinking.

² STEM: Science, Technology, Engineering and Mathematics.

³ IDIS: Investigación y Desarrollo en Ingeniería de Software. <http://www.unicauca.edu.co/idis/>

desarrollo del CT. Sin embargo, los conceptos de CT no se encuentran ligados exclusivamente al código y pueden, por tanto, impulsarse mediante otras herramientas y actividades [11], [12]. Tabesh [13], por ejemplo, sugiere llevar el pensamiento computacional al salón de clases mediante laboratorios de matemática computacional y de inteligencia artificial, incluyendo en estos últimos al Internet de las cosas (IoT⁴) y a la robótica, mientras que Hacker [10] afirma que el CT puede enseñarse empleando sistemas robóticos y de diseño de juegos, que sean familiares para los profesores de tecnología e ingeniería y con los que estos se sientan a gusto. En este sentido, los robots educativos ofrecen una forma concreta de enganchar el pensamiento computacional y el trabajo en equipo. Aunque no existe una definición estandarizada de lo que es la Robótica Educativa (ER⁵), sí existe consenso en que es un espacio de conocimiento interdisciplinar que requiere la construcción de un objeto tecnológico con un fin concreto y que desarrolla habilidades claves para el alumnado del siglo XXI [14]. Existe una fuerte correlación entre los principios ER y las ideas del CT [15], y la combinación del desarrollo de software y hardware da lugar a un trabajo multidisciplinar, que requiere de nuevas formas de organización de los equipos de trabajo.

Se ha postulado que las actividades de robótica programable en la educación STEM tienen un impacto positivo en el desarrollo del CT [16], [17]. La robótica educativa puede facilitar el aprendizaje activo, promover el razonamiento y el pensamiento crítico, y también mejorar el interés y motivación del estudiante para abordar temas a menudo complejos o abstractos [18]. El hecho de que éste pueda manipular y experimentar con estas herramientas robóticas en un ambiente de aprendizaje hace que pueda trabajar con problemas del mundo real y centrar sus percepciones y observaciones en la actividad que está realizando [14]. Particularmente, la aparición y el relativamente bajo costo de material educativo, en forma de robots comerciales DIY (Do It Yourself) ha permitido, a personas de todas las edades, construir diferentes prototipos robóticos sin necesidad de tener conocimientos avanzados en mecánica, electrónica o programación [19] y ha impulsado el uso de ER sobre software (como Scratch⁶ y Coffee) y hardware libres para promover las habilidades de CT [11], [20].

Además de estimular y mantener la atención y el interés del estudiante, las actividades que tienen resultados tangibles, como la ER, proporcionan oportunidades para enseñar CT [11], ayudando a la transición de aprendices pasivos a activos, desarrollando habilidades mentales y creando nuevo conocimiento [17]. Aun así, es obvio que la robótica por sí sola no posee la capacidad de garantizar el aprendizaje, lo que hace necesario impulsar la investigación sobre cómo trabajar con ER para desarrollar habilidades específicas en los estudiantes, tales como la solución de problemas y el pensamiento computacional en general [21], [22]. Considerando entonces el CT, la ER y ChildProgramming desde una perspectiva integral, surge la pregunta de investigación: ¿cómo incentivar la participación y motivación hacia el desarrollo del Pensamiento Computacional (CT) en niños entre 10 y 12 años, a partir de los conceptos de STEM relacionados con la Robótica Educativa, dentro del contexto del trabajo de ChildProgramming?

⁴ IoT: Internet of Things.

⁵ ER: Educational Robotics.

⁶ Scratch: <https://scratch.mit.edu/>

1.2 OBJETIVOS

1.2.1 Objetivo General

Establecer el efecto que tiene la ER en la motivación y participación hacia el desarrollo del CT en niños entre los 10 y los 12 años, dentro del contexto de ChildProgramming.

1.2.2 Objetivos Específicos

- Determinar de acuerdo a los estudios previos, qué aspectos de la Robótica Educativa contribuyen al desarrollo del pensamiento computacional.
- Proponer una extensión metodológica para ChildProgramming que permita integrar aspectos de la ER en el desarrollo del CT.
- Evaluar la asimilación de los conceptos del CT (abstracción, descomposición, reconocimiento de patrones y diseño de algoritmos) por parte de niños entre los 10 y los 12 años, a través de un experimento controlado en el contexto de la extensión para ChildProgramming propuesta.

1.3 RESULTADOS OBTENIDOS

Monografía de grado: este documento describe el proceso para el desarrollo del proyecto, herramientas seleccionadas, estudios realizados y aportes, conclusiones y recomendaciones para el desarrollo de futuras investigaciones

Anexos: este documento contiene aportes necesarios y complementarios del proyecto, no incluidos en la monografía.

Artículo: el artículo “Robótica Educativa como herramienta para el desarrollo del Pensamiento Computacional. Una revisión de la literatura” se hizo con base en el Capítulo 2 de esta monografía.

1.4 ORGANIZACIÓN DEL DOCUMENTO

El presente documento está organizado como sigue a continuación:

Capítulo 1. Introducción.

Capítulo 2. Revisión de la literatura.

Capítulo 3. ChildProgramming-RE: una extensión de ChildProgramming para Robótica Educativa.

Capítulo 4. Planeación, ejecución y resultados del estudio

Capítulo 5. Conclusiones y trabajo futuro.

2 MARCO TEÓRICO Y REVISIÓN DE LA LITERATURA

2.1 MARCO TEÓRICO

2.1.1 Pensamiento Computacional (CT)

Las raíces del término Pensamiento Computacional pueden rastrearse hasta el trabajo de Seymour Papert en las décadas de 1980 y 1990 con el lenguaje de programación LOGO y el desarrollo del pensamiento procedimental en niños a través de la programación [17], [23]. Luego, ya en 2006, Jeannette Wing reutilizó y popularizó el CT para referirse a una habilidad fundamental para cualquier persona y que, junto con la lectura, la escritura y la aritmética, debería hacer parte de la capacidad analítica de cada niño [24]. En este acercamiento inicial, ella sostiene que el CT involucra la solución de problemas, el diseño de sistemas y la comprensión del comportamiento humano con base en los conceptos fundamentales de la informática, y comprende - pero no se limita a - la reformulación de problemas, el pensamiento recursivo, el procesamiento paralelo, la interpretación de código como información y viceversa, la abstracción, la descomposición y la modularización, y el razonamiento heurístico. A partir de esta visión originaria, han surgido diversos intentos por obtener una definición concreta y certera de lo que es el Pensamiento Computacional, sin que hasta el momento se haya llegado a un consenso al respecto [15], [25], [26]. De acuerdo con Selby y Woollard [25], para algunos autores lo realmente importante es cómo se enseña el CT y cómo éste se evidencia en los aprendices, para lo cual basta con una definición amplia de CT; en contraparte, para otros, es necesaria una definición robusta y común, sin la cual se hace difícil desarrollar herramientas evaluativas que midan la capacidad de pensar computacionalmente, sin olvidar que, como sostienen Haseski et al. [27], “la definición de conceptos, fenómenos o eventos es crucial para su comprensión y en la determinación de los límites del tema”.

Es así como, tras hacer una revisión metódica de la literatura existente, Selby y Woollard [25] encontraron que, en cuanto a CT se refiere, existen tres términos que aparecen consistentemente: Proceso de pensamiento, abstracción y descomposición. Sobre ellos, pareciera haber un acuerdo para que hagan parte de la definición. Para Selby y Woollard, el Pensamiento Computacional es una actividad enfocada en, mas no restringida a, la solución de problemas, que incorpora procesos de pensamiento que involucran abstracción, descomposición, diseño algorítmico, evaluación (de tiempo y espacio, de potencia y almacenamiento) y generalización.

Con base en el lenguaje de programación Scratch y la comunidad surgida a su alrededor, Brennan y Resnick [28] desarrollaron su propia definición de CT que involucra tres dimensiones:

- Conceptos computacionales (aquellos que los diseñadores emplean mientras programan): En Scratch son bloques de programación y son comunes a muchos lenguajes de programación (secuencias, bucles, paralelismo, eventos, condicionales, operadores y datos).
- Prácticas computacionales (aquellas que los diseñadores desarrollan mientras programan): Son prácticas enfocadas en el pensamiento y aprendizaje (ser incremental e iterativo, prueba y depuración, reutilización y combinación, y abstracción y modularización).
- Perspectivas computacionales (aquellas que los diseñadores se forman sobre el mundo que los rodea y sobre ellos mismos): Son formas de ver la computación como un medio

de expresión de ideas, de conexión (crear con otros y para otros) y de cuestionamiento sobre y con tecnología.

La misma Wing refinaría más adelante su concepción del CT para visualizarlo como un marco para el pensamiento y, más precisamente, como “el proceso de pensamiento involucrado en la formulación de un problema y la expresión de su(s) solución(es), de tal manera que una computadora -humana o máquina- pueda ejecutarla efectivamente” [1], destacando la abstracción como su esencia y resaltando, al igual que otros investigadores, que no se trata solamente de solucionar un problema, sino también de formularlo [2], [29], [30], [31].

La Sociedad Internacional para la Tecnología en la Educación (ISTE⁷) y la Sociedad de Maestros de Ciencias de la Computación (CSTA⁸) elaboraron conjuntamente una definición operacional que provee un marco y un vocabulario para el CT [29]. Es esta misma definición la que aparece también en Google for Education [31], según la cual, el CT es un proceso de solución de problemas que incluye un conjunto de características y actitudes. Entre las habilidades involucradas en el CT se cuentan: Formular problemas, de forma que permita emplear una computadora u otra herramienta para ayudar a resolverlos; organizar y analizar datos de manera lógica; representar datos mediante abstracciones, tales como modelos y simulaciones; automatizar soluciones a través de pensamiento algorítmico (una serie de pasos ordenados); identificar, analizar e implementar posibles soluciones con el fin de alcanzar la más eficiente y efectiva combinación de pasos y recursos; generalizar y transferir este proceso de solución a una amplia variedad de problemas.

Las anteriores destrezas se soportan y mejoran por actitudes como: confianza para enfrentar la complejidad; persistencia al trabajar con problemas difíciles; tolerancia a la ambigüedad; habilidad para lidiar con problemas de final abierto, y habilidad para comunicarse y trabajar con otros para alcanzar una solución o meta común.

Los conceptos de CT, de acuerdo con Google for Education, son aquellos procesos mentales y los resultados tangibles asociados con la solución de problemas. Son ellos:

- Abstracción: Identificar y extraer información relevante para definir la(s) idea(s) principal(es).
- Diseño de algoritmos: Crear una serie ordenada de instrucciones para resolver problemas similares o para realizar una tarea.
- Automatización: Hacer que computadoras u otras máquinas realicen tareas repetitivas.
- Análisis de datos: Lograr que los datos tengan sentidos al encontrar patrones o indicios.
- Recolección de datos: Recoger información.
- Representación de datos: Representar y organizar información en gráficas, tablas, palabras o imágenes adecuadas.
- Descomposición: Separar los datos, procesos o problemas en partes más pequeñas y manejables.
- Paralelismo: Procesamiento simultáneo de tareas más pequeñas a partir de una más grande para alcanzar una meta común de forma más eficiente.
- Generalización de patrones: Creación de modelos, reglas, principios o teorías de patrones observados para evaluar los resultados previstos.
- Simulación: Desarrollo de un modelo para imitar procesos del mundo real.

⁷ ISTE: International Society for Technology in Education: <https://www.iste.org/>

⁸ CSTA: Computer Science Teachers Association: <https://www.csteachers.org/>

2.1.2 Robótica Educativa (ER)

Los robots educativos pueden ser vistos como un paso más en la evolución de la tecnología educativa. Una gran parte de las aplicaciones de la tecnología robótica en la educación se ha enfocado en apoyar la enseñanza de áreas íntimamente relacionadas con aquella, tales como la construcción y programación de robots [21]. No obstante, la robótica tiene un impacto potencial en el aprendizaje de las áreas STEM [21], en el desarrollo cognitivo y también de habilidades de investigación, pensamiento creativo, toma de decisiones, resolución de problemas, comunicación y habilidades de trabajo en equipo [32], todas ellas muy en sintonía con los lineamientos del Pensamiento Computacional. De acuerdo con lo anterior, la Asociación de Robótica de Japón, la Comisión Económica de las Naciones Unidas y la Federación Internacional de Robótica han evidenciado un reciente y considerable crecimiento en el uso de robots con fines educativos, y han anticipado que esta tendencia continuará durante los años venideros [33], [21]. Los muchos y diferentes kits robóticos surgidos a partir de la década de 2000 (LEGO Mindstorms RCX, NXT, EV3 y WeDo, Arduino, mBlock, KOBi, Makey Makey, Raspberry Pi, PicoBoard, Beagleboard, Crickets, Bee-bot, Cubetto, etc.), con diseños mejorados y cada vez más amigables, han abonado el terreno para la popularización de la robótica entre estudiantes de todas las edades [32]. No se puede discutir que la Robótica Educativa está cada vez más presente en los centros educativos de todo el mundo [21], [32] pero, a pesar de su creciente uso, no existe un concepto común de aquello que representa. Román et al. [14], citando a otros autores, dicen por ejemplo que la ER es una herramienta al servicio del aprendizaje que permite practicar las habilidades del siglo XXI; o también, de forma más elaborada, que es un contexto de aprendizaje que promueve un conjunto de habilidades vinculadas a la creatividad, el diseño, la construcción, la programación y divulgación de creaciones propias, primero mentales y luego físicas, construidas con diferentes materiales y recursos tecnológicos, que pueden ser programadas y controladas desde un computador o dispositivo móvil. Existe acuerdo, eso sí, en que el valor de la ER no está en enseñar robótica a los estudiantes, sino en aprovechar su carácter multidisciplinar para la construcción de un objeto tecnológico que tiene un fin concreto y que desarrolla habilidades claves para el alumnado del siglo XXI [14], habilidades entre las cuales, por supuesto, sobresale el Pensamiento Computacional ya que, cuando se habla de programación y de robótica, se habla inherentemente de CT. Volviendo sobre los pasos de la ER puede llegarse hasta Papert, quien dio origen a los robots educativos cuando creó su Tortuga, diseñada como un dispositivo real que los niños pudieran controlar con programas desarrollados por ellos mismos en LOGO, un lenguaje de computadora creado por el propio Papert [15], [34]. Las teorías principales detrás de la ER son el constructivismo y el construccionismo [34], [32], [35]. Piaget argumentaba que la manipulación de artefactos es clave para que los niños construyan su conocimiento, a lo que Papert agregó la idea de que esa construcción se hace especialmente efectiva en un contexto en el que el aprendiz se involucra conscientemente en la construcción de una entidad pública, ya se trate de un castillo en la playa o de un artefacto tecnológico [32]. Es por eso que, al constituir algo tangible, y siendo que el desarrollo mental comienza a partir de objetos concretos, antes que el desarrollo de un razonamiento abstracto, la ER no solamente es pertinente, sino que además es apropiada [35]. Cuando se involucran con ER, los niños se motivan y participan con entusiasmo en los proyectos, alcanzando objetivos de aprendizaje y/o desarrollando nuevas habilidades [14], [32].

Finalmente, Benitti [21] apunta hacia el uso de la robótica como herramienta para el desarrollo de habilidades de pensamiento, de solución de problemas y de trabajo en equipo, por tratarse de un área en la que los resultados son inexactos, haciendo necesaria asimismo la elaboración de herramientas de evaluación sobre el particular, coincidiendo con el

pensamiento de Atmatzidou y Demetriadis en su ya citado estudio. Otro de los hallazgos resaltado por Bennitti es que, en los diseños experimentales utilizados con mayor frecuencia, los participantes no fueron asignados aleatoriamente, y en el 40% de los casos no se utilizó un grupo de control. Infiere, por lo tanto, que existe una necesidad evidente de llevar a cabo más estudios que involucren un buen diseño experimental y muestras más significativas.

2.1.3 ChildProgramming

ChildProgramming, propuesto por el grupo IDIS de la Universidad del Cauca, formaliza un modelo de procesos para soportar el desarrollo del pensamiento computacional a través de la enseñanza del desarrollo de software y la solución de problemas complejos a temprana edad [7] y es el resultado de un proceso de abstracción de conceptos, prácticas y actividades evidenciadas en niños con edades comprendidas entre los 8 y los 10 años [8], aunque también se propone el reto de llevar ChildProgramming hasta los últimos grados de formación media [9]. Dicho modelo se compone de tres dimensiones: una cognitiva, relacionada con la comprensión, análisis y apropiación de situaciones; una ágil, basada en las metodologías ágiles de desarrollo de software; una colaborativa, que promueve la interacción entre los alumnos en un intento por incrementar la calidad del aprendizaje y de favorecer en ellos la adquisición de conocimientos de los alumnos mediante el desarrollo de software [6], [7].

Ahondando un poco más en este modelo, ChildProgramming, más que como un trabajo único, debe ser visto como un proceso de desarrollo evolutivo y continuo, aún en curso. Así, por ejemplo, aunque el concepto de CT no aparecía de forma explícita en el modelo original de Hurtado et al. [6], sí ha hecho parte de trabajos posteriores como el de Chimunja et al. [7], llamado ChildProgramming-C, y el estudio de Zúñiga et al. [9]. Son precisamente esos y otros trabajos más, a veces en forma de extensiones, los que han venido dando forma y cuerpo al modelo inicial. Entre ellas se cuentan ChildProgramming-C, dirigida a satisfacer la dimensión colaborativa del modelo, y ChildProgramming-G [36], enfocada en la gamificación como forma de incorporar el aspecto lúdico al mismo. Los citados resaltan, como punto en común, la importancia y necesidad de fomentar la motivación en el proceso de aprendizaje, mientras que, al menos los recién mencionados Chimunja et al. y Zúñiga et al., evidencian la carencia de material educativo que facilite la comprensión de conceptos computacionales.

2.2 REVISIÓN DE LA LITERATURA

2.2.1 Selección de los documentos

A continuación, se expone una revisión de la literatura, siguiendo el formato propuesto por Turner et al. [37]. Esta revisión se concentró en artículos que utilizan o proponen utilizar la Robótica Educativa como medio para el desarrollo de habilidades de Pensamiento Computacional, y tuvo como insumos publicaciones que incluyeran los términos “Educational Robotics” y “Computational thinking”, o sus equivalentes en español,

encontrados en las bases SpringerOpen⁹, SpringerLink¹⁰, ResearchGate¹¹, ACM^{12,13}, IEEE Xplore¹⁴, ERIC¹⁵ y ASEE^{16,17}, además de tres publicaciones escogidas bajo el mismo criterio con ayuda del buscador de Google. Siguiendo criterios de exclusión, se descartaron documentos en otros idiomas. La búsqueda se llevó a cabo entre el 19 y el 23 de agosto de 2020, empleando la cadena de búsqueda "(\"educational robotics\" OR \"educational robots\" OR \"robótica educativa\" OR \"robots educativos\") AND (\"computational thinking\" OR \"pensamiento computacional\")", habiéndose obtenido 395 resultados.

Sobre esas 395 publicaciones se aplicaron los criterios de exclusión que se indican a continuación:

- Criterio de Exclusión 1 (CE1): Documentos que no fueron revisados. Los artículos encontrados no pudieron revisarse, sea porque el documento se encontraba en un idioma diferente de inglés o español, estaba incompleto, no se encontraba disponible en el momento de la consulta, la referencia no era válida u otra razón.
- Criterio de Exclusión 2 (E2): Documentos duplicados. Los artículos ya aparecen en alguna de las otras bases consultadas y, por tanto, se trata de documentos duplicados.
- Criterio de Exclusión 3 (CE3): Documentos no relacionados con el objeto de estudio. Aunque las expresiones referentes a Robótica Educativa y Pensamiento Computacional aparecen en el documento, el artículo no se trata de RE o no se trata de PC; es decir, al menos uno de ellos no es parte principal del mismo. En ocasiones, estas expresiones aparecen tan solo en la bibliografía.
- Criterio de Exclusión 4 (CE4): Documentos fuera del alcance de este proyecto. La relación entre RE y PC, es decir, la forma en que la RE influye en el desarrollo de las habilidades de PC, no es el tema central, no se establece o no se explica. Bajo este criterio también se agrupan artículos en los que los conceptos de PC divergen de los considerados para el presente proyecto, en algunos casos porque no se derivan de alguna definición conocida y concreta de él.

La figura 1 ilustra el proceso de selección de las publicaciones estudiadas, partiendo de los resultados de la búsqueda inicial y siguiendo con la posterior aplicación de los criterios de exclusión ya explicados.

⁹ SpringerOpen. <https://www.springeropen.com/>

¹⁰ SpringerLink. <https://link.springer.com/>

¹¹ ResearchGate. <https://www.researchgate.net/>

¹² Association for Computing Machinery

¹³ ACM Digital Library. <https://dl.acm.org/>

¹⁴ IEEE Xplore. <https://ieeexplore.ieee.org/Xplore/home.jsp>

¹⁵ Education Resources Information Center. <https://eric.ed.gov/>

¹⁶ American Society for Engineering Education.

¹⁷ ASEE PEER. <https://peer.asee.org/>

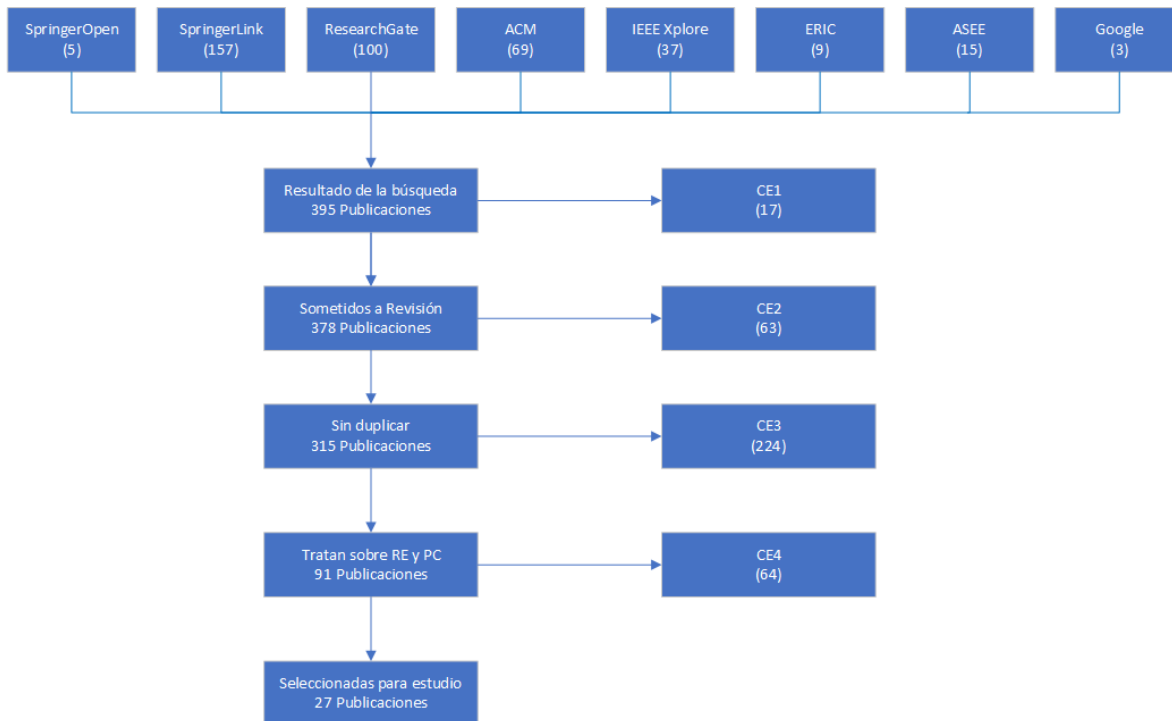


Figura 1. Proceso de selección de las publicaciones, de acuerdo con la aplicación sucesiva de los criterios de exclusión.

Tras ejecutar la búsqueda con la cadena indicada se obtuvieron 395 publicaciones. De ellas, 17 cayeron en el primer criterio de exclusión, quedando 378 publicaciones sometidas a revisión. El segundo criterio de exclusión cubrió a 63 publicaciones, obteniéndose 315 publicaciones únicas. El tercer criterio de exclusión abarcó 224 publicaciones, por lo que 91 tratan sobre RE y PC. De estas, 64 fueron eliminadas por el cuarto criterio de exclusión, por lo que, finalmente, 27 fueron seleccionadas para este estudio. En el anexo A se presenta una lista completa de éstas últimas.

2.2.2 Distribución de los documentos seleccionados

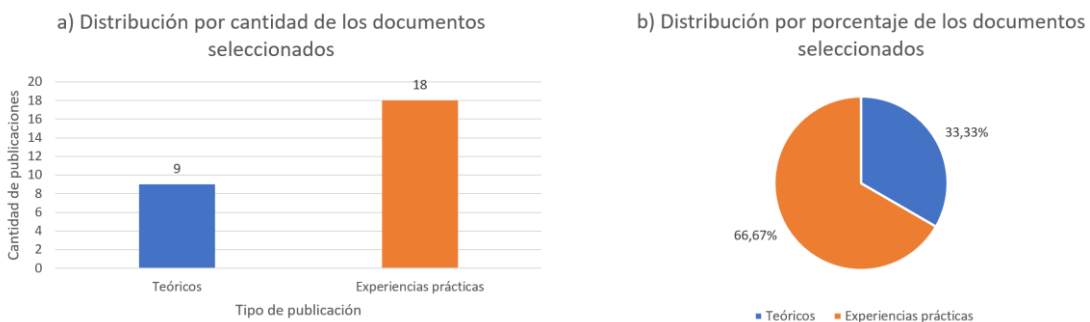


Figura 2. Distribución a) por cantidad y b) por porcentaje de los documentos seleccionados para revisión.

En general, y tras aplicar los criterios de exclusión anteriormente expuestos, las publicaciones consideradas pueden clasificarse en dos grandes grupos: aquellas que tratan asuntos netamente teóricos, como las dedicadas a la revisión de literatura o la presentación de modelos, y aquellas que, solamente o en adición a lo anterior, reportan el resultado de

experiencias prácticas. En el primer grupo se encontraron 9 publicaciones, correspondientes a un 33.33% del total, mientras que, en el segundo, las 18 publicaciones restantes, para un 66.67%. La distribución de los documentos seleccionados se observa en la figura 2.

2.2.3 La Robótica Educativa y el Pensamiento Computacional

Como había sido anticipado por Catlin y Woollard [15], varios autores, como Atmatzidou y Demetriadis [22], Ioannou [38] o Istikomah y Budiyanto [39], señalan un incremento reciente del interés de la comunidad investigativa por la contribución de la robótica en el desarrollo de la solución de problemas y las habilidades de pensamiento computacional, especialmente durante la última década. De hecho, citando a Caballero et al. [40], la comunidad internacional promueve el uso de programación tangible, a través de RE, para alcanzar el desarrollo del PC. El desarrollo de nuevas tecnologías y su adaptación al contexto escolar han permitido que se establezcan iniciativas y proyectos con el propósito de fortalecer nuevas habilidades y competencias en los participantes. Estos proyectos son posibles con el uso de la robótica y de las interfaces de programación gráfica [41], [42].

2.2.3.1 Confluencia entre la Robótica Educativa y el Pensamiento computacional

En general, existe coincidencia entre los investigadores consultados al ubicar al construccionismo, una extensión del constructivismo, como la teoría dominante en la intersección entre la Robótica Educativa y el Pensamiento Computacional, estrechamente relacionado con la programación, pero haciendo énfasis en que PC no es programación [15]. El constructivismo de Piaget afirma que los humanos aprenden no solamente mediante la transmisión del saber, sino también involucrándose en actividades o conocimientos basados en su propia experiencia, de forma que la teoría es muy bien interpretada por los alumnos cuando se aplica a cuestiones o problemas de la vida real, a través de la creación y la innovación. La teoría construccionista de Papert agrega que el conocimiento será más efectivo si los aprendices, además, construyen productos relacionados con él a través de sus propias experiencias [43], convirtiéndose en aprendices activos mientras interactúan con artefactos físicos y empleando tecnología; dicho de otra forma, que el aprendizaje es verdaderamente significativo, duradero y exitoso cuando los estudiantes participan activamente en la construcción de su propio conocimiento, de forma que lo interiorizan mejor y lo manifiestan tangiblemente en el mundo real a través de sus creaciones [41], [44], trátase un robot, un castillo de arena, una historia o un programa de computadora, y luego reflexionan cuidadosamente sobre el proceso. El construccionismo, por sí mismo, incluye el involucramiento activo del aprendiz, de forma que éste construye un entendimiento o significado de lo que ha hecho. En síntesis, la robótica educativa se presenta como una herramienta propicia para el aprendizaje mediante la filosofía construccionista ya que permite trasladar la experiencia obtenida mediante la interacción de la herramienta con el entorno en un determinado contexto, en ideas que transforman las percepciones y conocimientos previos [45].

2.2.3.2 Aportes de la Robótica Educativa al Pensamiento Computacional

Partiendo de lo anterior, en concordancia con las aproximaciones constructivistas y construccionistas, la importancia de la ER para el desarrollo del PC, radica en su capacidad

de involucrar el uso de computadoras y el diseño, construcción y operación de robots, que son empleados a manera de controles, sensores y procesadores de información, a través de los cuales los estudiantes pueden lograr un valioso aprendizaje, pasando de los conceptos abstractos y poco familiares a la robótica como objeto tangible [39], [41] o, dicho de otra forma, uniendo la abstracción y la realidad de una manera concreta, pasando de lo abstracto a lo tangible [22], [40]. Las actividades de ER necesarias para ello, que incluyen montaje, programación y pruebas, resaltan las habilidades del PC [43], ya que, los estudiantes pueden pensar, diseñar, crear y manipular objetos, mientras reflexionan y colaboran entre ellos [46], además de que motivan y animan a los estudiantes a resolver problemas auténticos que son significativos para ellos, brindándoles la oportunidad de ver directa y visualmente el resultado de su solución [22]. Tratar con ER, por medio de la programación de un robot, puede dar a los estudiantes el beneficio adicional de interactuar con un objeto concreto en la construcción del conocimiento, al tiempo que los alienta a pensar activamente, analizar situaciones y aplicar pensamiento crítico y solución de problemas a problemas del mundo real [38]. Los robots programables son considerados como una herramienta cognitiva de gran valor, gracias a las características físicas y tangibles del recurso que se convierte en un factor motivador que favorece el interés de los niños [41], [42] y les ofrece la oportunidad de interesarse y desarrollar habilidades de PC [39], [47]. El uso de robots permite a los estudiantes ver el resultado de su programación, recibir realimentación visual, ya que pueden observar con mayor facilidad el comportamiento del programa y los errores a depurar, los motiva a resolver inconvenientes de programación e implementación, ofreciendo una forma concreta de engancharse con el CT [15] y facilitando el aprendizaje de sus conceptos; entonces, además de ser una oportunidad para aprender sobre sensores y programación, la RE también permite aprender sobre secuenciación, depuración y bucles de programación [48].

La robótica es uno de los recursos de tecnología educativa que actualmente se están utilizando para fomentar el desarrollo de actividades educativas que faciliten el aprendizaje del pensamiento computacional y las habilidades de programación [49]. La integración de las actividades con robótica en el proceso de enseñanza-aprendizaje es posible, entre otras formas, al vincular las primeras con la construcción y programación de robots, empleando engranes, sensores, actuadores y lenguajes de programación, por mencionar algunas. [41], [42]. Además, la ER ha sido empleada para actuar como herramienta de construcción y programación para promover la solución de problemas y el desarrollo del PC., la codificación y la ingeniería. Por esto, la ER no solamente ofrece una plataforma apropiada para el desarrollo de habilidades en forma divertida y significativa, sino que también proporciona oportunidades para lidiar con un rango de disciplinas como ciencia, tecnología, ingeniería y matemáticas (STEM), alfabetización, estudios sociales, danza, música y arte [38].

Además de los beneficios aportados directamente por la RE al PC, Isaini y Budiyanto, agregan que el uso de robots motiva el aprendizaje de los estudiantes [47] y que puede incrementar la confianza durante el aprendizaje [50], así como también, según otros estudios, produce una mejora significativa en la comunicación, el trabajo colaborativo, la creatividad, la autonomía, el liderazgo [40], [51], el interés y la motivación [52].

Resumiendo, la RE es un camino que ha sido probado en múltiples ocasiones para acercar el PC a los estudiantes. La RE proporciona un conjunto de materiales que satisfacen la necesidad de recursos del PC, al tiempo que las actividades robóticas aportan una madurez experimental que puede ayudar a que la teoría del PC se transforme en práctica exitosa [15].

2.2.4 Eficacia de la Robótica Educativa en el desarrollo del Pensamiento Computacional

Algunas de las publicaciones teóricas seleccionadas, como [39], [47] y [50] afirman haber encontrado evidencia de que la RE no sólo facilita el desarrollo de las habilidades de PC de los estudiantes, sino que produce un impacto positivo en ellas. En forma más cautelosa, Anwar et al. sugieren que tal evidencia no es suficiente para llegar a una conclusión sobre la efectividad de los robots educativos en la respuesta de aprendizaje y habilidades de los estudiantes [53]. En forma similar, Para Ioannou y Makridou [38], el intento de apoyar el desarrollo del PC a través de la robótica educativa parece haber resultado en el desarrollo de habilidades relevantes del siglo XXI; sin embargo, la mejora del PC como una habilidad más compleja que incluye descomposición, abstracción, algoritmos y depuración no estaba del todo clara; ellos encontraron un número cada vez mayor de investigadores argumentando que es posible mejorar varias habilidades del PC a través de la robótica educativa, particularmente algunas como descomposición, abstracción, algoritmos y depuración; creen, sin embargo, que aún hay muy poca evidencia empírica para confirme o rechace estos resultados.

Pasando a hablar de las 18 publicaciones que reportan 15 diferentes experiencias de aplicación de la RE para el desarrollo de habilidades del PC, puede decirse que han estudiado un amplio rango de ellas, recordando que no existe una definición única y estandarizada ni del PC, ni mucho menos de las habilidades que la componen. En el anexo B se muestra una tabla indicando los estudios de este tipo que se revisaron, relacionándolos con la edad o grado de escolaridad de los participantes, el robot (hardware) y el entorno o lenguaje de programación (software) empleados, las habilidades de PC y otros conceptos que se busca desarrollar o reforzar, así como el tipo de evaluación aplicado y la existencia o no de un grupo de control.



Figura 3. Relación entre las habilidades del PC y el número de veces estudiada en los estudios prácticos.

Las habilidades de PC referidas en estos estudios son secuenciación, descomposición, modularidad, algoritmos, abstracción, generalización de patrones, depuración y otras como correspondencia instrucción-acción o pensamiento lógico que, no siendo menos importantes, no se encuentran cubiertas por alguna de las definiciones trabajadas en el presente proyecto. Como puede apreciarse en la figura 3, los algoritmos son la habilidad que más se estudia, en 10 de los 15 estudios, seguida por abstracción y generalización de patrones, ambas en 9 estudios. Entre otros conceptos que los investigadores persiguen desarrollar con este tipo de estudios están la colaboración, interés, motivación, autoconfianza, creatividad, autonomía, liderazgo y comunicación.

Desafortunadamente, la edad de los participantes no aparece en todos los estudios en forma explícita y, en varios casos, solamente puede inferirse aproximadamente a partir de su escolaridad; puede decirse, eso sí, que no es uniforme y que cubre desde los 3 años hasta estudiantes en edad universitaria.

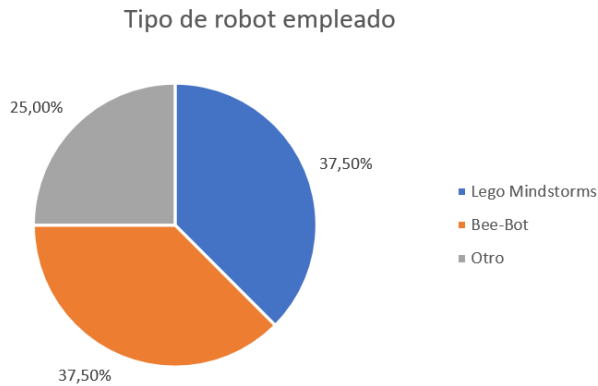


Figura 4. Tipos de robot empleados en los estudios.

En la figura 4 se observa el tipo de robot utilizado, siendo los más populares Lego Mindstorms, en cualquiera de sus variantes, y Bee-Bot, ambos con un 37,50%. Es justo mencionar aquí la existencia de muchísimas herramientas de este tipo, como las incluidas por González en [45], destacando KIBO, Blue-Bot y el mismo Bee-Bot, Roamer, Cubetto y Code a Pillar, a los que se puede añadir mBot, Makey Makey y muchos otros. Puede verse también que el tipo de robot a utilizar guarda una fuerte relación con la edad de la población objeto del estudio, estando Bee-Bot dirigido a los más jóvenes, a partir de los 3 años, y Lego Mindstorm en edades mayores.

La forma de evaluar los resultados también es bastante variada, y abarca técnicas como entrevistas, cuestionarios, tests (antes y después), quizzes, rúbricas, “think-aloud”, etc.

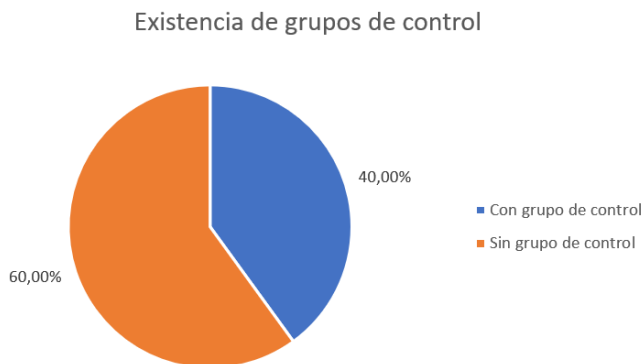


Figura 5. Existencia de grupos de control entre los estudios.

La figura 5 permite comparar el porcentaje de estudios que emplean un grupo de control como medio para contrastar los resultados. Tan sólo 6 de los 15 estudios presentan un grupo de control, equivalente a un 40%, contra 9 que no lo hacen y que corresponde al 60%.

En cuanto a los resultados particulares de los estudios en cuestión, Phetsrikran et al. [54] dicen haber conseguido un claro patrón de mejoramiento del PC (lógica, descomposición, algoritmos, abstracción, patrones y evaluación). En [17], Atmatzidou y Demetriadis

sostienen que los estudiantes se familiarizan con los conceptos de CT y los integran satisfactoriamente al solucionar problemas en actividades de ER. En su investigación, los estudiantes desarrollaron con bastante éxito las habilidades de CT, siendo algoritmos, modularidad y descomposición aquellas con que se familiarizaron más y en menor tiempo, y abstracción y generalización con las que tuvieron mayor dificultad. Vallance y Towndrow [55], apoyados en este trabajo previo de Atmatzidou y Demetriadis, confirmaron estos mismos hallazgos. También Aristawati et al. [43] reportan que los participantes en su estudio mostraron variados grados de habilidades de PC.

Más adelante, Atmatzidou y Demetriadis retoman su trabajo en [56], sugiriendo con su investigación que ni la edad ni el género son impedimento para que los estudiantes alcancen, eventualmente, el mismo nivel de desarrollo de habilidades de Pensamiento Computacional y que éstas, en la mayoría de los casos, requieren tiempo para desarrollarse completamente. Es importante traer a colación aquí, como aporte para este proyecto, su modelo para el desarrollo de algunas habilidades de CT y que se indica en la tabla 1.

Habilidad de PC	Definición	Guía para su desarrollo
Abstracción	Es el proceso de crear algo simple a partir de algo complicado, omitiendo los detalles irrelevantes, encontrando los patrones relevantes y separando ideas de los detalles tangibles.	<ol style="list-style-type: none"> 1. Separar la información importante de la redundante. 2. Analizar y especificar comportamientos comunes o estructuras de programación entre scripts diferentes. 3. Identificación de abstracciones entre ambientes de programación diferentes.
Generalización	Es transferir un proceso de solución de un problema a una amplia variedad de problemas.	<ol style="list-style-type: none"> 1. Expandir una solución existente en un problema dado para cubrir más posibilidades / casos.
Algoritmos	Es la práctica de escribir, paso a paso, instrucciones específicas y explícitas, para llevar a cabo un proceso.	<ol style="list-style-type: none"> 1. Describir explícitamente los pasos del algoritmo. 2. Posibilidad de diferentes algoritmos para el mismo problema. 3. Esfuerzo para encontrar el algoritmo más efectivo.
Modularidad	Es el desarrollo de procesos autónomos que encapsulan un conjunto de órdenes usadas con frecuencia que ejecutan una función específica y que podrían utilizarse en el mismo o en diferentes problemas.	Desarrollar secciones de código autónomas para ser utilizadas para el mismo o para diferentes problemas.

Descomposición	Es el proceso de separar los problemas en partes más pequeñas que pueden resolverse con mayor facilidad.	Separar problemas en partes más pequeñas / sencillas que son más fáciles de resolver.
----------------	--	---

Tabla 1. Modelo para el desarrollo de habilidades de CT propuesto por Atmatzidou y Demetriadis, condensado de [56] y [22].

Los resultados reportados en varios estudios sucesivos de Caballero, junto con otros autores [40], [51], [57], [44], [49], [41] y [42] muestran un avance significativo en el pensamiento computacional (abstracción, secuenciación, algoritmos, patrones y depuración), encontrando diferencias significativas entre los miembros del grupo experimental y el de control que apoyan su argumento. Dicen demostrar, por tanto, que es posible desarrollar cada una de las habilidades de PC analizadas, así como la comunicación y la colaboración, a través de actividades de RE.

Para Liu y Rojas [58], la realimentación conseguida de los niños indicó que la robótica fue una herramienta motivante para aprender conceptos de CT, en comparación con trabajar solamente con pantallas, debido a que los estudiantes pueden controlar directamente un robot físico cuando lo programan, lo que captura la atención de la mayoría de ellos. En [52], Haut et al. afirman haber notado muestras de interés y motivación en los estudiantes, al igual que de signos de pensamiento computacional como abstracción, descomposición y desarrollo de algoritmos; del pensamiento computacional, eso sí, concuerdan con Atmatzidou y Demetriadis en que su adquisición debe ser el resultado de un proceso continuo en el tiempo.

En otro tipo de estudio, Huang et al. [48] comparan el efecto de emplear Dash & Dot contra un grupo de control que utiliza LEGO NXT en cuanto a desempeño y actitud sobre las actividades de aprendizaje, teniendo en cuenta el género de los estudiantes. Su conclusión es que el aprendizaje no se ve afectado por el robot empleado.

A modo de observación, entre los estudios revisados, se halló que solamente en [17] y [56] se habla de la asignación de roles entre los participantes, siendo estos el de analista, diseñador de algoritmos y programador o depurador / evaluador.

2.3 RECOMENDACIONES DE LOS INVESTIGADORES

A pesar de la importancia que recientemente ha cobrado, la ER, por sí sola no afecta directamente el aprendizaje de los estudiantes; se requiere de una filosofía educacional, un ambiente de aprendizaje y una metodología de enseñanza apropiadas para su incorporación exitosa en la enseñanza. No está oculta la necesidad de más investigación sobre los beneficios de la ER en el desarrollo de habilidades específicas, tales como el pensamiento computacional y la solución de problemas [22]. La mayoría de estudios carece de un diseño experimental o cuasiexperimental y, aun habiendo suficiente literatura sobre PC y ER, hay muy poco trabajo conectando sus bases teóricas con su implementación [38], [53], siendo notoria la necesidad de continuar investigando sobre la efectividad de la RE para el desarrollo del PC [50], pues el impacto de la primera en el segundo necesita ser validada a través de evidencia investigativa. A pesar del reclamo de Atmatzidou y Demetriadis, entre muchos, para quienes existe una carencia empírica al definir los límites explícitos de CT [17], otros como Liu y Rojas [58] subrayan que varios estudios se enfocan

en conceptos de PC ligeramente diferentes, discutiendo repetidamente sobre su definición e importancia, pero pasando por alto la investigación del desarrollo de CT. Finalmente, Ioannou [38] pone de manifiesto la exigencia de un marco práctico para el desarrollo de CT a través de la robótica, de modo que los diseñadores instruccionales y los educadores puedan implementarlo de manera consistente y a escala.

3 CHILDPROGRAMMING-RE: UNA EXTENSIÓN DE CHILDPROGRAMMING PARA ROBÓTICA EDUCATIVA.

En el marco de la metodología ChildProgramming, se buscan permanentemente nuevas prácticas que permitan una mejor apropiación de los conceptos de Pensamiento Computacional. Es bajo esta visión que en este proyecto se ha trabajado en la creación de ChildProgramming-RE como una extensión que hace uso de la Robótica Educativa como herramienta para la motivación y participación en el desarrollo de habilidades específicas del PC en niños.

Para alcanzar su objetivo, esta extensión brinda un conjunto de directrices concretas para la utilización de aspectos de Robótica Educativa durante actividades encaminadas hacia el desarrollo del Pensamiento Computacional en niños entre los 10 y los 12 años siguiendo la dirección sugerida por Benitti [21], quien precisamente señala, como línea de investigación, el uso de la robótica como herramienta para el desarrollo de habilidades de pensamiento; además, se tienen en cuenta los elementos encontrados en el estado del arte presentado en el capítulo 2 y los hallazgos del estudio exploratorio que se presenta en el capítulo 4.



Figura 6. Participantes programando en mBlock durante el estudio exploratorio.

Aunque, en términos generales, la extensión propuesta no depende un de un lenguaje de programación particular ni de alguno de los kits robóticos educativos comerciales existentes, fue evaluada en el contexto del lenguaje de programación visual mBlock (figura 6), basado en Scratch, y del kit robótico mBot Ranger, ambos del fabricante Makeblock.



Figura 7. El mBot Ranger, de Makeblock.

El mBot Ranger (figura 7) está construido con aluminio extruido y plástico, principalmente, haciéndolo apropiado para este tipo de prácticas en el rango de edad requerido.

3.1 LA EXTENSIÓN CHILDPROGRAMMING-RE

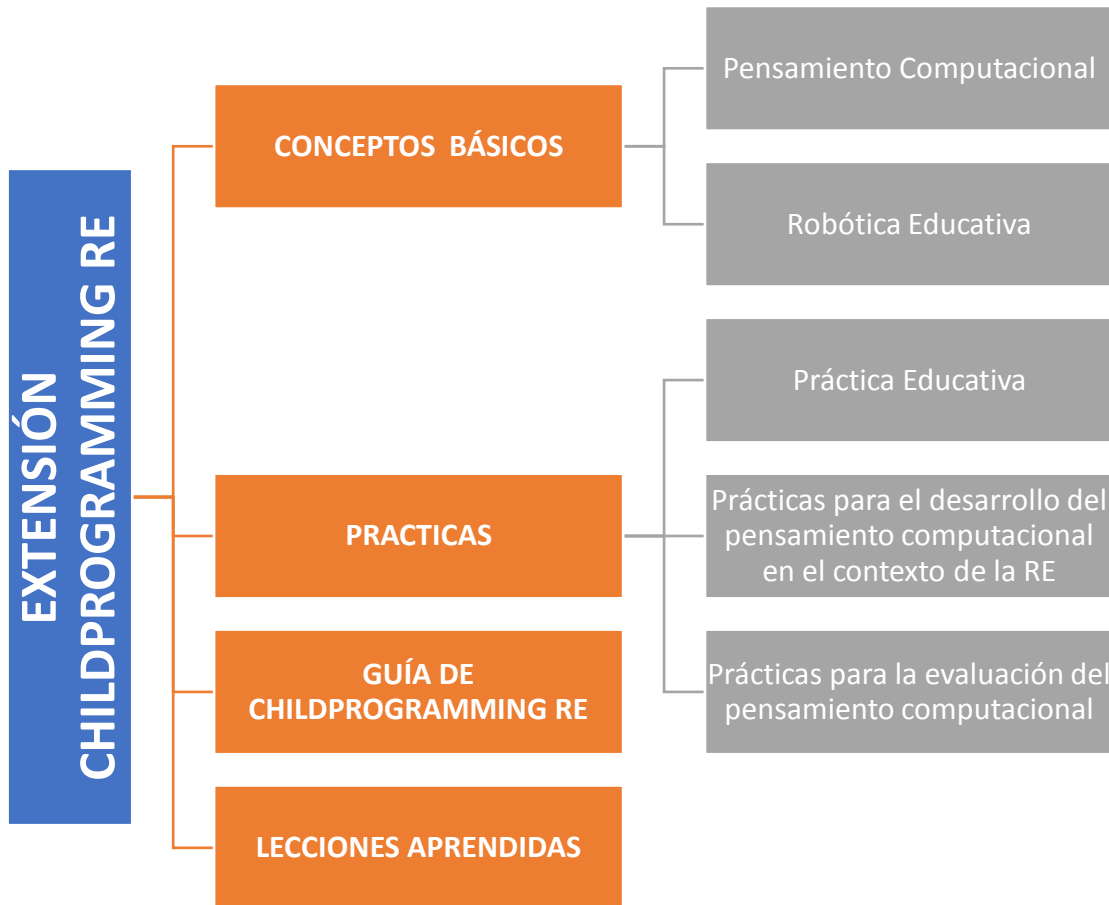


Figura 8. Extensión de Childprogramming y sus paquetes metodológicos (Fuente Propia).

La extensión Childprogramming-RE, tal como se muestra en la figura 8, está organizada en 4 paquetes metodológicos, que se describen a continuación:

1. Conceptos básicos, que integran el pensamiento computacional con la robótica educativa. Este paquete describe los conceptos básicos de Pensamiento Computacional y de robótica que tanto el profesor como los alumnos deben conocer para entender el desarrollo de las sesiones de entrenamiento.
2. Prácticas CT-RE: Se describen tres tipos de prácticas que buscan usar la Robótica Educativa como una herramienta para el desarrollo del pensamiento computacional:
 - 2.1. Prácticas Educativas: se centran en la organización de los grupos de trabajo.
 - 2.2. Prácticas para el desarrollo del Pensamiento Computacional en el contexto de la Robótica Educativa. Estas prácticas permiten integrar los conceptos de Pensamiento Computacional con aplicaciones de la Robótica Educativa.
 - 2.3. Prácticas para la evaluación del Pensamiento Computacional. El objetivo de estas prácticas es evaluar y verificar la asimilación de los conceptos del Pensamiento Computacional.
3. Guía práctica de ChildProgramming-RE. Esta guía tiene una serie de sesiones bien definidas para orientar un curso de robótica, siguiendo el modelo planteado en ChildProgramming.
4. Lecciones aprendidas: Son recomendaciones que se dan hacen para realizar una sesión o desarrollar un curso de robótica para niños.

En las subsecciones que prosiguen se describe en detalle cada uno de estos paquetes metodológicos.

3.2 CONCEPTOS BÁSICOS DEL PENSAMIENTO COMPUTACIONAL Y ROBÓTICA

En este primer paquete metodológico se establecen los conceptos, tanto del Pensamiento Computacional, como de la Robótica, con el fin de dar claridad conceptual al profesor que orientará los talleres, siguiendo esta extensión metodológica de Childprogramming. El paquete consta de los subpaquetes “Conceptos del pensamiento computacional” y “Conceptos de robótica educativa”, que se presentan en la figura 9 y se describen a continuación.

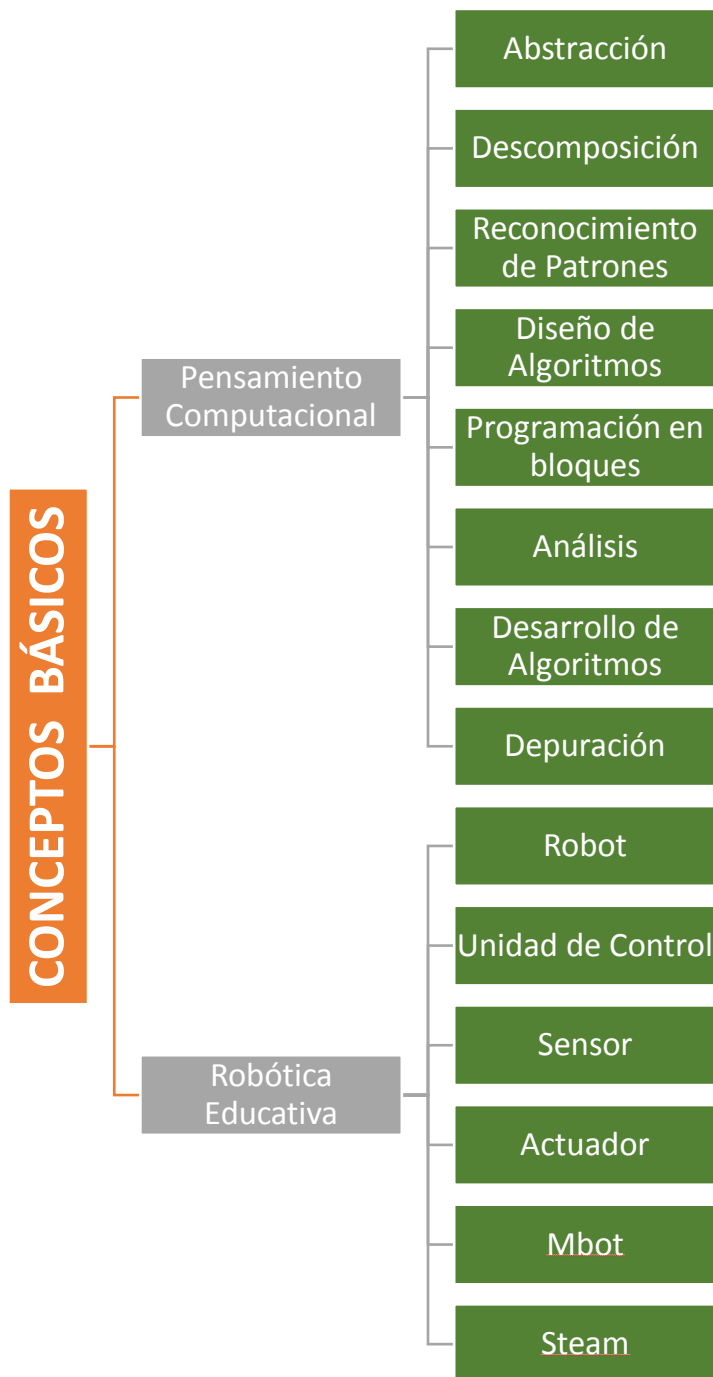


Figura 9. Conceptos básicos y sus subpaquetes (fuente propia).

3.2.1 Conceptos del Pensamiento Computacional

“El pensamiento computacional incluye los procesos de pensamiento implicados en la formulación de problemas y de sus soluciones, de modo que queden representados de una manera que pueda ser abordada de forma efectiva por un agente-procesador de la información, como una computadora” [24]. En este tipo de pensamiento se enmarcan los siguientes conceptos fundamentales [22]:

- **Abstracción:** es la habilidad de identificar y extraer información relevante de un problema o una solución.
- **Descomposición:** habilidad de separar los datos, procesos, soluciones o problemas en partes más pequeñas y manejables.
- **Reconocimiento de patrones:** Transferir un proceso de solución de un problema a una amplia variedad de problemas.
- **Algoritmo:** Crear una serie ordenada de instrucciones para que una máquina o persona al ejecutarlos, resuelva un problema o realice una tarea.
- **Programación basada en bloques:** La programación basada en bloques implica encastrar piezas prediseñadas (como en un rompecabezas) de forma tal que se genere una lista de pasos o acciones a seguir para solucionar un problema planteado o creado por el mismo usuario. Es una manera práctica y visual de programar en entornos gráficos.
- **Análisis:** Consiste en el estudio detallado del problema. Se deben identificar los datos de entrada, de salida y la descripción del problema.
- **Desarrollo de algoritmos:** práctica de escribir, paso a paso, instrucciones específicas y explícitas para llevar a cabo un proceso.
- **Depuración:** Es el proceso de identificar y corregir errores de programación.

3.2.2 Conceptos de Robótica [59]

- **Robot:** es una máquina programable que puede sentir el entorno físico y ejecutar distintos tipos de operaciones y movimientos de forma automática.
- **Unidad de Control:** dispositivo central del robot que procesa la información proporcionada por los sensores (entrada) y toma decisiones de acuerdo con dicha información (salida).
- **Sensor:** es un dispositivo electrónico que recolecta información del entorno físico y la envía a la unidad de control en forma de una señal eléctrica.
- **Actuador:** es un dispositivo que realiza una acción sobre el entorno físico a partir de los datos de los sensores y/o que permite actuar sobre un determinado objeto del entorno físico.
- **mBot:** mBot Bluetooth es un kit STEAM de robótica ideal para que los niños se inicien en la robótica, programación y electrónica basado en Arduino y Scratch.
- **STEAM:** Iniciales de las palabras en inglés que identifican a las disciplinas Science, Technology, Engineering, Art y Mathematics, es decir, ciencia, tecnología, ingeniería, arte y matemáticas.

3.3 PRÁCTICAS DE CHILDPROGRAMMING-RE

Este paquete metodológico permite al docente, llámase instructor o profesor, y a los estudiantes, integrar en forma práctica los conceptos de Pensamiento Computacional y Robótica Educativa, mejorando aspectos como

- El trabajo colaborativo en las actividades de la práctica de organización de las actividades, de forma tal que todos los integrantes del grupo trabajen de forma activa alrededor de un objetivo común.
- Las estrategias planteadas por los grupos de trabajo cuando buscan la solución a las misiones propuestas en ChildProgramming.

- La evaluación de los conceptos teóricos transmitidos durante las sesiones de entrenamiento.

El paquete de prácticas de ChildProgramming-RE se ha dividido en los subpaquetes: Prácticas educativas, Prácticas para el desarrollo del Pensamiento Computacional en el contexto de la RE y Prácticas para la evaluación del Pensamiento Computacional. La figura 10 muestra en forma gráfica la distribución de las prácticas propuestas por la extensión de Childprogramming-RE.

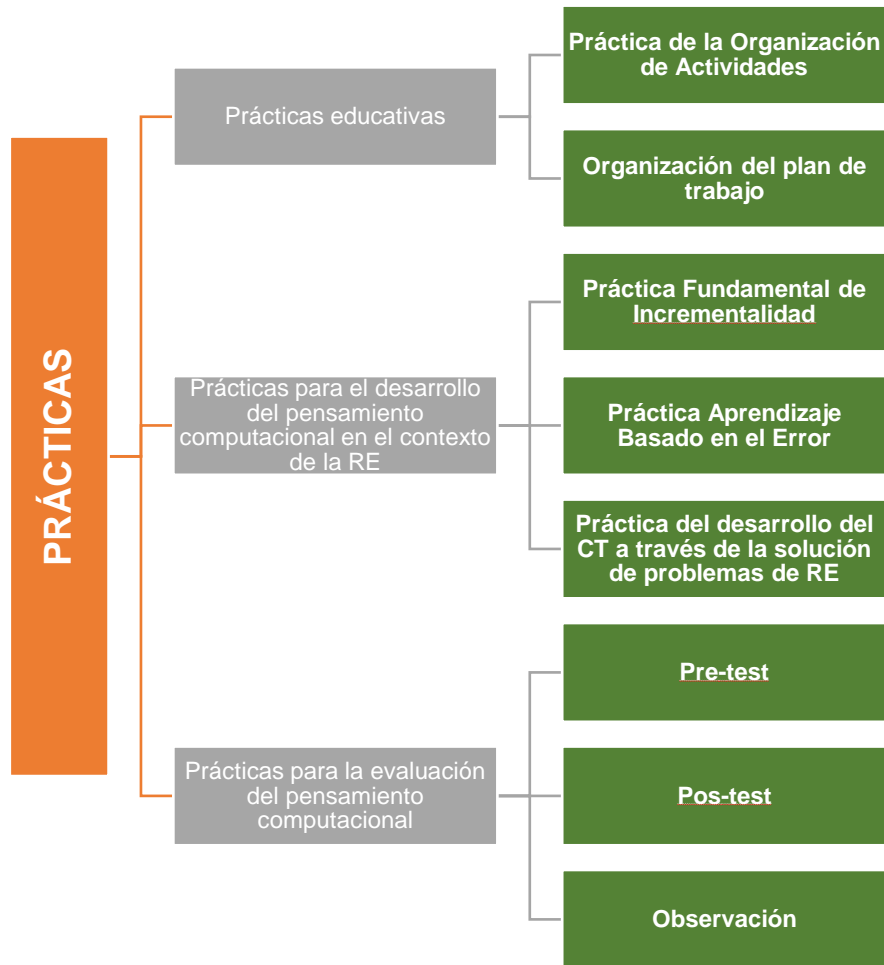


Figura 10. Clasificación de las prácticas para Childprogramming con RE. Fuente Propia.

Cada una de las prácticas de estos subpaquetes, se describen mediante una estructura, a manera de plantilla, que corresponde a la tabla 2. Esta estructura consta de un Título, una Descripción, a quién se encuentra Dirigida, el Contexto, las Actividades a realizar, los Requisitos de entrada y los Resultados esperados.

Título	•Nombre de la práctica.
Descripción	•Describe de forma general la práctica y el propósito de esta.
Dirigido a:	•Personas que deben realizar la práctica.
Contexto	•Nombra el escenario o punto de la sesión de entrenamiento dónde debe aplicarse la práctica.
Actividades a realizar	•Se enuncian actividades que permitan lograr el propósito de la práctica.
Requisitos de entrada	•Son los insumos o los elementos necesarios para el desarrollo de la práctica.
Resultados esperados	•El comportamiento de los grupos de trabajo o el producto de la realización de la práctica.

Tabla 2. Estructura para la descripción de prácticas. Fuente propia

3.3.1 Prácticas educativas

Con las prácticas educativas se busca mejorar la organización de los grupos de trabajo durante una sesión de entrenamiento en un curso de robótica educativa. En este subpaquete se incluyen dos actividades, una práctica llamada Organización de Actividades (Tabla 3) y Organización del Plan de Trabajo (Tabla 4).

Práctica de la Organización de Actividades
<i>Descripción:</i>
Esta práctica está orientada a asignar las funciones y distribuir los equipos de trabajo durante una sesión de clase. Es una manera de fomentar en los estudiantes el trabajo colaborativo y mejorar la asignación de roles en el desarrollo de las misiones propuestas en cada sesión de entrenamiento.
<i>Dirigida a:</i>
Docentes o instructores
<i>Contexto:</i>
Organizar a los grupos de trabajo durante una sesión de un curso de robótica educativa.
<i>Actividades a realizar:</i>
✓ A excepción de la última, las sesiones se dividen en dos partes: una primera parte de entrenamiento, en la que se exponen conceptos de programación y de robótica, y una segunda parte, en la que se pone en práctica lo aprendido, para resolver un reto propuesto por el (los) instructor(es). La duración sugerida de cada sesión es de cuatro (4) horas, siendo posible dividir cada una en dos (2) sesiones de dos (2)

<p>horas (una de entrenamiento y una de práctica). Paralelamente, los conceptos de pensamiento computacional se refuerzan todo el tiempo, no siempre de forma explícita.</p> <ul style="list-style-type: none"> ✓ Los equipos de trabajo deben conformarse en la primera sesión, antes de comenzar con el ensamblaje del robot. Es recomendable que los estudiantes sean asignados de forma aleatoria en los equipos. Se sugiere que el número de miembros de cada equipo esté entre 2 y 4 estudiantes, como máximo. ✓ Los roles que adoptarán los estudiantes serán el de “Analista”, “Desarrollador de algoritmos”, “Programador” y “Depurador”. En general, es deseable que sean los propios estudiantes quienes decidan los roles que desempeñarán en el equipo y el orden en que los rotarán; sin embargo, es necesario asegurarse de que tanto la asignación como la rotación de papeles ocurran, evitando que uno o varios participantes monopolicen o se adueñen de alguna parte del trabajo, restringiendo la participación de los demás. Esto se observa con frecuencia durante el ensamblaje, la programación y la puesta en marcha del robot.
<i>Requisitos de entrada</i>
<p>Personas que participan en el curso de robótica:</p> <ul style="list-style-type: none"> ● El grupo total de estudiantes ● El profesor (docente o instructor del curso de robótica) <p>Tener un espacio de trabajo adecuado y contar con los materiales necesarios para el desarrollo de la misión.</p>
<i>Resultados esperados:</i>
Participación activa de cada integrante del grupo en el cumplimiento de las misiones.

Tabla 3. Organización de Actividades.

Práctica de la Organización del Plan de Trabajo
<i>Descripción:</i>
Organizar y planear la distribución del trabajo de los integrantes de cada uno de los equipos. durante el desarrollo de las misiones.
<i>Dirigida a:</i>
Docentes o instructores
<i>Contexto:</i>
Durante una sesión de un curso de robótica educativa.
<i>Actividades a realizar:</i>
<ul style="list-style-type: none"> ✓ Realizar una lista de los roles a desempeñar, entre ellos están: <ul style="list-style-type: none"> ○ Analista

<ul style="list-style-type: none"> ○ Programador ○ Depurador ○ Desarrollador de algoritmos <p>✓ Hacer una lista de las tareas y asignárselas a los integrantes del equipo según su rol.</p> <p>✓ Realizar la secuencia de las tareas y asignar un tiempo para desarrollarlas.</p> <p>✓ Hacer una lista de chequeo de las tareas terminadas y las pendientes.</p>
<i>Requisitos de entrada</i>
<p>Personas que participan en el curso de robótica:</p> <ul style="list-style-type: none"> ● El grupo total de estudiantes ● El profesor (docente o instructor del curso de robótica) <p>Tener un espacio de trabajo adecuado y contar con los materiales necesarios para el desarrollo de la misión.</p>
<i>Resultados esperados:</i>
<ul style="list-style-type: none"> ● Participación activa de cada integrante del grupo en el cumplimiento de las misiones ● Optima distribución de las tareas ● No se recarga el trabajo a una sola persona.

Tabla 4. Organización del plan de trabajo.

3.3.2 Prácticas para el Desarrollo del Pensamiento Computacional en el Contexto de la Robótica Educativa

Las siguientes prácticas proponen actividades que permiten desarrollar el Pensamiento Computacional en misiones de ChildProgramming con aplicaciones de Robótica Educativa.

Práctica Fundamental de descomposición e incrementalidad.

Práctica Fundamental de Descomposición e Incrementalidad
<i>Descripción:</i>
El objetivo de esta práctica es estimular a los grupos de trabajo para que realicen un plan de trabajo y vayan probando la solución por partes de código (en la programación) o por partes físicas (al ensamblar el robot).
<i>Dirigida a:</i>
Docentes o instructores.
<i>Contexto:</i>
En el planteamiento de las estrategias en las misiones de Childprogramming con RE
<i>Actividades a realizar:</i>

<ul style="list-style-type: none"> ✓ Pedir a los estudiantes una documentación del plan a seguir durante el desarrollo de la misión, ya sea un algoritmo como diagrama de flujo o como una secuencia explícita de instrucciones. ✓ No es necesario que el análisis y el algoritmo sean perfectos antes de pasar a programar y probar el robot. Basta con que hayan seguido un proceso lógico y que sean razonablemente aceptables (admitiendo que éste es un criterio subjetivo), considerando que la corrección y el mejoramiento hacen también parte de las prácticas en las sesiones. ✓ Probar cada proceso por partes les permitirá el mejoramiento y la evolución en el aprendizaje y lograr el objetivo propuesto en la misión.
<i>Requisitos de entrada</i>
<ul style="list-style-type: none"> • Algoritmos o plan de trabajo de cada equipo. • Código de programación en bloques. • Robot en funcionamiento.
<i>Resultados esperados:</i>
<p>Los estudiantes, pueden reconocer los bloques de programación y los dispositivos físicos del robot.</p> <p>Los equipos prueban la funcionalidad del robot por partes, en paralelo la programación y la parte física del robot.</p> <p>Los equipos alcanzan la misión en forma incremental</p>

Tabla 5. Práctica Fundamental de Incrementalidad.

Práctica Aprendizaje Basado en el Error

Práctica Aprendizaje Basado en el Error
<i>Descripción:</i>
Esta práctica invita al docente a brindar a los estudiantes un espacio de comodidad y confianza, que le permita aceptar el error como una estrategia de aprendizaje.
<i>Dirigida a:</i>
Docentes o instructores.
<i>Contexto:</i>
En el momento de hacer las pruebas del código en bloques y el ensamble del robot.
<i>Actividades a realizar:</i>
<ul style="list-style-type: none"> ✓ Al iniciar cada sesión de entrenamiento crear un ambiente de confort, por medio de charlas de retroalimentación con los estudiantes, de forma tal que puedan ver al docente como un consultor. ✓ Dar asesorías a cada estudiante cuando éste lo solicite, o cuando se observa un estancamiento en el desarrollo de la misión.

<ul style="list-style-type: none"> ✓ En cuanto a la asesoría prestada a los equipos durante las prácticas, la intervención del (los) instructor(es) debe ser limitada, intentándose, siempre que sea posible, que sean los mismos estudiantes quienes se percaten de los problemas y se ocupen de su solución, siguiendo la secuencia de preguntas ¿qué está fallando?, ¿por qué está fallando? y ¿cómo se corrige? No debe llegarse nunca al extremo, eso sí, de dejar que los equipos caigan en un punto de estancamiento en el que sienten que han dado con algo que no pueden resolver. Es necesario recordarles (y demostrarles) que hay más de una manera de solucionar un problema. ✓ Estimular a los estudiantes para abordar tareas que ellos consideren difíciles mostrándole alguna estrategia para solucionarlas. ✓ Se puede seguir estrategias como las planteadas en el modelo de ChildDebugging [60] como, por ejemplo, reducir el código, análisis paso a paso, razonamiento hacia adelante o hacia atrás, manejo de errores, prueba y error entre otras.
<i>Requisitos de entrada.</i>
<ul style="list-style-type: none"> ● Docente atento y dispuesto a asesorar a los equipos de trabajo. ● Algoritmos y códigos de programación de cada equipo de trabajo. ● Detección de los errores presentados en las pruebas de funcionamiento del robot. ● Robot armado y equipo de cómputo.
<i>Resultados esperados:</i>
<ul style="list-style-type: none"> ● Los estudiantes buscan e identifican defectos en su solución. ● Los estudiantes empiezan a plantear por sí solos la posible solución a un problema que se les había presentado anteriormente. ● Emplear una herramienta de soporte “Bug Box” mencionada en [60], que le permite encontrar una posible solución a los errores en la programación de las misiones propuestas en las sesiones de entrenamiento.

Tabla 6. Práctica Aprendizaje Basado en el Error.

Práctica del desarrollo del pensamiento computacional a través de la solución de problemas de robótica educativa.

Práctica del desarrollo del pensamiento computacional a través de la solución de problemas de robótica educativa.
<i>Descripción:</i>
Se dan unas pautas para combinar los conceptos de pensamiento computacional con robótica educativa.
<i>Dirigida a:</i>
Docentes o instructores.
<i>Contexto:</i>

Los conceptos de pensamiento computacional se deben tener en cuenta tanto al inicio del planteamiento de la estrategia como durante el desarrollo de la misma, ya sea en una sesión de Robótica Educativa o de una actividad de Childprogramming.

Actividades a realizar:

- ✓ Dar a conocer los conceptos del pensamiento computacional y la robótica educativa, en la estructura misma del robot, de sus partes y aplicarlos a las actividades propias de la robótica, como lo es el ensamblaje del robot, en la primera aproximación que se les hace a los estudiantes.
- ✓ En [3], Atmatzidou y Demetriadis proponen el siguiente modelo de habilidades de CT, que se muestra en la siguiente tabla.

Habilidad	Guía para su desarrollo
Abstracción	<ul style="list-style-type: none"> • Pedir a los estudiantes que separen la información importante de la redundante. • Analizar y especificar comportamientos comunes o estructuras de programación entre scripts diferentes. • Identificación de abstracciones entre ambientes de programación diferentes.
Generalización	<ul style="list-style-type: none"> • Expandir una solución existente en un problema dado para cubrir más posibilidades / casos. • Emplear variables en una solución
Modularidad	<ul style="list-style-type: none"> • Desarrollar secciones de código autónomas para ser utilizadas para el mismo o para diferentes problemas.
Descomposición	<ul style="list-style-type: none"> • Separar problemas en partes más pequeñas / sencillas que son más fáciles de resolver.

Tabla de habilidades del CT y guía para su desarrollo.

Requisitos de entrada:

- Exposición o asesoría del docente sobre cada concepto del pensamiento computacional.
- Códigos de programación.
- Algoritmos o planes de trabajo.
- Descomposición y distribución de tareas en ellos equipos de trabajo.

Resultados esperados:

Los estudiantes reconocen los conceptos de pensamiento computacional ya sea de forma explícita o por medio de ejemplos, al interactuar con el robot y su programación.

Tabla 7. Práctica del desarrollo del pensamiento computacional a través de la solución de problemas de robótica educativa.

3.3.3 Prácticas de Evaluación

Estas prácticas permiten la evaluación de conceptos, previos o aprendidos, y recopilar información necesaria para mejorar tanto la forma de orientar las sesiones como su planeación. Son ellas el Pre-test (Tabla 8), el Pos-test (Tabla 9) y la Observación (Tabla 10).

Pre-test
<i>Descripción:</i>
Es una evaluación que se hace al inicio de una sesión de entrenamiento o al iniciar un curso (ya sea de robótica educativa o de programación), con el fin saber que conocimientos previos se tienen de los temas a desarrollar en las sesiones de entrenamiento.
<i>Dirigida a:</i>
Docentes o instructores.
<i>Contexto:</i>
En un curso de robótica educativa o en un curso de programación para niños de 10 a 12 años.
<i>Actividades a realizar:</i>
Este test, recomienda aplicarlo antes de iniciar la primera sesión de entrenamiento. ✓ En la primera sesión de clase aplicar un test escrito, ya sea preguntas de selección múltiple o abiertas. ✓ Explicar al estudiante que no hay preguntas buenas o malas, sólo se realiza con fines informativos.
<i>Requisitos de entrada:</i>
Test escrito u oral con preguntas sobre los conceptos a tratar durante el desarrollo del curso.
<i>Resultados obtenidos:</i>
Nivel de desarrollo del pensamiento computacional del niño antes de iniciar cualquier actividad de aprendizaje relacionada.

Tabla 8. Pre-test

Pos-test
<i>Descripción:</i>
Test o evaluación que se realiza al final del curso, con el fin de evaluar los conceptos aprendidos durante el desarrollo de las sesiones de entrenamiento.
<i>Dirigida a:</i>

Docentes o instructores.
<i>Contexto:</i>
En un curso de robótica educativa o en un curso de programación para niños de 10 a 12 años.
<i>Actividades a realizar:</i>
Estas actividades son recomendadas para la penúltima o última sesión. ✓ Aplicar un test escrito, ya sea con preguntas de selección múltiple o abiertas, similares a las del pretest ✓ Explicar al estudiante que no hay preguntas buenas o malas, sólo se realiza con fines informativos.
<i>Requisitos de entrada.</i>
<ul style="list-style-type: none"> • Test escrito u oral con preguntas relacionadas sobre los conceptos tratados durante el desarrollo del curso. • Soluciones a las misiones propuestas (código de programación en bloques o robot funcionando)
<i>Resultados esperados:</i>
Nivel de desarrollo del pensamiento computacional del niño después de completar todas las actividades de aprendizaje planeadas.

Tabla 9. Pos-test

Observación
<i>Descripción:</i>
La observación implica mirar el trabajo de los equipos y, además, indagar sobre las dificultades que se presentan en el desarrollo de las misiones, así como escuchar las soluciones planteadas a los problemas propuestos.
<i>Dirigida a:</i>
Docentes o instructores.
<i>Contexto:</i>
En un curso de robótica educativa o en un curso de programación para niños de 10 a 12 años.
<i>Actividades a realizar:</i>
<ul style="list-style-type: none"> ✓ Tomar nota de los motivos que generan inconformidad en cada equipo. ✓ Preguntar continuamente a los estudiantes sobre su progreso y, de alguna manera, hacer preguntas que les permitan identificar si están aplicando correctamente los conceptos explicados previamente.

<ul style="list-style-type: none"> ✓ Separar, aislar y, de ser necesario, reemplazar cualquier material o componente esencial que esté presentando fallas y que, en consecuencia, no permita un buen desarrollo de la misión.
<i>Requisitos de entrada:</i>
<ul style="list-style-type: none"> • Distribución de los grupos de trabajo. • Docente atento al trabajo de los alumnos. • Plan de actividades del docente • Diario de observación registrar algunos datos observados, como los propuestos por Daniela en [60]
<i>Resultados esperados:</i>
Diario de observación que permita identificar problemas, soluciones y el impacto de las observaciones en el aprendizaje.

Tabla 10. Observación.

3.4 GUÍA DE CHILDPROGRAMMING CON LA EXTENSIÓN PENSAMIENTO COMPUTACIONAL Y ROBÓTICA EDUCATIVA.

La guía de ChildProgramming con la extensión Pensamiento Computacional y Robótica Educativa tiene por objetivo servir como orientación para aquellas personas interesadas en trabajar en el desarrollo del Pensamiento Computacional de niños entre los 10 y los 12 años de edad, empleando para ello como herramienta la Robótica Educativa. Se propone para tal efecto la realización de un curso básico de robótica, dividido en mínimo 5 sesiones de clase con una intensidad horaria de 3 a 4 horas. Cada sesión de entrenamiento está diseñada para ejecutarse en las siguientes etapas, acorde con el ciclo de vida de ChildProgramming (figura 11), el cual se basa en la analogía de una ronda de juego.

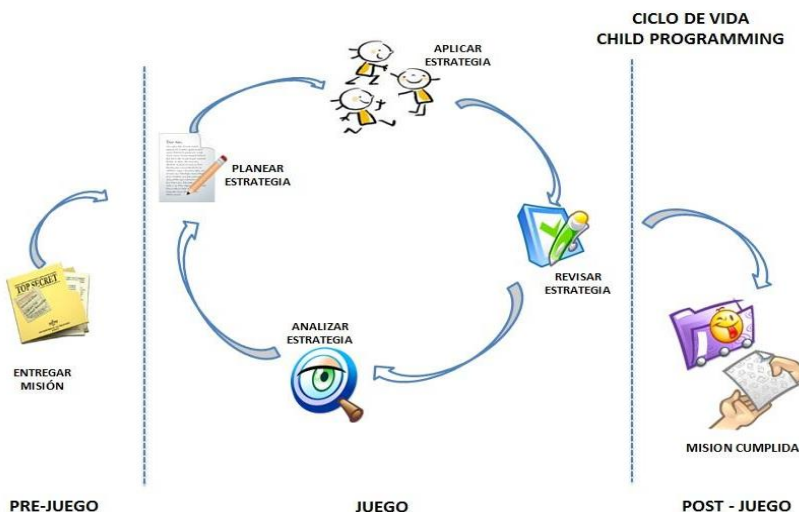


Figura 11. Ciclo de vida de ChildProgramming. Tomado de [8]

Requisitos mínimos para cada sesión:

En la figura 12 se describen los requisitos para cumplir la misión propuesta en cada una de las sesiones de entrenamiento.

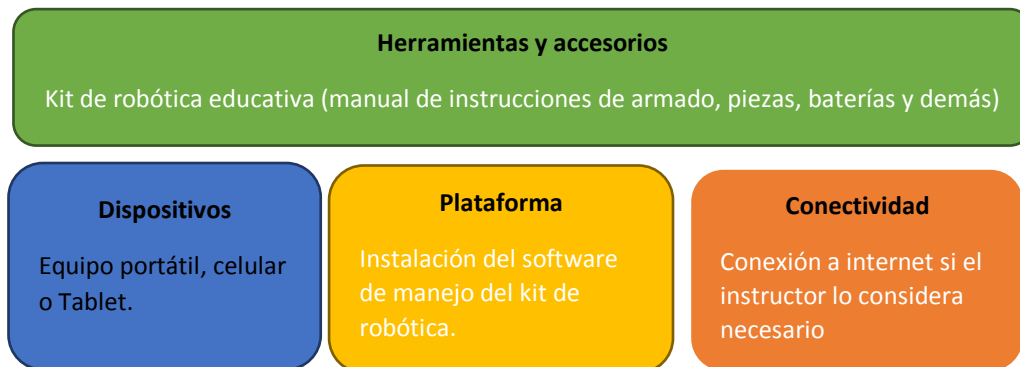


Figura 12. Requisitos mínimos de una sesión de entrenamiento en robótica educativa. Adaptado de: RobotLab (<http://www.bnm.me.gov.ar/>)

Estructura de una sesión de robótica:

Se propone una adaptación del modelo de ChildProgramming para obtener la estructura de una sesión de Robótica Educativa. Con anterioridad a las etapas de ChildProgramming, es necesario al inicio dar una charla o actividad teórica en la que el docente o instructor defina los conceptos a aplicar en la misión propuesta; una vez cumplida esta estación, se suceden el pre-juego, juego, post-juego y las consideraciones finales. Entonces, cada sesión de entrenamiento se ha dividido en:

1. **Inicio:** En las actividades iniciales se alimenta el curso con fundamentos teóricos del tema a desarrollar, que también pueden incluir recomendaciones o pautas a seguir durante el desarrollo de la misión.
2. **Pre-juego:** Se establecen y entregan a los equipos los requerimientos de la misión a cumplir.
3. **Juego:** Cada equipo analiza los requerimientos, crea y desarrolla una solución que satisfaga la misión encomendada. El docente monitorea el comportamiento de los equipos.
4. **Post-Juego:** Los equipos entregan la misión, tras lo cual el docente se encarga de verificar y evaluar el cumplimiento de los requerimientos. Se entregan estímulos o reconocimientos a los equipos.
5. **Consideraciones finales:** En esta etapa se hace una retroalimentación tanto de la fundamentación teórica como de los resultados obtenidos en el desarrollo de la misión.

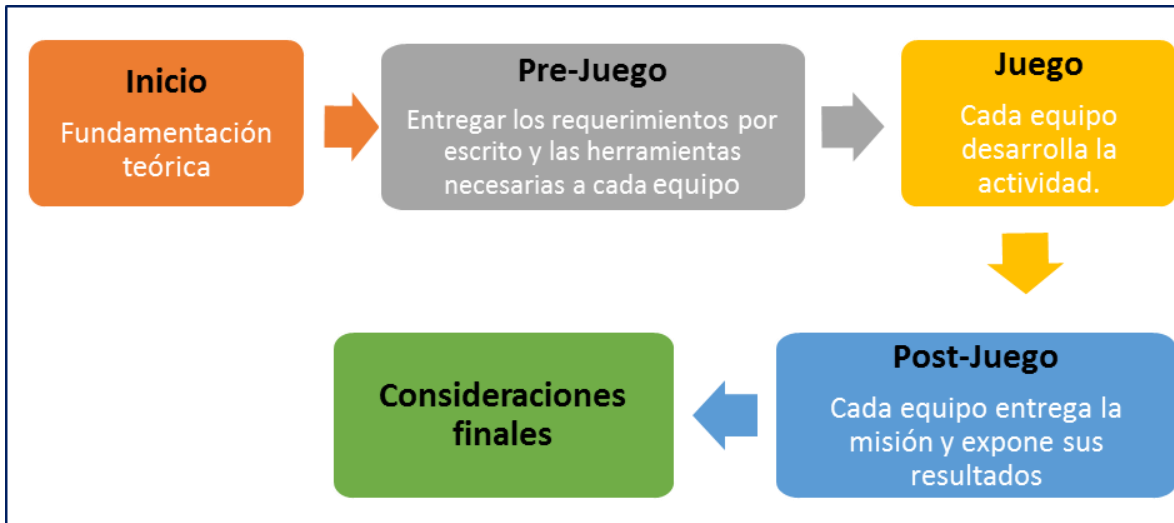


Figura 13. Estructura por etapas de una sesión de robótica.

La figura 13 ilustra un diagrama en bloques de las etapas de una sesión de robótica, considerando el resto de paquetes metodológicos.

Primera Sesión de entrenamiento:

La primera sesión de entrenamiento tiene por finalidad conocer los grupos de trabajo, darles información general acerca de cómo se va a desarrollar el curso, cuáles son sus alcances y cuáles son las reglas de trabajo. Es importante delimitar los alcances del curso para evitar crear falsas expectativas; para dar un ejemplo, cuando a los niños se les habla de robots pueden llegar a imaginarse aparatos muy sofisticados que tal vez obedezcan desde el principio comandos gestuales o de voz. En esta sesión también se hace una introducción de los conceptos de Pensamiento Computacional a aplicaciones de la vida cotidiana:

1. Como primera medida es necesario establecer algunas reglas, entre las que están:
 - Tiempos de cada actividad.
 - Forma de asignar los grupos de trabajo.
 - El objetivo de la sesión.
 - Manejo de los dispositivos.
 - Lista de las herramientas o partes físicas que se necesitan para armar el robot o para ponerlo en funcionamiento; por ejemplo, tornillos, motores, sensores, unidades de control, ruedas, baterías y cables de conexión, entre otros.
2. Realizar un Pre-test para conocer la experiencia previa de los estudiantes con respecto a los conceptos a desarrollar durante el transcurso del curso, con el fin de mejorar la planeación del mismo y mantener la motivación de los estudiantes.
3. Con el fin de dar una orientación general, es recomendable dar una charla o dinámica teórica, que debe ser breve para no generar cansancio y pérdida de interés por la actividad central. En esta orientación se desarrollan los conceptos de:
 - Pensamiento computacional
 - Abstracción, descomposición y diseño de algoritmos.
 - Robótica Educativa:
 - ¿Qué es un robot?, sensores, actuadores y unidad de control.
4. Pre-juego:

En el Pre-juego se entrega la misión a cada grupo y no más que los elementos necesarios para desarrollarla. Es importante lograr el acercamiento del niño con el robot, por lo que se sugiere que la primera misión sea armar el robot. Cuando el kit robótico admite múltiples formas de ser ensamblado, se recomienda escoger una forma básica, dado que es la primera vez que se trabaja con él. Al final de la actividad el niño podrá jugar con su robot, y conocerlo mejor.

5. Juego:

Durante el desarrollo del juego, el instructor o docente debe estar pendiente de detalles como la forma de ensamblar las piezas, revisar que se siga el manual de instrucciones y fomentar el trabajo en equipo. Idealmente, cada miembro del equipo debería armar una parte del robot y se debería hacer una rotación de roles, con el fin de dinamizar la sesión y de permitir a cada estudiante ser partícipe activo de la creación del equipo. Si se presentaran inconvenientes durante el ensamblaje del robot, el docente deberá actuar como un facilitador pero, en ningún caso, debe participar como un miembro más, permitiendo que cada equipo resuelva sus propios problemas, mas sin dejar que el grupo llegue al punto de la frustración.

Una vez armado completamente el robot, para los niños es motivador poder darle un nombre y personalizarlo con adornos, dibujos de expresiones, en fin, detalles característicos. Ésta es una actividad que disfrutan mucho y mejora su estado de ánimo sobre todo cuando se presentan inconvenientes a la hora de ensamblar el robot.

6. Post-Juego

Cada grupo debe mostrar su robot y, junto con el software de programación respectivo, ir descubriendo las funciones, reconocer sus partes y divertirse.

7. Consideraciones finales

Aquí se realiza una retroalimentación para verificar el cumplimiento de los objetivos de la misión y su relación con la fundamentación teórica. Si el docente lo considera necesario, puede aplicar un Pos-test que le permita evaluar los conceptos aprendidos y recopilar algunas dudas o sugerencias que pudieran surgir sobre la misión.

Segunda sesión de entrenamiento:

El objetivo de esta sesión es empezar a integrar de forma práctica el desarrollo del Pensamiento Computacional con RE, mediante la programación del robot. A partir de esta segunda sesión de entrenamiento, los alumnos deben crear los algoritmos de programación para cumplir las misiones y entregar los códigos de la programación en bloques.

1. Inicio:

Reforzar los conceptos vistos en la primera sesión y añadir algunos nuevos como reconocimiento de patrones.

Dar una breve descripción del entorno de programación, la creación de los bloques de programación, la forma de conectar el robot para transferir a él las instrucciones desde el computador y la comunicación de aquél con el medio ambiente.

2. Pre-Juego:

Entregar la misión en forma escrita a cada grupo, con los objetivos bien definidos. El robot armado por cada grupo en la primera misión es el mismo a usar en la segunda.

3. Juego:

Durante el diseño del algoritmo y la programación, el instructor o docente debe hacer la analogía entre los conceptos revisados en la teoría y su aplicación en el desarrollo de la misión.

Se sugiere realizar actividades tales como encender leds, hacer movimientos coordinados (adelante, atrás, derecha, izquierda y en círculos) y, en general, trabajar con los actuadores (motores, luces, sonido, etc.)

4. Post-Juego

Cada equipo entrega su misión con el algoritmo correspondiente y hace la exposición de su robot.

Tercera, cuarta y quinta sesiones de entrenamiento:

Desde la tercera sesión de entrenamiento, el objetivo es común: desarrollar una solución software que le permita al robot realizar la misión propuesta. Se busca, en cada una de estas sesiones, mejorar el planteamiento de las estrategias de solución, aprender a programar cada una de las partes del robot (esencialmente actuadores y sensores) y corregir los errores que se puedan presentar.

1. Inicio:

De la tercera sesión en adelante se puede seguir con el mismo modelo, haciendo énfasis en cada sesión en los conceptos teóricos ya repasados y también en necesarios para programar del robot y/o alcanzar con éxito el objetivo propuesto.

2. Pre-Juego:

Entregar la misión al grupo, con los objetivos bien definidos. En las sesiones de la tercera en adelante, es conveniente probar los sensores y combinarlos con los diferentes actuadores.

Las misiones a desarrollar pueden estar enfocadas en probar los sensores de temperatura, de luz, el seguidor de líneas, etc., y los diferentes tipos de actuadores. Siempre en cada misión deben pedirse los algoritmos de diseño.

3. Juego:

Cada equipo debe contar con sus propias herramientas de trabajo. El robot a programar puede ser el mismo que se armaron en la primera sesión, pero puede cambiarse o armarse uno nuevo se cuenta con el tiempo es suficiente.

4. Post-Juego

Cada equipo entrega su misión con el algoritmo correspondiente y hace la exposición de su robot.

Para la última sesión es conveniente aplicar un Post -test para evaluar la evolución del aprendizaje y la motivación de los aprendices con respecto al curso.

En la tabla 11 se hace un resumen de las actividades sugeridas para el desarrollo del pensamiento computacional en el contexto de la robótica educativa, donde se especifican el propósito, la descripción y las actividades en cada sesión de entrenamiento.

Sesión	Propósito	Descripción	Actividades
I	Entender la percepción previa de los estudiantes sobre los conceptos de robótica y pensamiento computacional.	Recopilación de información de los mismos estudiantes, para tener una idea de los conocimientos previos que ellos tienen sobre robótica y pensamiento computacional.	Pre-test.
	Conocer los conceptos de Robótica y Pensamiento Computacional.	Primera aproximación, en forma de explicación, a los conceptos de Robótica y de Pensamiento Computacional.	Identificación de las características de los robots. Identificación del pensamiento computacional y de sus conceptos asociados.
	Familiarizar a los estudiantes con el robot.	Contacto de los estudiantes con el robot. Los estudiantes ensamblan el robot y juegan con él, al tiempo que se refuerzan en ellos los conceptos de robótica.	Ensamblaje y juego con mBot.
II	Entender qué son los actuadores y cuál es su función.	Explicación de los diferentes tipos de actuadores en el robot, enseñándoles a los estudiantes con una breve explicación de funcionamiento y dando ejemplos de situaciones en las que podrían utilizarse.	Identificación de los actuadores en el mBot.
	Entender la importancia de la abstracción y de los algoritmos, y la construcción de estos últimos.	Se explica mediante ejemplos a los estudiantes que la abstracción y los algoritmos se realizan muchas veces en la vida diaria, de manera inconsciente, y que es conveniente aplicarlo de forma consciente y metodológica a muchos procesos. Se les pide desarrollar algoritmos para tareas del colegio o actividades que ellos mismos propongan, haciendo hincapié en la existencia de más de una solución para un mismo problema.	Abstracción y algoritmos en la cotidianidad.
	Conocer el entorno de programación mBlock.	Primera aproximación al entorno de programación y desarrollo de un programa sencillo que demuestre cómo las acciones programadas se reflejan en el funcionamiento del robot, haciendo énfasis en los actuadores (movimiento, luces y sonido). Se sugiere combinar acciones obligatorias, que permitan verificar que el robot hace lo que los estudiantes ordenan, con acciones libres que les permitan expresarse libremente y hacer uso de su creatividad.	Abstracción y diseño de algoritmos como paso previo a la codificación. Construcción de bloques de código que involucren actuadores.
III	Entender los conceptos de descomposición y reconocimiento de patrones.	Se explica mediante ejemplos a los estudiantes en qué consiste la descomposición y el reconocimiento de patrones y su utilidad para resolver problemas al reducir su complejidad, o al abordar problemas total o parcialmente similares. Se les pide descomponer problemas comunes	Descomposición y reconocimiento de patrones en tareas complejas.

		propuestos por ellos mismos y/o por el profesor.	
	Aprender a manejar el sensor infrarrojo como seguidor de línea.	Explicación del sensor infrarrojo del mBot, y su funcionamiento como seguidor de línea, junto con los comandos de mBlock que permiten conocer la información que entrega (su estado).	Verificación en mBlock del estado del sensor infrarrojo del mBot.
	Aprender a utilizar operadores booleanos y las estructuras de control.	Creación de programas sencillos que permitan la inclusión de operadores booleanos y de las sentencias de mBlock “si... entonces... sino...” y “repite...” junto con el uso del sensor infrarrojo. Antes de empezar con la codificación es necesario que los estudiantes presenten y sustenten la descomposición del problema y un algoritmo de solución.	Descomposición del problema. Algoritmo de solución. Seguidor de línea con el mBot.
IV	Evaluar el conocimiento acerca de los conceptos de robótica y pensamiento computacional	Recuento del significado de pensamiento computacional y los conceptos ya trabajados de abstracción, algoritmos, descomposición y reconocimiento de patrones.	Quiz.
	Aprender a manejar el sensor infrarrojo como seguidor de línea.	Explicación del sensor ultrasónico del mBot, y su funcionamiento como detector de obstáculos y como medidor de distancia, junto con los comandos de mBlock que permiten conocer la información que entrega.	Verificación en mBlock del estado del sensor ultrasónico del mBot.
	Poner en práctica el pensamiento computacional y la programación de robots para solucionar un problema particular.	Dado un problema planteado por el profesor, los estudiantes deberán, bajo su guía, ayuda y supervisión, aplicar los conceptos de pensamiento computacional y programación de robots para solucionarlo.	Solución de problemas con la aplicación de pensamiento computacional a la robótica.
V	Poner en práctica el pensamiento computacional y la programación de robots para solucionar en forma autónoma.	Dado un problema planteado por el profesor, los estudiantes deberán, con ayuda mínima, aplicar los conceptos de pensamiento computacional y programación de robots para solucionarlo.	Reto final y PostTest

Tabla 11. Actividades para el desarrollo del Pensamiento Computacional en el contexto de la Robótica Educativa.

3.5 Lecciones aprendidas

- Al iniciar el curso de robótica es importante que el docente o instructor haga una planeación de cada una de las sesiones, realizar las misiones antes de proponerlas a los alumnos, con el fin de: prever posibles fallas de las partes físicas, errores de

programación y saber cuáles son los elementos necesarios para el cumplimiento de la misión.

- Para lograr una integración del desarrollo del pensamiento computacional con la robótica educativa, es importante hacer un recuento teórico de los conceptos del pensamiento computacional, para que al estudiante le sea fácil relacionar la actividad práctica con su componente teórico.
- El robot, en sí mismo, constituye una distracción. Se recomienda permitir que los equipos tengan acceso a él solamente cuando hayan presentado y expuesto un análisis (que requiere abstracción y descomposición) y algoritmo aceptables. Para disminuir la distracción se propone:
 1. Salvo en la primera sesión, con el fin de alejarse de distractores, debe evitarse el uso de pantallas diferentes a la(s) empleada(s) para la programación del dispositivo, excluyendo por supuesto aquellas requeridas por el (los) instructor(es) para el desarrollo de la actividad.
 2. El espacio de trabajo debe ser lo suficientemente amplio, con un área dedicada a probar el robot, en la que es muy probable que dos o más equipos coincidan al mismo tiempo.
- Al crear o diseñar las misiones a cumplir en cada sesión de entrenamiento, es recomendable incluir aquellas que involucren el ensamblar o armar el robot, debido a que es una de las actividades que más motiva a los alumnos y, de ser posible, dar espacio a que lo personalicen (le pongan un nombre y adornos a su gusto, por ejemplo).
- En cuanto al costo, el mBot Ranger está situado en un punto medio en comparación con otros kits similares. Es un robot con una unidad de control basada en Arduino, programable con mBlock (basado en Scratch) mediante conexión Bluetooth. El mBot Ranger cuenta con un sensor de seguimiento de línea, sensores de luz, giroscopio y actuadores que incluyen motores, luces led y una bocina básica capaz de reproducir tonos de variada frecuencia.

4. PLANEACIÓN, EJECUCIÓN Y RESULTADOS DEL ESTUDIO

En este capítulo se describen, en ese orden, la planeación, el diseño, la ejecución y el análisis del estudio realizado en el desarrollo del proyecto de investigación.

4.1 PLANEACIÓN DEL ESTUDIO

En la planeación del estudio se tuvo en cuenta el objetivo de estudio y, a continuación, se plantearon tres hipótesis.

Objetivo del Estudio: Analizar el efecto de la inclusión de elementos de la robótica educativa en el modelo ChildProgramming con el propósito de evaluar su efecto con respecto a la motivación, participación y la apropiación de los conceptos del CT desde un punto de vista de los investigadores en el contexto de un curso de tecnología con alumnos que se encuentran en las edades de 10 a 12 años.

Hipótesis 1: Al incluir la robótica educativa en el contexto de ChildProgramming como herramienta del desarrollo del pensamiento computacional, se produce un incremento significativo en la motivación en el desarrollo CT en los niños que participan en el experimento.

Hipótesis 2: Al incluir la robótica educativa en el contexto de ChildProgramming como herramienta del desarrollo del pensamiento computacional, se produce un incremento significativo en la participación en el desarrollo CT en los niños que participan en el experimento.

Hipótesis 3: Al incluir la robótica educativa en el contexto de ChildProgramming como herramienta del desarrollo del pensamiento computacional, se produce un incremento significativo en la apropiación de los conceptos de CT en los niños que participan en el experimento.

4.1.1 Metodología de Estudio

Para su desarrollo, este estudio se condujo en 3 etapas de acuerdo con las directrices del ciclo de investigación científica definida por Hurtado [59], el cual integra lineamientos generales de Bunge [60] y particulares de Shaw [61]. Las fases principales del estudio son tres: exploración, formulación y evaluación. En la exploración se realizó una revisión de la literatura y se diseñó y ejecutó el estudio exploratorio; en la fase de formulación se diseñó y construyó la extensión; el diseño y ejecución del experimento confirmatorio corresponden a la fase de evaluación.

4.1.1.1 Exploración

El propósito de esta actividad es explorar el contexto, las herramientas y posibles metodologías de implementación de ER con CT. Esto incluye consultar desde fuentes relevantes, el uso de robótica educativa para el desarrollo del pensamiento computacional,

incluyendo una exploración de las herramientas software y hardware candidatas para proyecto (equipo de cómputo, lenguajes de programación, kits robóticos, etc.). Para esto se realizó una revisión de la literatura siguiendo o adaptando la guía planteada por Turner et al. [37], es el contenido del capítulo 2 de la presente monografía. Además, se diseñó y ejecutó de un experimento preliminar para lo cual se siguió la guía de Wohlin et al. [62]. En párrafos posteriores se explica el diseño del estudio.

4.1.1.2 Construcción de la Extensión

En esta actividad se crea la extensión de Childprogramming como un componente de método, siguiendo el enfoque de ingeniería de método situacional propuesto por Ralyté [63] lo cual incluye capturar los requerimientos del método, seleccionar componentes de método y ensamblar los componentes del método en el contexto de una extensión a ChildProgramming. Esta actividad está desarrollada en el capítulo 3 del presente documento.

4.1.1.3 Evaluación

El objetivo de esta actividad es evaluar el efecto de la extensión (en términos de la asimilación de los conceptos del CT) a través del diseño y ejecución de un experimento controlado, siguiendo la guía de Wohlin et al. [62]. Se encuentra desarrollada en el apartado experimento confirmatorio.

4.1.2 Variables

Para probar la validez de las hipótesis planteadas en la planeación del estudio, se manejan dos tipos de variables: dependientes e independientes. Las variables independientes son aquellas que se pueden controlar por parte de los investigadores y las dependientes son las que se pretende medir con los datos recogidos en la ejecución del estudio. Para tal efecto, se definieron las siguientes:

Variables independientes: Actividades: Actividad ChildProgramming con RE y ChildProgramming sin RE.

Variables dependientes: Motivación, participación y apropiación del CT.

4.2 DISEÑO DEL ESTUDIO

El diseño del estudio comprende los siguientes aspectos: selección de los participantes, el contexto, la pregunta de investigación y el objetivo del estudio.

4.2.1 Selección de los participantes del estudio

La población se seleccionó considerando que las edades y el desarrollo físico y mental estuvieran dentro de los valores típicos y además considerando la disponibilidad y autorización de los individuos (y sus padres) para fines experimentales. Se realizaron dos

experimentos: exploratorio y confirmatorio. En el experimento exploratorio se trabajó con un grupo de estudiantes con el modelo de ChildProgramming con RE; en el experimento confirmatorio participaron dos grupos, uno se trabaja con el modelo original de ChildProgramming y el otro con la extensión de ChildProgramming con RE.

4.2.2 Contexto del Estudio

La ejecución del estudio exploratorio se realizó en las instalaciones del Colegio CESE – Centro de Soluciones Educativas, ubicado en la comuna 1 de la ciudad de Popayán (Colombia) en el barrio Prados del Norte, Carrera 7 # 12N-16. En el momento de la investigación Colombia atravesaba por un estado de emergencia sanitaria debido a la pandemia generada por el Covid-19, por esta razón las instalaciones del Colegio estaban desocupadas y fue posible adecuar una zona segura para el trabajo con los niños.

Debido al estado de emergencia sanitaria decretado con ocasión de la pandemia de COVID-19, se trabajó con un grupo de 6 niños con edades entre 10 y 12 años (figura 14), quienes accedieron a reunirse, contando con el respectivo permiso expreso de sus padres o acudientes. Dado que entre los niños compartían diariamente juegos, algunos son compañeros de clase y viven en casas vecinas, estas características permitieron desarrollar sesiones presenciales.



Figura 14. Participantes del estudio exploratorio.

4.2.3 Pregunta de investigación

Al incluir la robótica educativa en el contexto de ChildProgramming como herramienta del desarrollo del pensamiento computacional, ¿se produce un incremento significativo en la motivación en el desarrollo CT?

4.2.4 Objetivo del estudio:

Introducir la robótica educativa en el modelo de ChildProgramming, para lograr incrementar la motivación de los niños en las actividades propuestas durante la ejecución del experimento.

4.3 EJECUCIÓN DEL ESTUDIO

La ejecución del estudio tuvo dos fases; en la primera se desarrolló un estudio exploratorio y en la segunda un experimento confirmatorio.

4.3.1 Estudio Exploratorio

En el estudio exploratorio se pretende indagar cuál es el impacto de incluir la robótica educativa en el desarrollo de habilidades del pensamiento computacional. En el desarrollo del estudio exploratorio se tienen en cuenta las siguientes características: selección del estudio, los sujetos de investigación, ejecución, instrumentos de medición, resultados cuantitativos y análisis de resultados del estudio exploratorio.

4.3.1.1 Selección del estudio:

Dado que las actividades principales de este estudio van orientadas al trabajo específicamente con niños, en este caso la unidad de análisis son los equipos de trabajo. El estudio es de tipo exploratorio con el fin de indagar y recoger información necesaria para proponer una extensión de ChildProgramming.

4.3.1.2 Los sujetos de investigación

Como unidades de análisis en el estudio, se tienen los dos grupos de trabajo compuestos por niños con edades de 10 a 12 años. El nivel educativo de los participantes es de cuarto y quinto grado de educación básica primaria, quienes conforman las fuentes primarias de información. El estudio es exploratorio, con el objetivo de incluir la robótica educativa en el modelo de ChildProgramming.

Para ejecutar el estudio exploratorio, se desarrolló un curso con el nombre de “Scratch + Arduino: Introducción a la Robótica”, con una intensidad de 16 horas, distribuidas en cuatro sesiones de entrenamiento de 4 horas cada una.

4.3.1.3 Ejecución del estudio exploratorio

Este estudio, en forma de curso corto, se llevó a cabo en el período comprendido entre el 1 y el 21 de diciembre de 2021. A lo largo del curso, los participantes, la distribución de los equipos y el ambiente de trabajo se mantuvieron sin cambios, tal y como se indica en la tabla 12.

Curso “Scratch + Arduino: Introducción a la Robótica”	
Participantes:	<ul style="list-style-type: none"> Se trabajó con un grupo de 6 niños, de entre los 10 y 12 años de edad. En cuanto a género, tomaron parte del curso 4 niños y 2 niñas, quienes contaron con la autorización expresa y escrita por parte de su(s) padre(s) para tomar parte en el experimento, para lo cual previamente se les informó sobre los fines del curso y el tipo de actividades a desarrollar. Investigadores / Observadores.
Conformación de los Equipos:	<p>Se conformaron dos grupos de tres niños cada uno. Los grupos se formaron en forma voluntaria, sin intervención de los observadores. Así, resultaron un grupo de tres niños y otro de un niño y dos niñas.</p> <p>Ésta y todas las sesiones del curso tuvieron lugar en uno de los salones de clase del Colegio CESE (Centro de Soluciones Educativas), que fue cedido especialmente para este propósito.</p> <p>El salón contó con un televisor con pantalla táctil, al cual tanto los observadores como los niños tuvieron acceso y que fue muy útil para el desarrollo de las actividades, pues en él se mostraba información relevante para el curso, como presentaciones de diapositivas e instrucciones para el montaje del robot.</p>
Ambiente de Trabajo:	<p>En ese salón se dispusieron tres mesas, una para cada equipo y otra para los dos observadores, además de una silla por cada una de las ocho personas en el salón. Las mesas se ubicaron con suficiente espaciamiento entre sí, brindando a los participantes suficiente espacio para trabajar sin interferirse entre sí, para poder hacer las observaciones con cierta comodidad, para tener un espacio de pruebas accesible para ambos equipos y, muy especialmente, atendiendo las recomendaciones de distanciamiento social con razón de la pandemia en curso en el momento del experimento.</p> <p>Por esta última razón también se hizo obligatorio ejecutar una desinfección del salón y del mobiliario antes de comenzar ésta y cada una de las sesiones, así como exigir de los participantes el cumplimiento de las normas de bioseguridad, como lavado y desinfección de manos, desinfección de calzado y uso de tapabocas.</p>

Tabla 12. Curso “Scratch + Arduino: Introducción a la Robótica”

El curso se dividió en cuatro sesiones de entrenamiento y, en cada una, se propuso una misión para ser ejecutada por los grupos de trabajo. En las tablas 13 a 16 se describen cada una de las sesiones realizadas durante el desarrollo del experimento exploratorio “Scratch + Arduino: Introducción a la Robótica”, junto con sus correspondientes actividades.

Sesión 1 de entrenamiento	
Nombre de la Actividad:	Ensamblaje del robot.
Descripción:	Esta actividad se dividió en dos partes. En la primera, los estudiantes, trabajando por equipos, ensamblaron el robot mBot, en su variante conocida como mBot Ranger, siguiendo las instrucciones suministradas por el

	<p>fabricante. En la segunda parte, los equipos probaron sus construcciones robóticas, dirigiéndolas por medio de la aplicación móvil mBlock, también desarrollada por el fabricante.</p>
<p>Materiales:</p>	<p>El salón contó con un televisor con pantalla táctil, al cual tanto los observadores como los niños tuvieron acceso y que fue muy útil para el desarrollo de las actividades, pues en él se mostraba información relevante para el curso, como presentaciones de diapositivas e instrucciones para el montaje del robot.</p> <p>A cada equipo se le entregó el material de trabajo respectivo el cual incluía:</p> <ul style="list-style-type: none"> • 1 computador portátil por mesa de trabajo, cada uno con su mouse. • 1 Hoja de papel por participante • 1 Lápiz por participante • 1 Borrador • 1 Cuestionario (Pretest) por participante • Kit robótico mBot, necesario para el mBot Ranger, junto con sus respectivas instrucciones de ensamblaje en versión impresa y en PDF. • Dispositivo móvil, necesario para dar instrucciones al mBot una vez ensamblado. Se utilizaron dispositivos basados en Android. <p>La metodología empleada para esta actividad se describe a continuación:</p>
<p>Metodología:</p>	<ol style="list-style-type: none"> 1. Asignar a cada equipo en su mesa de trabajo, la que ocuparán a lo largo del curso. 2. Entregar el pretest a todos los participantes, junto con lápiz y borrador, y pedirles que lo respondan, aclarando que el fin que persigue es investigativo y que no tiene incidencia en su continuación en el curso. 3. Explicar a los participantes la actividad a realizar. 4. Entregar a cada grupo el computador, el kit robótico junto con las instrucciones de armado, tanto en versión impresa, como en PDF. 5. En este punto, los equipos comienzan el ensamblaje del robot, siguiendo las instrucciones del manual. Se les pidió que asignaran roles entre los miembros de cada equipo. Así, uno o dos de ellos se encargaban de interpretar las instrucciones y ensamblar las piezas, mientras que otro verificaba al tiempo su correcta ejecución, de acuerdo con el manual. Estos roles eran intercambiados a medida que la actividad avanzaba. 6. Una vez el robot estaba ensamblado, se hizo entrega a cada equipo del dispositivo móvil que contenía la aplicación para controlarlo y se les dio una muy breve explicación de su manejo. 7. Los participantes de cada equipo se turnaron para jugar con su robot, manipulándolo por medio de la aplicación.
<p>Entregables de la Actividad:</p>	<p>Como resultado de esta actividad los equipos de trabajo debían entregar los siguientes productos entregables:</p> <ul style="list-style-type: none"> • Pretest. • Robot mBot Ranger ensamblado y funcionando.
<p>Defectos:</p>	<p>Las piezas del robot no se ajustaron correctamente, generando inestabilidad en su estructura.</p>

Tabla 13. Sesión 1 de entrenamiento

Sesión 2 de entrenamiento	
Nombre de la Actividad:	Semáforo

Descripción:

Esta actividad se dividió en dos partes:

1. Explicación teórica sobre pensamiento computacional y robótica:
 - a. Conceptos del Pensamiento Computacional:
 - Definición del PC.
 - Abstracción.
 - Descomposición.
 - Diseño de algoritmos.
 - b. Conceptos de Robótica:
 - ¿Qué es un robot?
 - Unidad de control.
 - Sensores.
 - Actuadores.
2. Desarrollo de la misión: los equipos deben programar el robot que ensamblaron en la primera sesión para que avance o se detenga al tiempo que sus leds se iluminan con los colores del semáforo. El robot debe cumplir con la secuencia:
leds en verde → avanzar durante 3 segundos → leds en rojo → se detiene 3 segundos → leds en amarillo → emite sonido y retrocede durante 3 segundos.

Materiales:

A cada equipo se le entregó su respectivo material de trabajo:

- 1 computador portátil por mesa de trabajo, cada uno con su mouse y con el programa mBlock instalado.
- 1 Hoja de papel por participante
- 1 Lápiz por participante
- 1 Borrador
- Robot mBot Ranger que ensamblaron en la sesión 1.

La metodología empleada para esta actividad se describe a continuación:

Metodología:

1. Presentar a los estudiantes los conceptos de Pensamiento Computacional y de robótica.
2. Entregar papel, lápiz y borrador a cada participante, un computador portátil por equipo y el mismo robot mBot Ranger que ensamblaron en la sesión inmediatamente anterior.
3. Explicar el funcionamiento básico de mBlock como entorno de programación del robot mBot. Mientras tienen el robot en sus manos, se realiza un refuerzo sobre los conceptos de robótica para que se familiaricen con ellos.
4. Explicar a los participantes la actividad a realizar, pidiéndoles que pongan en práctica los conceptos de descomposición y algoritmos.
5. Se les pide a los participantes que asignen entre ellos los roles de analista, desarrollador de algoritmos y programador.
6. En este punto, los equipos comienzan a discutir sobre la forma de resolver el problema planteado.
7. Los equipos debían escribir su versión de un algoritmo que ejecutara las acciones solicitadas.
8. Una vez que contaban con el algoritmo, los equipos procedían con la programación del robot.
9. Los equipos comprueban su solución poniendo en marcha el robot y, con él, la ejecución del programa logrado.
10. Se realizan nuevamente los pasos 6, 7, 8 y 9, ya sea para corregir acciones o efectos no deseados, o para realizar mejoras en el programa.
11. Durante el proceso, los observadores deben prestar especial atención a indicios que puedan mostrar los participantes sobre la aplicación de los conceptos de PC, especialmente abstracción y generalización de patrones.

Entregables de la Actividad:

Como resultado de esta actividad los equipos de trabajo debían entregar los siguientes productos entregables:

- Algoritmo
- Código realizado en mBlock.
- Robot mBot Ranger programado y funcionando.

Defectos:

Tabla 14. Sesión 2 de entrenamiento

Sesión 3 de entrenamiento	
Nombre de la Actividad:	Robot seguidor de línea.
Descripción:	<p>Las dos partes de esta sesión fueron:</p> <ol style="list-style-type: none">1. Refuerzo de los conceptos de robótica y de Pensamiento Computacional. Se les recordó a los participantes los conceptos del PC y de robótica revisados, haciendo énfasis en funcionamiento y programación de los actuadores y del sensor seguidor de línea.2. Programación y puesta en marcha del robot seguidor de línea. Se les pide a los equipos que programen el robot para que se desplace siguiendo una línea negra que describe un óvalo. También deben programar actuadores como leds y bocina para acompañar el movimiento del robot.
Materiales:	<p>A cada equipo se le entregó su respectivo material de trabajo:</p> <ul style="list-style-type: none">• 1 computador portátil por mesa de trabajo, cada uno con su mouse y con el programa mBlock instalado.• 1 Hoja de papel por participante• 1 Lápiz por participante• 1 Borrador• Robot mBot Ranger que ensamblaron en la sesión 1.
Metodología:	<p>La metodología empleada para esta actividad se describe a continuación:</p> <ol style="list-style-type: none">1. Explicar a los participantes la actividad a realizar, recordándoles la importancia de descomponer el problema y planear una solución en forma de algoritmo. Se hace un corto recuento de lo tratado en la última sesión. También se hace una breve introducción al funcionamiento, manejo y programación del sensor seguidor de línea.2. Se les pide a los participantes que asignen entre ellos los roles de analista, desarrollador de algoritmos y programador, intentando no repetir los roles de la sesión pasada.3. En este punto, los equipos comienzan a discutir sobre la forma de resolver el problema planteado.4. Los equipos debían expresar la forma en que descompusieron el problema y escribir su versión de un algoritmo que ejecutara las acciones solicitadas.5. Una vez que contaban con el algoritmo, los equipos procedían con la programación del robot.6. Los equipos comprueban su solución poniendo en marcha el robot y, con él, la ejecución del programa logrado.7. Se realizan nuevamente los pasos 3, 4, 5 y 6, ya sea para corregir acciones o efectos no deseados, o para realizar mejoras en el programa.8. Durante el proceso, los observadores deben prestar especial atención a indicios que puedan mostrar los participantes sobre la aplicación de los

Entregables de la Actividad:	<p>conceptos de PC, especialmente abstracción y generalización de patrones.</p> <p>Como resultado de esta actividad los equipos de trabajo debían entregar los siguientes productos entregables:</p> <ul style="list-style-type: none"> • Algoritmo • Código realizado en mBlock. • Robot mBot Ranger programado y funcionando.
Defectos:	

Tabla 15. Sesión 3 de entrenamiento.

Sesión 4 de entrenamiento	
Nombre de la Actividad:	Papá Noel entrega regalos
Descripción:	<p>Las dos partes de esta sesión fueron:</p> <ol style="list-style-type: none"> 1. Refuerzo de los conceptos de robótica y de Pensamiento Computacional. Se les recordó a los participantes los conceptos del PC y de robótica revisados, haciendo énfasis en funcionamiento y programación de los actuadores y de los sensores seguidor de línea y sensor de intensidad de luz.. 2. Programación y puesta en marcha del robot programado para esta actividad. <p>Se les pide a los equipos que programen el robot para que se desplace por un circuito (demarkado por una línea blanca en fondo negro), cuando detecte una señal lumínica pare y realice sonidos, espere un segundo y continúe su desplazamiento por la línea hasta que encuentre un obstáculo en el camino y pare.</p>
Materiales:	<p>A cada equipo se le entregó su respectivo material de trabajo:</p> <ul style="list-style-type: none"> • 1 computador portátil por mesa de trabajo, cada uno con su mouse y con el programa mBlock instalado. • 1 Hoja de papel por participante • 1 Lápiz por participante • 1 Borrador • Robot mBot Ranger que ensamblaron en la sesión 1
Metodología:	<p>La metodología empleada para esta actividad se describe a continuación:</p> <ol style="list-style-type: none"> 1. Explicar a los participantes la actividad a realizar, recordándoles la importancia de descomponer el problema y planear una solución en forma de algoritmo. Se hace un corto recuento de lo tratado en la última sesión. También se hace una breve introducción al funcionamiento, manejo y programación del sensor seguidor de línea. 2. Se les pide a los participantes que asignen entre ellos los roles de analista, desarrollador de algoritmos y programador, intentando no repetir los roles de la sesión pasada. 3. En este punto, los equipos comienzan a discutir sobre la forma de resolver el problema planteado. 4. Los equipos debían expresar la forma en que descompusieron el problema y escribir su versión de un algoritmo que ejecutara las acciones solicitadas. 5. Una vez que contaban con el algoritmo, los equipos procedían con la programación del robot. 6. Los equipos comprueban su solución poniendo en marcha el robot y, con él, la ejecución del programa logrado.

Entregables de la Actividad:

Defectos:

7. Se realizan nuevamente los pasos 3, 4, 5 y 6, ya sea para corregir acciones o efectos no deseados, o para realizar mejoras en el programa.

Durante el proceso, los observadores deben prestar especial atención a indicios que puedan mostrar los participantes sobre la aplicación de los conceptos de PC, especialmente abstracción, generalización de patrones, diseño de algoritmos y descomposición.

Como resultado de esta actividad los equipos de trabajo debían entregar los siguientes productos entregables:

- Pretest.
- Robot mBot Ranger ensamblado y funcionando.
- El robot se salía del camino demarcado
- Hubo mucha dificultad al programar el sensor de intensidad de luz.

Tabla 16. Sesión 4 de entrenamiento

4.3.1.4 Instrumentos de medición del Estudio Exploratorio

Los resultados del estudio exploratorio se obtuvieron a partir instrumentos y técnicas de recolección. Con el propósito de recolectar la información producida durante el experimento exploratorio, se diseñaron dos instrumentos, Pre-test y Pos-test, y se adoptó el Protocolo de Observación propuesto por Daniela et al. [64]. Además, se utilizó la herramienta Dr. Scratch¹⁸.

El Pre-test (véase el Anexo C) es un cuestionario de 30 preguntas que se aplicó al inicio del curso. De esas 30, 11 corresponden a preguntas tipo Likert con escala de 5 valores sobre motivación y participación (ítems 1,2,3,4,5,6,7,8,18,23,24), y 19 a preguntas abiertas, repartidas así: 7 acerca de las preferencias de los participantes (ítems 9,10,11,19,26,28,29), 6 sobre conocimientos previos de Pensamiento Computacional (ítems 12,13,14,15,16,17,20,21,22,25,27,30) y otras 6 sobre robótica. En la tabla 17 se muestra la distribución de las preguntas en el Pre-test.

Variables	Indicador	Ítem
Motivación	Expresa directamente su motivación	1, 2, 3, 4, 5, 6
Participación	Expresa directamente su participación en equipo	7, 8, 11
Apropiación del CT	Definición de CT	12
	Abstracción	13
	Algoritmos	14

¹⁸ Dr. Scratch. <http://www.drscratch.org/>

	Descomposición	15
	Reconocimiento de patrones	16
	Programación	22

Tabla 17. Distribución de las preguntas según las variables a medir en el Pre-test. Fuente Propia

El Pos-test (véase el Anexo D) es un cuestionario de 27 preguntas, similar al Pre-test, aplicado al final del curso. De esas 27, 8 corresponden a preguntas tipo Likert con escala de 5 valores sobre motivación y participación, y 19 a preguntas abiertas, repartidas así: 7 acerca de las preferencias de los participantes, 6 sobre conocimientos de Pensamiento Computacional y otras 6 sobre robótica. La tabla 18 resume la clasificación de las preguntas en el Pos-test.

Variables	Indicador	Ítem
Motivación	Expresa directamente su motivación	1, 2, 3, 4, 5, 6, 21
Participación	Expresa directamente su participación en equipo	7
	Definición de CT	8
Apropiación del CT	Abstracción	9
	Algoritmos	10
	Descomposición	11
	Reconocimiento de patrones	12
	Programación	16

Tabla 18. Distribución de las preguntas según las variables a medir en el Pos-test. Fuente Propia

El Protocolo de Observación (véase el Anexo E), fue una herramienta utilizada para medir la motivación y el trabajo en grupo de los participantes en el desarrollo de las misiones. El protocolo se sigue con cada estudiante y cubre 12 categorías o campos, a cada uno de los cuales se asigna un puntaje en la escala de 1 a 4, cada uno con su correspondiente significado (0 = “no se puede observar”; 1 = “nivel bajo”; 2 = “se puede observar casi en todas las situaciones”; 3 = “se puede observar durante todos los proyectos”; 4 = “hace más de lo esperado”).

Además de emplear los anteriores instrumentos, se hizo uso de la herramienta Dr. Scratch, para revisar el código de programación en bloques que los estudiantes generaron en las diferentes sesiones. Dr. Scratch es una aplicación web que permite analizar proyectos codificados en Scratch para verificar si han sido programados correctamente, aprender de sus errores y obtener retroalimentación que puede ser usada para mejorar sus destrezas de programación y desarrollar sus habilidades de PC [65].

La tabla 19 muestra el nivel de competencia que evalúa la herramienta Dr. Scratch, al analizar los códigos de la programación en Scratch.

Aspecto del CT	Nivel de Competencia			
	Ninguno (0 puntos)	Básico (1 punto)	En desarrollo (2 puntos)	Competente (3 puntos)
Abstracción y descomposición de problemas	-	Más de un programa y más de un objeto	Definición de bloques	Uso de clones
Paralelismo	-	Dos programas “en bandera verde”	Dos programas en ‘tecla presionada’, dos programas en ‘al presionar’ el mismo objeto	Dos programas en ‘cuando reciba mensaje’, crear clon, dos programas en ‘cuando %s es > %’, dos programas en ‘cuando el escenario cambie a’
Pensamiento Lógico	-	Si	Si - sino	Operaciones lógicas
Sincronización	-	Esperar	Enviar, cuando reciba mensaje, parar todos, parar programas, parar programas del objeto	Esperar hasta, cuando el escenario cambie a, enviar y esperar
Control de flujo	-	Secuencia de bloques	Repetir, por siempre	Repetir hasta
Interactividad con el usuario	-	Bandera verde	Tecla presionada, objeto presionado, preguntar y esperar, bloques de operaciones con ratón	Cuando %s es > %s, vídeo, audio
Representación de la información	-	Modificadores de propiedades de objetos	Operaciones con variables	Operaciones con listas

Tabla 19. Nivel de Competencia para cada aspecto del pensamiento computacional [65]

4.3.1.5 Resultados cuantitativos.

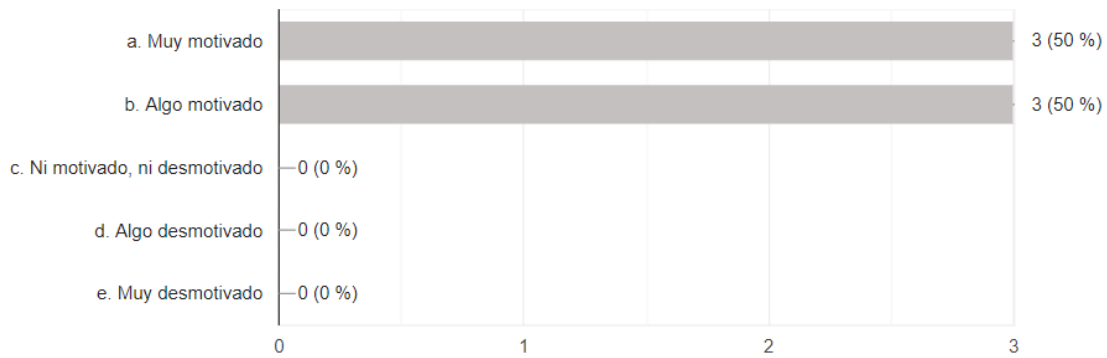
Como resultados cuantitativos, se tomaron unas mediciones directas y se hizo un análisis de resultados. Entre las mediciones directas están las relativas a la motivación, la participación y la apropiación de conceptos de CT

Motivación

Con el fin de medir la variación en la motivación de los participantes entre el inicio y el final del curso, se hizo una comparación entre los resultados del Pre-test y del Pos-test, tomando la pregunta N.1 como la más representativa. En ambos casos, se indaga directamente por la motivación de los estudiantes, obteniéndose los resultados que aparecen en la figura 15.

1. ¿Qué tan motivado te sientes al iniciar este curso?

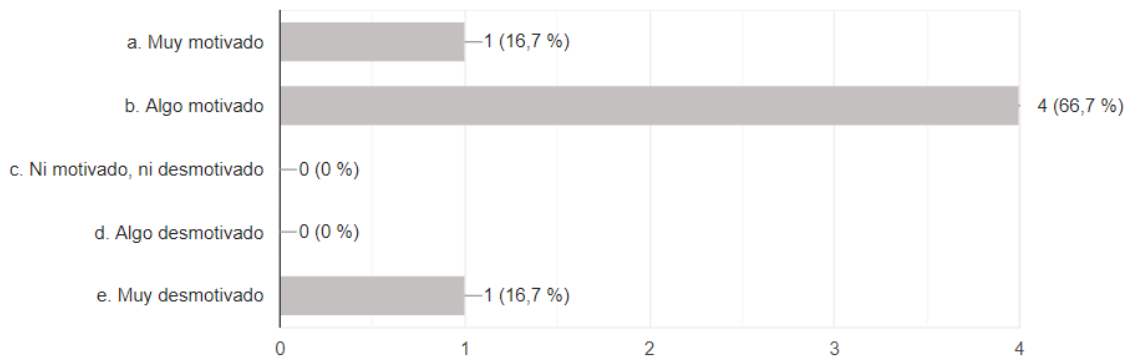
6 respuestas



a) Pre-test.

1. ¿Qué tan motivado te sentiste en este curso?

6 respuestas



b) Pos-test.

Figura 15. Distribución de las respuestas a la pregunta N.1 a) en el Pre-test y b) en el Pos-test.

Comparando los datos de la figura 15, se observa que, entre el inicio y el final del curso, el porcentaje de estudiantes que afirmó estar muy motivado pasó del 50% al 16,7%, el de estudiantes que dijo estar algo motivado pasó del 50% al 66,7% y el de estudiantes que dijo estar muy desmotivado pasó de 0% a 16,7%. Se percibe entonces, en general, una disminución en la motivación de los participantes en el curso.

La aplicación del Protocolo de Observación, en cuanto a los campos relativos a la motivación de los participantes del estudio se refiere, arrojó los resultados que se muestran en la tabla 20.

Criterio de Observación	P. 1	P.2	P.3	P.4	P.5	P.6
Está motivado para superar las dificultades en la realización de tareas.	4	3	3	3	3	2
Está motivado para comprender los errores para corregirlos.	4	4	4	3	3	2
Promedio de los criterios de observación	4	3,5	3,5	3	3	2

Tabla 20. Puntuación de los criterios relacionados con la motivación en el Protocolo de Observación

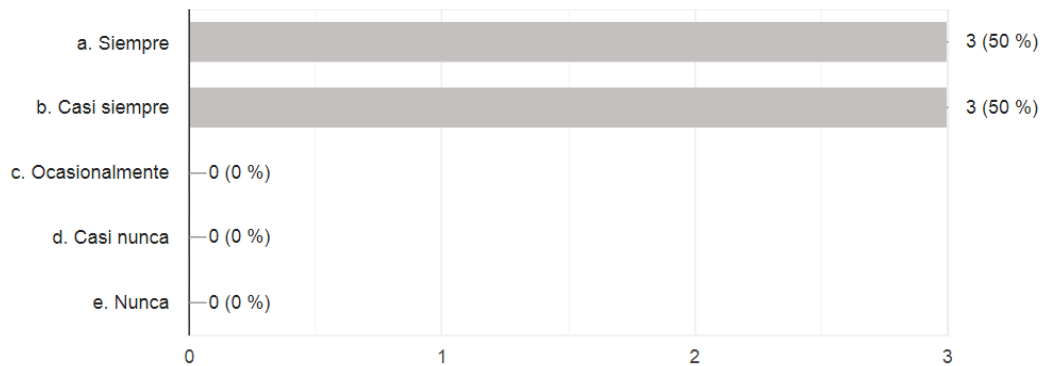
En los resultados de la tabla 20, se muestra que hay un solo participante con un puntaje de 2 en ambos criterios (se puede observar en casi todas las situaciones), los demás participantes tienen un puntaje de 3 o 4 para los dos criterios de observación. El rango de los promedios para los criterios de observación va desde 2 hasta 4.

Participación.

Con el propósito de medir la participación de los niños en la solución de las misiones en grupos se tuvo en cuenta su disposición para trabajar en grupo; por ello, se analizó la pregunta 7 realizada tanto en el Pre-test como en Pos-test.

7. ¿Te gusta trabajar en equipo?

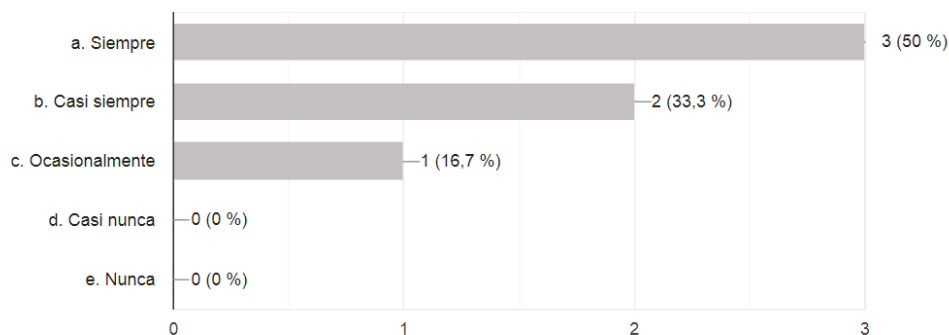
6 respuestas



a) Pre-test

7. ¿Te gustó trabajar en equipo durante el curso?

6 respuestas



b) Pos-test

Figura 16. Distribución de las respuestas a la pregunta N.7 a) en el Pre-test y b) en el Pos-test.

De los resultados obtenidos, y que se muestran en la figura 16, se evidenció el gusto inicial por trabajar en grupo, repartido así por igual entre siempre y casi siempre, cada uno con el 50% de los participantes. Al final del curso, los porcentajes cambiaron, habiendo respondido siempre el mismo 50%, pero pasando casi siempre al 33% y ocasionalmente al 17% de los participantes.

En cuanto a los campos relativos a la participación de los participantes del experimento se refiere, la aplicación del Protocolo de Observación arrojó los resultados que se muestran en la tabla 21.

Criterios de Observación	P1	P2	P3	P4	P5	P6
Coopera con otros miembros del grupo durante las actividades de manera positiva.	4	2	2	3	2	1
Obedece las reglas de comportamiento durante los proyectos.	3	2	2	3	3	3
Participa en trabajos en grupo.	4	4	4	4	3	3
Ayuda a los compañeros a resolver problemas	4	4	4	3	2	1
Promedio de los criterios de observación	3,75	3	3	3,25	2,5	2

Tabla 21. Puntuación de los criterios relacionados con la participación en el Protocolo de observación.

En promedio, según se desprende de la tabla, los alumnos tienen un rango de participación de 2 a 3.75, la clasificación del puntaje para este protocolo se encuentra en el Anexo E

Apropiación del Pensamiento Computacional.

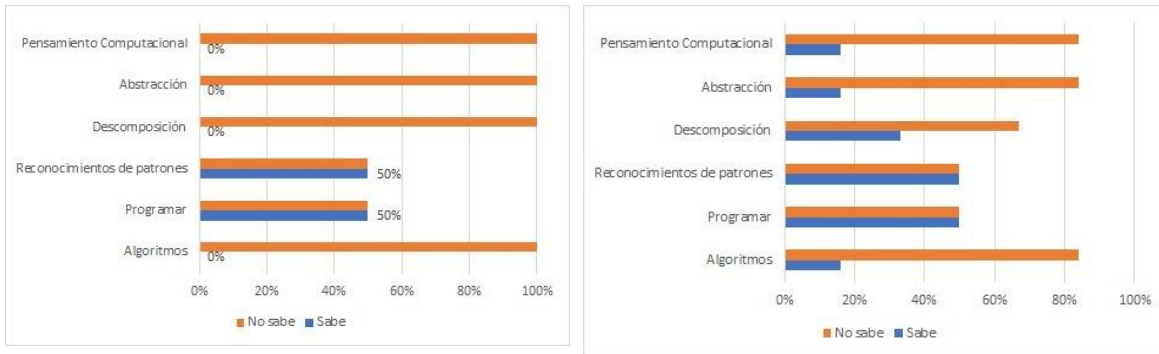
Para medir la apropiación del PC, se tuvieron en cuenta las preguntas relacionadas en las tablas 17 y 18. Dada su naturaleza de preguntas abiertas, eran respondidas por los niños con sus propias palabras. En este punto, se evaluó cada uno de los conceptos de PC teniendo como base las definiciones del modelo Atmatzidou y Demetriadis [22], al que se agregó el concepto de programación y se adaptó la generalización al reconocimiento de

patrones, para obtener el modelo adaptado que se empleó en este proyecto y que se muestra en la tabla 22.

Habilidad de CT	Definición	Guía para su desarrollo
Abstracción	Es el proceso de crear algo simple a partir de algo complicado, omitiendo los detalles irrelevantes, encontrando los patrones relevantes y separando ideas de los detalles tangibles.	<ol style="list-style-type: none"> 1. Separar la información importante de la redundante. 2. Analizar y especificar comportamientos comunes o estructuras de programación entre scripts diferentes. 3. Identificación de abstracciones entre ambientes de programación diferentes.
Generalización (Adaptado a reconocimiento de patrones)	Es transferir un proceso de solución de un problema a una amplia variedad de problemas.	<ol style="list-style-type: none"> 1. Expandir una solución existente en un problema dado para cubrir más posibilidades / casos.
Algoritmos	Es la práctica de escribir, paso a paso, instrucciones específicas y explícitas, para llevar a cabo un proceso.	<ol style="list-style-type: none"> 1. Describir explícitamente los pasos del algoritmo. 2. Posibilidad de diferentes algoritmos para el mismo problema. 3. Esfuerzo para encontrar el algoritmo más efectivo.
Descomposición	Es el proceso de separar los problemas en partes más pequeñas que pueden resolverse con mayor facilidad.	Separar problemas en partes más pequeñas / sencillas que son más fáciles de resolver.
Programación	La programación informática es el arte del proceso por el cual se limpia, codifica, traza y protege el código fuente de programas computacionales, en otras palabras, es indicarle a la computadora lo que tiene que hacer.	Resolver un problema mediante un entorno de desarrollo o en un lenguaje de programación

Tabla 22. Modelo de conceptos de CT utilizados en este proyecto.

En la figura 17 se resume la evaluación de los conocimientos de los conceptos de PC en cada uno de los tests aplicados en el experimento exploratorio, tanto en el Pre-test, como en el Pos-test.



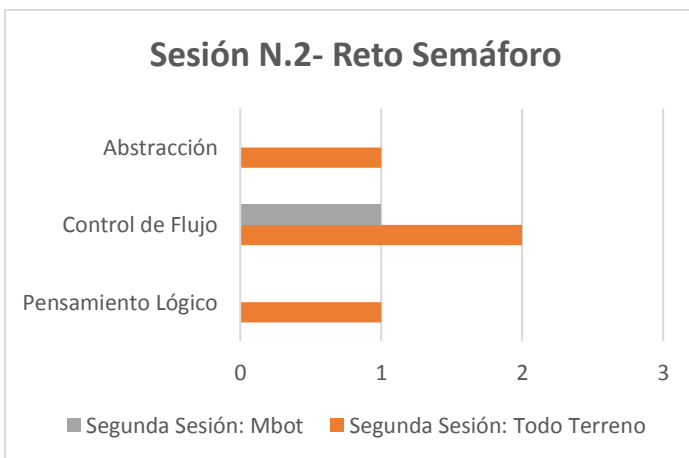
a) Pre-test

b) Pos-test

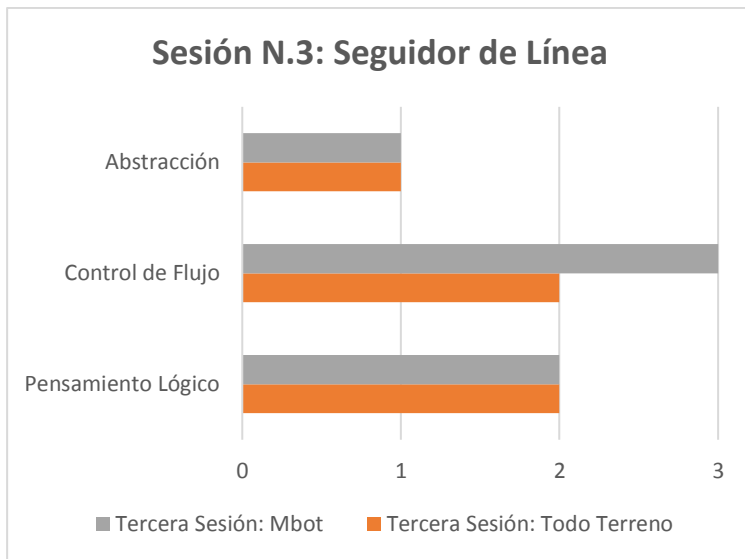
Figura 17. Comparativo a) Pre-test y b) Pos-test sobre la apropiación de los conceptos de PC.

Las gráficas ponen de manifiesto que, al inicio del curso, tan solo la mitad de los niños dijo tener algún conocimiento sobre los conceptos de programación y reconocimiento de patrones, pero la totalidad de ellos desconocía los conceptos de algoritmos, descomposición, abstracción y pensamiento computacional. Los resultados del Pos-test muestran una mejora en la apropiación de los demás conceptos del PC.

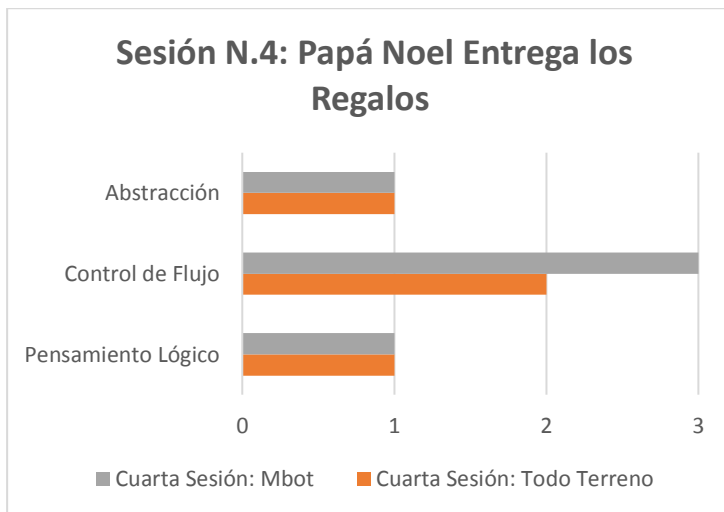
Además de indagar sobre el reconocimiento de los conceptos de PC, se investigó sobre la aplicación de las habilidades que representan. Para ello, se analizó con Dr. Scratch el código entregado por los equipos. Con el doble propósito de motivar a los participantes y de diferenciar los equipos uno de otro, se les pidió elegir un nombre estos; así, el equipo “mBot” quedó conformado por dos niñas y un niño, y el equipo “Todo Terreno” por tres niños. Para medir si además de reconocer los conceptos los alumnos aplican o tienen lagunas de las habilidades del CT, se hizo el análisis de las mediciones de Dr. Scratch. La comparación entre los códigos de los dos equipos, resultado de analizar los respectivos códigos con Dr. Scratch, se muestra en la figura 18.



a) Segunda sesión.



b) Tercera sesión.



c) Cuarta sesión.

Figura 18. Resultados del análisis con Dr. Scratch del código de a) la segunda sesión, b) la tercera sesión y c) la cuarta sesión de entrenamiento.

De acuerdo con los resultados de la figura 18, fueron tres los aspectos del PC evaluados con la herramienta Dr. Scratch: pensamiento lógico, control de flujo y abstracción.

A continuación, se muestran en la tabla 23 las habilidades según la evaluación de la herramienta Dr. Scratch [65].

Habilidad de CT	Criterios de competencia Dr. Scratch	Aplicación al estudio
Abstracción	Abstraer consiste en dividir un problema (o reto, o desafío) grande, en problemas pequeños y de fácil solución.	Mide la abstracción de la solución.

Control de flujo	Orden de ejecución de las instrucciones y uso de operadores de control (uso de operadores condicionales)	Está relacionado con el diseño del algoritmo.
Pensamiento lógico	Permite reconocer como es la calidad de la solución al problema planteado, según su codificación en boques. (uso de operadores, repetir, hasta)	Permite medir el desarrollo del CT

Tabla 23. Habilidades de Pensamiento Computacional, según son descritas vistas por la herramienta Dr. Scratch y su aplicación al estudio exploratorio.

4.3.1.6 Análisis de resultados

En este apartado se tienen en consideración las apreciaciones de los investigadores y las amenazas de validez.

Apreciaciones de los investigadores durante el desarrollo del estudio exploratorio

Durante el desarrollo de las misiones, los participantes manifestaron dificultad para expresar la solución del problema en forma de algoritmo. Aunque fueron capaces de expresar su plan de acción a grandes rasgos, no les fue tan fácil hacerlo como una secuencia ordenada de pasos concretos.

Los participantes demostraban frustración cuando no lograban hacer que las piezas del robot encajaran como les indicaban las instrucciones o cuando el robot no se comportaba como ellos suponían que debía hacerlo, de acuerdo con el código de programación que habían creado.

Aunque había rotación de los roles en los equipos, en algunas ocasiones, alguno o varios de los miembros del equipo se mostraban reacios a hacerlo.

En la primera sesión de entrenamiento se notó que el robot desviaba la atención de los niños durante las charlas teóricas; por tal razón, en sesiones posteriores, fue necesario ocultarlo a la vista mientras se explicaban las actividades o tenía lugar la parte teórica, y sólo se les entregaba el kit robótico cuando era necesario interactuar con él.

Amenazas de validez

El estudio exploratorio se llevó a cabo en diciembre de 2020 y, al igual que el resto del proyecto, durante la pandemia de COVID-19; por este motivo, muchos inconvenientes, incluyendo algunas amenazas de validez, se derivaron de ella, muy especialmente las siguientes:

- Se contó con la participación de muy pocos estudiantes, debido a las severas restricciones impuestas para las reuniones en grupo y a la suspensión de clases presenciales en todo el país. Como consecuencia, debe considerarse que la cantidad de datos recopilada en este experimento no es suficiente como para garantizar que los resultados se reproducirán en un estudio similar.
- Igualmente, se hizo necesaria la implementación de estrictos protocolos de bioseguridad, que incluían desinfección y lavado constante de manos, desinfección

de materiales y ambiente de trabajo, uso permanente de tapabocas y, muy especialmente, artefactos y rutinas que, muy probablemente afectan negativamente el curso. Mención especial amerita la obligación de tener que mantener al mínimo posible el acercamiento y contacto físico entre los investigadores/observadores y los equipos, entre los equipos mismos y hasta entre sus integrantes. Todas estas son variables que no se tuvieron en cuenta en el estudio, pero que sin lugar a dudas pudieron incidir en muchos, sino todos, los resultados. Así, por ejemplo, se dificultaron cosas tan simples como las explicaciones teóricas, sin hablar de las asesorías a los equipos y la interacción entre los propios equipos.

- No todos los participantes asistieron a todas las sesiones y, dado el bajo número de participantes, cualquier ausencia se convierte en un porcentaje representativo que, sin duda, tiene reflejo en los resultados.

Otra de las amenazas de validez está relacionada con la falta de estudios que confirmen la correspondencia propuesta entre los conceptos de PC evaluados por el modelo de la tabla 23 y aquellos analizados por la herramienta Dr. Scratch, lo que escapa del alcance de este proyecto.

4.3.2 Ejecución del experimento confirmatorio

Se tienen en cuenta los siguientes aspectos: contexto de la investigación: pregunta de investigación, objetivo del estudio, selección del estudio, sujetos de estudio, ejecución del experimento confirmatorio, instrumentos y técnicas para la recolección, resultados cuantitativos.

4.3.2.1 Contexto de la investigación

El experimento confirmatorio se realizó con dos grupos, uno de control y otro experimental.

Con el grupo experimental se trabajó de manera virtual, dado que no se tenían un espacio de trabajo para ello y los participantes no querían contacto físico con otras personas, debido a la situación de emergencia sanitaria. La consecución de participantes fue complicada, des pues de mucho esfuerzo sólo asistieron 4 niños entre los 10 y 12 años de edad, en ese momento los niños de Colombia asistían a clase virtual y no querían una carga académica adicional. Los participantes contaban con el permiso de sus padres de familia.

Con el grupo experimental el trabajo se hizo de manera presencial en una unidad residencial de la Ciudadela Llanos de Calibío ubicada en la calle 55N #22-80. La sala de la vivienda se adecuó como espacio de trabajo. Debido a la continuación del estado de emergencia sanitaria originado por la pandemia de COVID-19, se trabajó con un reducido grupo de 5 niños con edades entre 10 y 12 años, los cuales accedieron a reunirse, contando con la autorización de sus padres o acudientes. Dado que entre los niños compartían diariamente juegos, algunos son compañeros de clase y viven en casas vecinas, estas características permitieron desarrollar sesiones presenciales.

4.3.2.2 Pregunta de investigación

La pregunta formulada para este experimento confirmatorio fue ¿es ChildProgramming con RE una guía efectiva en para la apropiación de conceptos del del desarrollo del pensamiento computacional en los niños que participan en el experimento?

4.3.2.3 Objetivo del estudio:

Este estudio tiene por objetivo incluir la robótica educativa en el contexto de ChildProgramming como herramienta que incremente la motivación, la participación y la aprensión de los conceptos del pensamiento computacional.

4.3.2.4 Selección del estudio

Las actividades propuestas para el desarrollo del experimento están diseñadas con el modelo de ChildProgramming. Son los niños participantes los objetos de estudio.

El grupo de control se dividió en 2 grupos; uno compuesto por 2 niñas y el otro por un niño y una niña. En el grupo experimental se formaron dos equipos, uno con tres integrantes dos niños y una niña, y el otro equipo por un niño y una niña.

4.3.2.5 El experimento confirmatorio y los sujetos de investigación

Los sujetos de la investigación son niños con edades de 10 a 12 años; el nivel educativo de los participantes es de cuarto grado de primaria a segundo de bachillerato, quienes conforman las fuentes primarias de información. El experimento es confirmatorio, con el objetivo de evidenciar el resultado positivo de la aplicación de la extensión propuesta para ChildProgramming.

Para ejecutar el experimento confirmatorio, se dividió de la siguiente manera:

- Con el grupo de control, se desarrolló un curso de manera virtual con el nombre de **“Curso de Programación con Scratch para Niños”**, con una intensidad de 16 horas, dividido en 8 sesiones de dos horas.
- Para el grupo experimental se desarrolló un curso de **“Scratch + Arduino: Introducción a la Robótica”**, de forma presencial con una intensidad de 16 horas, divididas en 5 sesiones de 3 horas y 12 minutos.

4.3.2.6 Ejecución del experimento confirmatorio

Este experimento se ejecutó en el período comprendido entre el 20 de febrero y el 20 de marzo de 2021.

Para el grupo de control se aplicó el modelo de ChildProgramming, con algunas variantes ceñidas al entorno virtual. Las sesiones de entrenamiento se llevaron a cabo por la plataforma meet de Google. Se realizaron 8 sesiones de dos horas cada una. En la figura 19 se muestran algunos participantes del grupo de control.



Figura 19. Algunos participantes del grupo de control.

A lo largo del curso, los participantes, la distribución de los equipos, el ambiente de trabajo, materiales, metodología y entregables de la actividad se mantuvieron sin cambios, tal y como se indica en la tabla 24.

Curso “Programación con Scratch para Niños”	
Participantes:	<ul style="list-style-type: none"> Se trabajó con un grupo de 4 niños, de entre los 10 y 12 años de edad. En cuanto a género, tomaron parte del curso 3 niñas y 1 niño, quienes contaron con la autorización expresa y escrita por parte de su(s) padre(s) para tomar parte en el experimento, para lo cual previamente se les informó sobre los fines del curso y el tipo de actividades a desarrollar. Investigadores / Observadores.
Conformación de los Equipos:	Se conformaron dos grupos de dos niños cada uno. Los grupos se formaron en forma aleatoria. Así, resultaron un grupo de dos niñas y otro de un niño y una niña.
Ambiente de Trabajo:	Las sesiones del curso se desarrollaron de manera virtual, por la aplicación Google meet, cada niño contaba con un computador o Tablet que le permitía unirse a las sesiones de clase. Los niños se conectaban desde sus casas.
Materiales:	<ul style="list-style-type: none"> 1 computador portátil o Tablet para cada uno de los participantes. Aplicación Google meet. Aplicación Google Jamboard. Entorno de desarrollo Scratch online.
Metodología:	<p>La metodología empleada para esta actividad se describe a continuación:</p> <ol style="list-style-type: none"> Realizar la explicación de la actividad teniendo en cuenta la dinámica propuesta. Una vez iniciada los equipos de trabajo siguieron un protocolo de trabajo así: <ul style="list-style-type: none"> Leer el problema a resolver tratando de entender entre todos la situación Hacer una lluvia de ideas (Brainstorming) que les permitiera encontrar la forma correcta de solucionar el problema. Los equipos de trabajo establecieron la asignación de roles y se organizaron teniendo en cuenta una tarea asignada. Para garantizar orden y seguimiento de la actividad cada equipo realizó su plan de trabajo usando la aplicación Google jamboard. Cada uno asumía el rol que les correspondía. Cada niño realiza la programación según el plan realizado.
Entregables de la Actividad:	<p>Como resultado de esta actividad los equipos de trabajo debían entregar los siguientes productos entregables:</p> <ul style="list-style-type: none"> Proyecto codificado en el entorno de Desarrollo Scratch

	<ul style="list-style-type: none"> • Documentación del proyecto en la aplicación Google Jamboard.
--	--

Tabla 24. Curso “Programación con Scratch para Niños”

El curso se dividió en 8 sesiones, en las tres primeras de capacitación con la herramienta de programación Scratch adicionalmente se les aplicó en la primera sesión de capacitación un pres-test y también se les instruyó sobre los conceptos de pensamiento computacional, en las 5 sesiones restantes de entrenamiento se proponía una misión para los grupos de trabajo. En las siguientes tablas se describen cada una de las sesiones de entrenamiento realizadas durante el desarrollo del experimento confirmatorio “Scratch + Arduino: Introducción a la Robótica”, junto con sus correspondientes actividades:

Sesión 1 de entrenamiento	
Nombre de la Actividad:	Crear una historia.
Descripción:	En esta actividad los grupos deben recrear, en el entorno de desarrollo Scratch una historia, con un tema libre.
Entregables de la Actividad:	<p>Como resultado de esta actividad los equipos de trabajo debían entregar los siguientes productos entregables:</p> <ul style="list-style-type: none"> • Proyecto codificado en el entorno de Desarrollo Scratch, con una historia creada. • Documentación del proyecto en la aplicación Google Jamboard.
Defectos:	No hubo una documentación clara.

Tabla 25. Sesión 1 de entrenamiento

Sesión 2 de entrenamiento	
Nombre de la Actividad:	Campo de Baseball.
Descripción:	En esta actividad deben codificar en Scratch, una emulación de un campo de baseball, un bateador que golpee una pelota constantemente durante un tiempo.
Defectos:	Hubo algunos inconvenientes en la programación en el uso de los ciclos, el algoritmo creado no era claro.

Tabla 26. Sesión 2 de entrenamiento

Sesión 3 de entrenamiento	
Nombre de la Actividad:	Sami y las operaciones matemáticas
Descripción:	En esta actividad la misión a cumplir, es hacer que un personaje llamado Sami sea capaz de realizar una operación matemática (suma o multiplicación), con dos números ingresados por un usuario final.
Defectos:	<ul style="list-style-type: none"> • Hubo problemas con el uso de variables. • Aunque se les pidió sólo dos operaciones uno de los grupos codificó dos operaciones matemáticas más, resta y división, pero no hizo restricciones a las entradas (números suministrados por el usuario)

Tabla 27. Sesión 3 de entrenamiento.

Sesión 4 de entrenamiento	
Nombre de la Actividad:	Índice de masa corporal.
Descripción:	En esta misión los equipos deben calcular el índice de masa corporal de una persona, que ingresa el valor del peso y su estatura.
Defectos:	Algunos inconvenientes al pedir los datos al usuario

Tabla 28. Sesión 4 de entrenamiento

Sesión 5 de entrenamiento	
Nombre de la Actividad:	Frutalandia.
Descripción:	Se dividió en dos partes, en la primera parte se aplicó un pos-test, en la segunda se desarrolló la misión en esta actividad, los equipos diseñan un juego. El juego consiste en tomar la mayor cantidad de frutas y llevar un conteo de los puntos que se obtienen cada vez que capture o tome una fruta.
Defectos:	<ul style="list-style-type: none"> • Algoritmos no muy claros • No entregaron plan de trabajo

Tabla 29. Sesión 5 de entrenamiento

En grupo experimental se formaron dos equipos de trabajo, el conformado por dos niños y una niña se llamó “Walter 2.0” y el otro conformado por un niño y una niña lo denominaron “Walter”, se muestran en la figura 20.



Figura 20. A la derecha equipo “Walter”, a la izquierda equipo “Walter 2.0”

A lo largo del curso, los participantes, la distribución de los equipos y el ambiente de trabajo se mantuvieron sin cambios, tal y como se indica en la tabla 30.

Curso “Scratch + Arduino: Introducción a la Robótica”	
Participantes:	<ul style="list-style-type: none"> • Se trabajó con un grupo de 5 niños, de entre los 10 y 12 años de edad. En cuanto a género, tomaron parte del curso 3 niños y 2 niñas, quienes contaron con la autorización expresa y escrita por parte de su(s) padre(s) para tomar parte en el experimento, para lo cual previamente se les informó sobre los fines del curso y el tipo de actividades a desarrollar. • Investigadores / Observadores.
Conformación de los Equipos:	Se conformaron dos grupos de tres niños cada uno. Los grupos se formaron en forma voluntaria, sin intervención de los observadores. Así, resultaron un grupo de tres integrantes, dos niños y una niña, y otro de dos integrantes, un niño y una niña.

Ambiente de Trabajo:	<p>Ésta y todas las sesiones del curso tuvieron lugar en una casa de la Ciudadela Llanos de Calibío, en la sala que fue acondicionada especialmente para este propósito.</p> <p>En la sala se dispusieron dos mesas, una para cada equipo, además de una silla por cada uno de los niños. Las mesas se ubicaron con suficiente espaciamiento entre sí, brindando a los participantes suficiente espacio para trabajar sin interferirse entre sí, para poder hacer las observaciones con cierta comodidad, para tener un espacio de pruebas accesible para ambos equipos y, muy especialmente, atendiendo las recomendaciones de distanciamiento social con razón de la pandemia en curso en el momento del experimento. Por esta última razón también se hizo obligatorio ejecutar una desinfección del salón y del mobiliario antes de comenzar ésta y cada una de las sesiones, así como exigir de los participantes el cumplimiento de las normas de bioseguridad, como lavado y desinfección de manos, desinfección de calzado y uso de tapabocas.</p>
-----------------------------	--

Tabla 30. Curso “Scratch + Arduino: Introducción a la Robótica”

El curso se dividió en cinco sesiones de entrenamiento y, en cada una, se propuso una misión para ser ejecutada por los grupos de trabajo. En las siguientes tablas se describen cada una de las sesiones realizadas durante el desarrollo del experimento exploratorio “Scratch + Arduino: Introducción a la Robótica”, junto con sus correspondientes actividades:

Sesión 1 de entrenamiento	
Nombre de la Actividad:	Ensamblaje del robot.
Descripción:	Esta actividad se dividió en dos partes. En la primera, los estudiantes, trabajando por equipos, ensamblaron el robot mBot, en su variante conocida como mBot Ranger, siguiendo las instrucciones suministradas por el fabricante. En la segunda parte, los equipos probaron sus construcciones robóticas, dirigiéndolas por medio de la aplicación móvil mBlock, también desarrollada por el fabricante.
Materiales:	<p>A cada equipo se le entregó el material de trabajo respectivo el cual incluía:</p> <ul style="list-style-type: none"> • 1 computador portátil por mesa de trabajo, cada uno con su mouse. • 1 Hoja de papel por participante • 1 Lápiz por participante • 1 Borrador • 1 Cuestionario (Pretest) por participante • Kit robótico mBot, necesario para el mBot Ranger, junto con sus respectivas instrucciones de ensamblaje en versión impresa y en PDF. • Dispositivo móvil, necesario para dar instrucciones al mBot una vez ensamblado. Se utilizaron dispositivos basados en Android.
Metodología:	<p>La metodología empleada para esta actividad se describe a continuación:</p> <ol style="list-style-type: none"> 1. Asignar a cada equipo en su mesa de trabajo, la que ocuparán a lo largo del curso. 2. Entregar el pretest a todos los participantes, junto con lápiz y borrador, y pedirles que lo respondan, aclarando que el fin que persigue es investigativo y que no tiene incidencia en su continuación en el curso. 3. Realizar exposición de los conceptos de pensamiento computacional, por parte de los investigadores (instructores del curso). 4. Explicar a los participantes la actividad a realizar. 5. Entregar a cada grupo el computador, el kit robótico junto con las instrucciones de armado, tanto en versión impresa, como en PDF. 6. En este punto, los equipos comienzan el ensamblaje del robot, siguiendo las instrucciones del manual. Se les pidió que asignaran roles entre los miembros de cada equipo. Así, uno o dos de ellos se

	<p>encargaban de interpretar las instrucciones y ensamblar las piezas, mientras que otro verificaba al tiempo su correcta ejecución, de acuerdo con el manual. Estos roles eran intercambiados a medida que la actividad avanzaba.</p> <p>7. Una vez el robot estaba ensamblado, se hizo entrega a cada equipo del dispositivo móvil que contenía la aplicación para controlarlo y se les dio una muy breve explicación de su manejo.</p> <p>8. Los participantes de cada equipo se turnaron para jugar con su robot, manipulándolo por medio de la aplicación.</p>
Entregables de la Actividad:	<p>Como resultado de esta actividad los equipos de trabajo debían entregar los siguientes productos entregables:</p> <ul style="list-style-type: none"> • Pretest. • Robot mBot Ranger ensamblado y funcionando.
Defectos:	<p>Las piezas del robot no se ajustaron correctamente, generando inestabilidad en su estructura.</p>

Tabla 31. Sesión 1 de entrenamiento

Sesión 2 de entrenamiento	
Nombre de la Actividad:	Competencia de Baile
Descripción:	<p>Esta actividad se dividió en dos partes:</p> <ol style="list-style-type: none"> 1. Explicación teórica sobre pensamiento computacional y robótica: <ol style="list-style-type: none"> a. Conceptos del Pensamiento Computacional: <ul style="list-style-type: none"> • Definición del PC. • Abstracción. • Descomposición. • Diseño de algoritmos. b. Conceptos de Robótica: <ul style="list-style-type: none"> • ¿Qué es un robot? • Unidad de control. • Sensores. • Actuadores. 2. Desarrollo de la misión: los equipos deben programar el robot que ensamblaron en la primera sesión para que realice una secuencia de pasos de baile al ritmo de la música (movimientos adelante, atrás, giros, encender leds), la secuencia es libre.
Materiales:	<p>A cada equipo se le entregó su respectivo material de trabajo:</p> <ul style="list-style-type: none"> • 1 computador portátil por mesa de trabajo, cada uno con su mouse y con el programa mBlock instalado. • 1 Hoja de papel por participante • 1 Lápiz por participante • 1 Borrador • Robot mBot Ranger que ensamblaron en la sesión 1.
Metodología:	<p>La metodología empleada para esta actividad se describe a continuación:</p> <ol style="list-style-type: none"> 1. Presentar a los estudiantes los conceptos de Pensamiento Computacional y de robótica. 2. Entregar papel, lápiz y borrador a cada participante, un computador portátil por equipo y el mismo robot mBot Ranger que ensamblaron en la sesión inmediatamente anterior. 3. Explicar el funcionamiento básico de mBlock como entorno de programación del robot mBot. Mientras tienen el robot en sus manos, se realiza un refuerzo sobre los conceptos de robótica para que se familiaricen con ellos.

	<ol style="list-style-type: none"> 4. Explicar a los participantes la actividad a realizar, pidiéndoles que pongan en práctica los conceptos de descomposición y algoritmos. 5. Se les pide a los participantes que asignen entre ellos los roles de analista, desarrollador de algoritmos y programador. 6. En este punto, los equipos comienzan a discutir sobre la forma de resolver el problema planteado. 7. Los equipos debían escribir su versión de un algoritmo que ejecutara las acciones solicitadas. 8. Una vez que contaban con el algoritmo, los equipos procedían con la programación del robot. 9. Los equipos comprueban su solución poniendo en marcha el robot y, con él, la ejecución del programa logrado. 10. Se realizan nuevamente los pasos 6, 7, 8 y 9, ya sea para corregir acciones o efectos no deseados, o para realizar mejoras en el programa. 11. Durante el proceso, los observadores deben prestar especial atención a indicios que puedan mostrar los participantes sobre la aplicación de los conceptos de PC, haciendo énfasis en descomposición y abstracción.
Entregables de la Actividad:	<p>Como resultado de esta actividad los equipos de trabajo debían entregar los siguientes productos entregables:</p> <ul style="list-style-type: none"> • Algoritmo • Código realizado en mBlock. • Robot mBot Ranger programado y funcionando.
Defectos:	<ul style="list-style-type: none"> • El robot terminaba su secuencia antes de tiempo • Los participantes iban realizando el algoritmo al final de la programación

Tabla 32. Sesión 2 de entrenamiento

Sesión 3 de entrenamiento	
Nombre de la Actividad:	Robot sigue de líneas.
Descripción:	<p>Las dos partes de esta sesión fueron:</p> <ol style="list-style-type: none"> 1. Refuerzo de los conceptos de robótica y de Pensamiento Computacional. Se les recordó a los participantes los conceptos del PC y de robótica revisados, haciendo énfasis en funcionamiento y programación de los actuadores y del sensor seguidor de línea. 2. Programación y puesta en marcha del robot seguidor de línea. Se les pide a los equipos que programen el robot para que se desplace siguiendo una línea negra que describe un óvalo. También deben programar actuadores como leds y bocina para acompañar el movimiento del robot.
Materiales:	<p>A cada equipo se le entregó su respectivo material de trabajo:</p> <ul style="list-style-type: none"> • 1 computador portátil por mesa de trabajo, cada uno con su mouse y con el programa mBlock instalado. • 1 Hoja de papel por participante • 1 Lápiz por participante • 1 Borrador • Robot mBot Ranger que ensamblaron en la sesión 1.
Metodología:	<p>La metodología empleada para esta actividad se describe a continuación:</p> <ol style="list-style-type: none"> 1. Presentar a los estudiantes los conceptos de Pensamiento Computacional y de robótica.

	<ol style="list-style-type: none"> 2. Entregar papel, lápiz y borrador a cada participante, un computador portátil por equipo y el mismo robot mBot Ranger que ensamblaron en la sesión inmediatamente anterior. 3. Explicar el funcionamiento básico de mBlock como entorno de programación del robot mBot. Mientras tienen el robot en sus manos, se realiza un refuerzo sobre los conceptos de robótica para que se familiaricen con ellos. 4. Explicar a los participantes la actividad a realizar, pidiéndoles que pongan en práctica los conceptos de descomposición y algoritmos. 5. Se les pide a los participantes que asignen entre ellos los roles de analista, desarrollador de algoritmos y programador. 6. En este punto, los equipos comienzan a discutir sobre la forma de resolver el problema planteado. 7. Los equipos debían escribir su versión de un algoritmo que ejecutara las acciones solicitadas. 8. Una vez que contaban con el algoritmo, los equipos procedían con la programación del robot. 9. Los equipos comprueban su solución poniendo en marcha el robot y, con él, la ejecución del programa logrado. 10. Se realizan nuevamente los pasos 6, 7, 8 y 9, ya sea para corregir acciones o efectos no deseados, o para realizar mejoras en el programa. 11. Durante el proceso, los observadores deben prestar especial atención a indicios que puedan mostrar los participantes sobre la aplicación de los conceptos de PC (descomposición, abstracción, generalización, diseño de algoritmos)
Entregables de la Actividad:	<p>Como resultado de esta actividad los equipos de trabajo debían entregar los siguientes productos entregables:</p> <ul style="list-style-type: none"> • Algoritmo • Código realizado en mBlock. • Robot mBot Ranger programado y funcionando.
Defectos:	Se presentó alguna dificultad al manejar el sensor sigue líneas, incorporado en el mBot.

Tabla 33. Sesión 3 de entrenamiento

Sesión 4 de entrenamiento	
Nombre de la Actividad:	Guardián
Descripción:	<p>Las dos partes de esta sesión fueron:</p> <ol style="list-style-type: none"> 1. Refuerzo de los conceptos de robótica y de Pensamiento Computacional. Se les recordó a los participantes los conceptos del PC y de robótica revisados, haciendo énfasis en funcionamiento y programación de los actuadores y de los sensores sigue líneas e infrarrojo. 2. Programación y puesta en marcha del robot seguidor de línea. Se les entrega la misión a cada equipo, la cual consiste en programar al robot como un guardián. Se delimita una zona mediante una línea negra, esta es la zona a cuidar, una vez detecte un intruso (obstáculos), el robot se acerca al intruso y lo saca de su zona
Materiales:	<p>A cada equipo se le entregó su respectivo material de trabajo:</p> <ul style="list-style-type: none"> • 1 computador portátil por mesa de trabajo, cada uno con su mouse y con el programa mBlock instalado. • 1 Hoja de papel por participante • 1 Lápiz por participante

	<ul style="list-style-type: none"> • 1 Borrador • Robot mBot Ranger que ensamblaron en la sesión 1.
Metodología:	<p>La metodología empleada para esta actividad se describe a continuación:</p> <ol style="list-style-type: none"> 1. Presentar a los estudiantes los conceptos de Pensamiento Computacional y de robótica. 2. Entregar papel, lápiz y borrador a cada participante, un computador portátil por equipo y el mismo robot mBot Ranger que ensamblaron en la sesión inmediatamente anterior. 3. Explicar el funcionamiento básico de mBlock como entorno de programación del robot mBot. Mientras tienen el robot en sus manos, se realiza un refuerzo sobre los conceptos de robótica para que se familiaricen con ellos. 4. Explicar a los participantes la actividad a realizar, pidiéndoles que pongan en práctica los conceptos de descomposición y algoritmos. 5. Se les pide a los participantes que asignen entre ellos los roles de analista, desarrollador de algoritmos y programador. 6. En este punto, los equipos comienzan a discutir sobre la forma de resolver el problema planteado. 7. Los equipos debían escribir su versión de un algoritmo que ejecutara las acciones solicitadas. 8. Una vez que contaban con el algoritmo, los equipos procedían con la programación del robot. 9. Los equipos comprueban su solución poniendo en marcha el robot y, con él, la ejecución del programa logrado. 10. Se realizan nuevamente los pasos 6, 7, 8 y 9, ya sea para corregir acciones o efectos no deseados, o para realizar mejoras en el programa. 11. Durante el proceso, los observadores deben prestar especial atención a indicios que puedan mostrar los participantes sobre la aplicación de los conceptos de PC (descomposición, abstracción, generalización, diseño de algoritmos)
Entregables de la Actividad:	<p>Como resultado de esta actividad los equipos de trabajo debían entregar los siguientes productos entregables:</p> <ul style="list-style-type: none"> • Algoritmo • Código realizado en mBlock. • Robot mBot Ranger programado y funcionando.
Defectos:	Dificultades al programar dos sensores al tiempo y la programación de ciclos.

Tabla 34. Sesión 4 de entrenamiento

Sesión 5 de entrenamiento	
Nombre de la Actividad:	Duelo
Descripción:	La misión final del curso se le llamó duelo. El robot debía seguir un camino delimitado por una línea negra, al final de esta girar 180° detectar su oponente y disparar (hacer ruido), gana el primer equipo que dispare.
Materiales:	<p>A cada equipo se le entregó su respectivo material de trabajo:</p> <ul style="list-style-type: none"> • 1 computador portátil por mesa de trabajo, cada uno con su mouse y con el programa mBlock instalado. • 1 Hoja de papel por participante • 1 Lápiz por participante • 1 Borrador • Robot mBot Ranger que ensamblaron en la sesión 1.
Metodología:	La metodología empleada para esta actividad se describe a continuación:

	<ol style="list-style-type: none"> 1. Se les explica a los equipos el objetivo de la misión y el tiempo que tienen para cumplirla. 2. En este punto, los equipos comienzan a discutir sobre la forma de resolver el problema planteado. 3. Los equipos debían expresar la forma en que descompusieron el problema y escribir su versión de un algoritmo que ejecutara las acciones solicitadas. 4. Una vez que contaban con el algoritmo, los equipos procedían con la programación del robot. 5. Los equipos comprueban su solución poniendo en marcha el robot y, con él, la ejecución del programa logrado. 6. Se realizan nuevamente los pasos 2, 3, 4 y 5, ya sea para corregir acciones o efectos no deseados, o para realizar mejoras en el programa. 7. Durante el proceso, los observadores deben prestar especial atención a indicios que puedan mostrar los participantes sobre la aplicación de los conceptos de PC, revisados en las sesiones anteriores.
Entregables de la Actividad:	<p>Como resultado de esta actividad los equipos de trabajo debían entregar los siguientes productos entregables:</p> <ul style="list-style-type: none"> • Algoritmo • Código realizado en mBlock. • Robot mBot Ranger programado y funcionando.
Defectos:	Les tomó más del tiempo sugerido para entregar la misión.

Tabla 35. Sesión 5 de entrenamiento

4.3.2.7 Instrumentos y técnicas para recolección de información

Se diseñaron dos instrumentos de para recoger información con diferentes fines, Pre-test y Pos-test, y un protocolo de observación que fue adaptado:

El Pre-test es un cuestionario que se aplicó al inicio del curso. Fue diseñado en un formulario de Google para que fuera resuelto online, tanto para el grupo de control como para el experimental.

Para el grupo de control se aplicó un test de 23 preguntas con 8 preguntas con escala de satisfacción (ítems 1,2,3,4,5,6,7,8), 7 preguntas acerca de las preferencias de los participantes (ítems 9,10,11,18,20,21,22) y 8 preguntas (12,13,14,15,16,17,19,23) de conocimiento (5 de pensamiento computacional y 3 de programación) Anexo D.

En la tabla 36 se muestra la distribución de las variables, indicador y el número de la pregunta que se relaciona con el indicador.

Variables	Indicador	Ítem
Motivación	Expresa directamente su motivación	1, 2, 3, 4 ,5
Participación	Expresa directamente su participación en equipo	7, 8, 9
Apropiación del CT	Definición de CT	12
	Abstracción	13

	Algoritmos	14
	Descomposición	15
	Reconocimiento de patrones	16
	Programación	17, 19, 23

Tabla 36. Distribución de las preguntas según las variables a medir en el pre-test Grupo de control. Fuente Propia

Para el grupo experimental, el Pre-test, fue el mismo que se aplicó en el estudio exploratorio (anexo C).

Para el grupo de control se aplicó un Pos-test que fue diseñado en un formulario de Google para que fuera resuelto online, de 25 preguntas con 7 preguntas con escala de satisfacción (ítems 1,2,3,4,5,6,7), 11 preguntas acerca de las preferencias de los participantes (ítems 14,15,16,18,19,20,21,22,23,24,25) y 7 preguntas (8,9,10,11,12,13,17) de conocimiento (5 de pensamiento computacional y 3 de programación) Anexo E

En la tabla 37 se muestra la distribución de las variables, indicador y el número de la pregunta que se relaciona con el indicador.

Variables	Indicador	Ítem
Motivación	Expresa directamente su motivación	1, 2, 3, 4 ,5
Participación	Expresa directamente su participación en equipo	7
Apropiación del CT	Definición de CT	8
	Abstracción	9
	Algoritmos	10
	Descomposición	11
	Reconocimiento de patrones	12
	Programación	13, 17

Tabla 37. Distribución de las preguntas según las variables a medir en el pos-test Grupo de control. Fuente Propia

Para el grupo experimental, (véase el Anexo F) se aplicó un cuestionario de 28 preguntas, similar al Pre-test, aplicado al final del curso. De esas 28, 8 corresponden a preguntas tipo Likert con escala de 5 valores sobre motivación y participación, y 20 a preguntas abiertas, repartidas así: 7 acerca de las preferencias de los participantes, 6 sobre conocimientos de Pensamiento Computacional y otras 7 sobre robótica. La tabla 38 resume la clasificación de las preguntas en el Pos-test.

Variables	Indicador	Ítem
------------------	------------------	-------------

Motivación	Expresa directamente su motivación	1, 2, 3, 4, 5, 6, 22
Participación	Expresa directamente su participación en equipo	7
Apropiación del CT	Definición de CT	8
	Abstracción	9
	Algoritmos	10
	Descomposición	11
	Reconocimiento de patrones	21
	Programación	15

Tabla 38. Distribución de las preguntas según las variables a medir en el Pos-test – Grupo Experimental. Fuente Propia

El protocolo de observación es el mismo aplicado en el estudio exploratorio; en el grupo de control no fue posible su aplicación porque algunos alumnos no encendían las cámaras o tenían problemas con el audio, lo que impedía tomar datos precisos. En el grupo experimental sí se pudo aplicar el protocolo de observación, igual al adaptado para el estudio exploratorio.

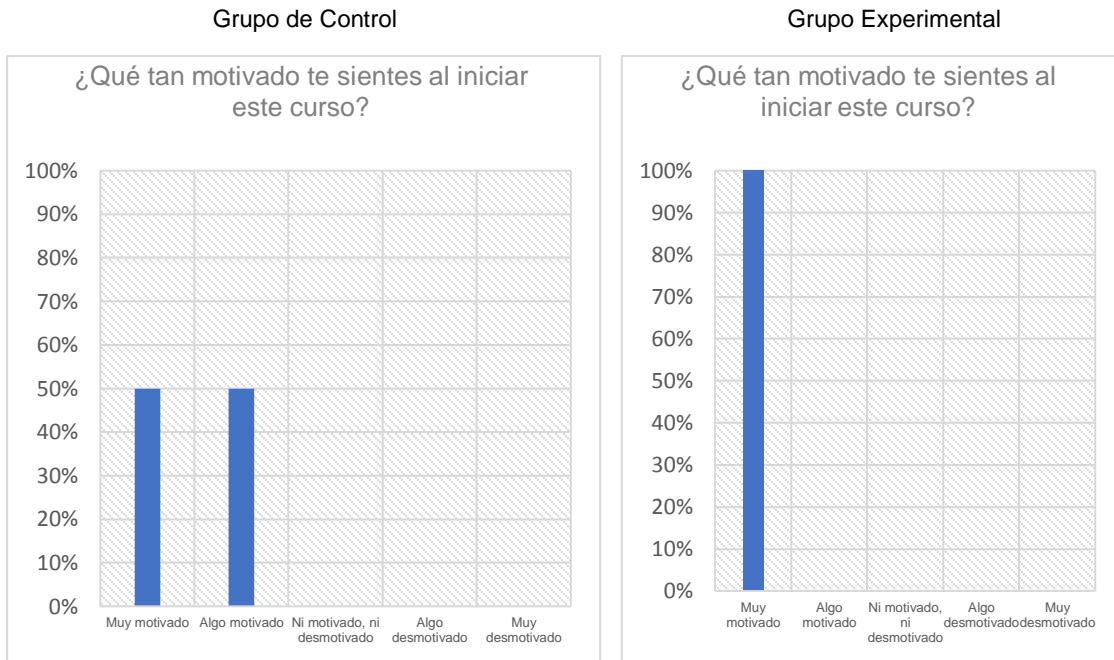
En la misma forma que en el estudio exploratorio, se usó la herramienta Dr. Scratch para evaluar los códigos generados tanto en el grupo de control como en el experimental.

4.3.2.8 Resultados cuantitativos.

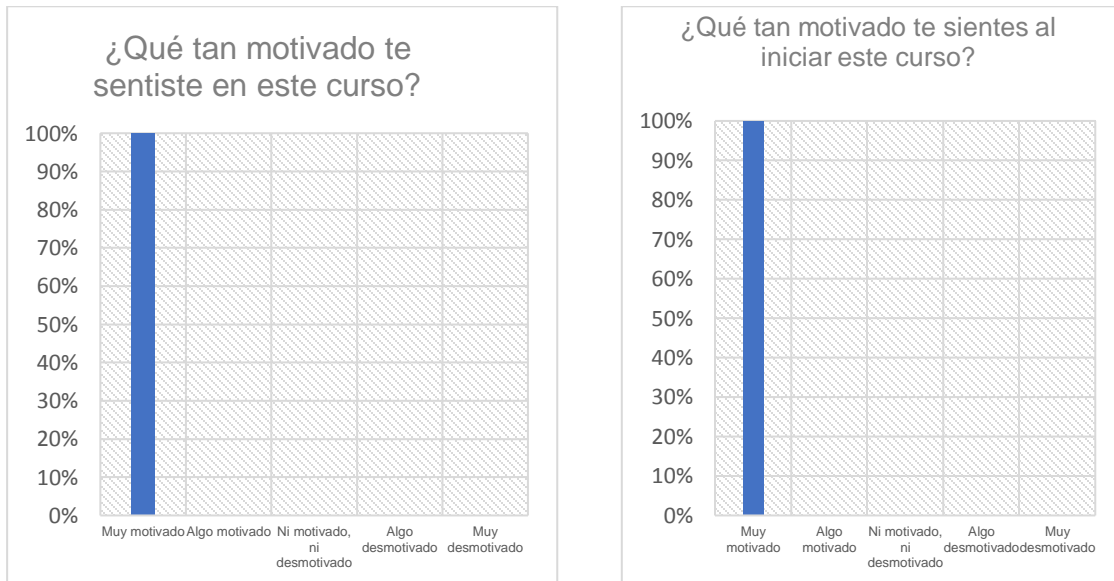
Los resultados cuantitativos para motivación, participación y apropiación del Pensamiento Computacional se obtuvieron a partir de mediciones directas.

Motivación.

Con el fin de medir la variación en la motivación de los participantes entre el inicio y el final del curso, se hizo una comparación entre los resultados del Pre-test y del Pos-test, tomando la pregunta N.1 como la más representativa. En ambos casos, se indaga directamente por la motivación de los estudiantes del grupo de control y el grupo experimental, obteniéndose los resultados que aparecen en la figura 21.



a) Pre-test



b) Pos-test

Figura 21. Distribución de las respuestas a la pregunta N.1 a) en el Pre-test y b) en el Pos-test. Comparativo entre el grupo de control y el grupo experimental.

Comparando los datos de la figura 21, se observa que, entre el inicio y el final del curso, el porcentaje de estudiantes que afirmó estar muy motivado fue del 100% tanto en el grupo de control como en el experimental

La aplicación del Protocolo de Observación solo se pudo aplicar al grupo experimental; en cuanto a los campos relativos a la motivación de los participantes del experimento se refiere, arrojó los resultados que se muestran en la tabla 39.

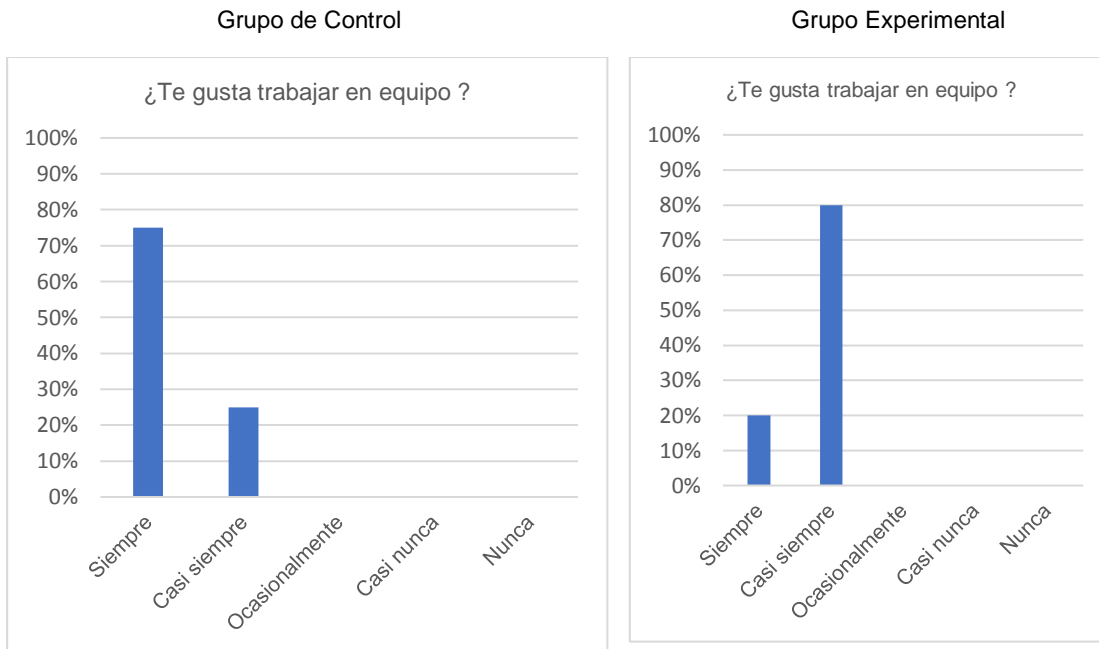
Criterio de Observación	P. 1	P.2	P.3	P.4	P.5
Está motivado para superar las dificultades en la realización de tareas.	4	4	3	4	4
Está motivado para comprender los errores para corregirlos.	4	4	4	3	4
Promedio de los criterios de observación	4	4	3,5	3,5	4

Tabla 39. Puntuación de los criterios relacionados con la motivación en el Protocolo de Observación para el grupo experimental.

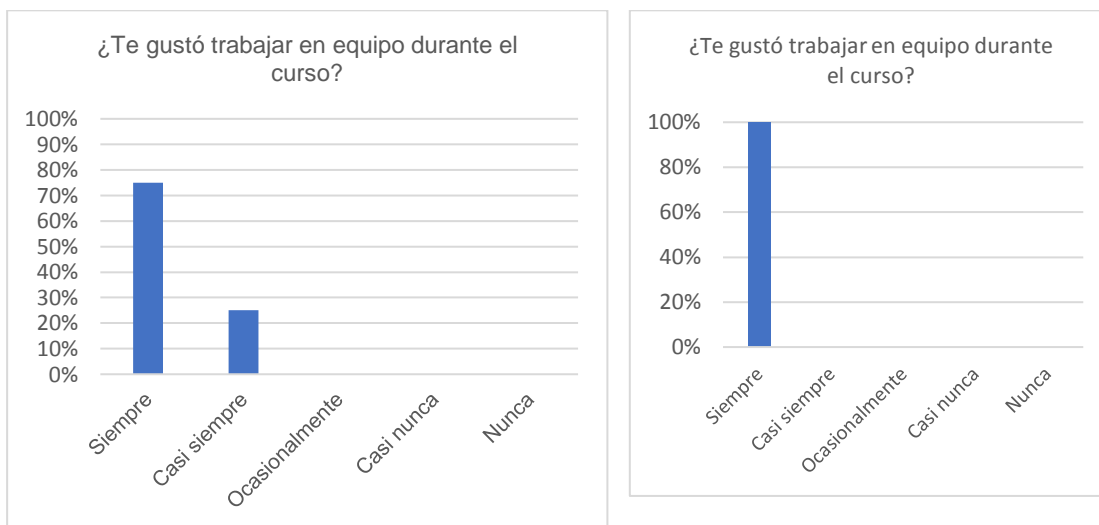
En los resultados de la tabla 39, se muestra que hay dos participantes con un puntaje de 3,5 en promedio para ambos criterios (se puede observar en casi todas las situaciones), los demás participantes tienen un puntaje de 4 para los dos criterios de observación. El rango de los promedios para los criterios de observación va desde 2 hasta 4.

Participación.

Con el propósito de medir la participación de los niños en la solución de las misiones en grupos se tuvo en cuenta su disposición para trabajar en grupo; por ello, se analizó la pregunta 7, realizada tanto en el Pre-test como en Pos-test, para los grupos de control y experimental.



a) Pre-test



b) Pos-test

Figura 22. Distribución de las respuestas a la pregunta N.7 a) en el Pre-test y b) en el Pos-test Grupo de Control y Experimental.

De los resultados obtenidos, y que se muestran en la figura 22, se evidenció el gusto inicial por trabajar en grupo, repartido así por igual entre siempre y casi siempre, cada uno con el 50% de los participantes para el grupo de control; mientras que para el grupo experimental el gusto inicial por trabajar en equipo, fue de siempre con un 25% y casi siempre un 75%. Al final del curso, los porcentajes cambiaron, en el grupo de control, la respuesta siempre la escogió el 75% y casi siempre el 25% de los participantes, para el grupo experimental la respuesta siempre la escogió el 100% de sus participantes.

En cuanto a los campos relativos a la participación de los participantes del experimento se refiere, la aplicación del Protocolo de Observación en el grupo experimental arrojó los resultados que se muestran en la tabla 40.

Criterios de Observación	P1	P2	P3	P4	P5
Coopera con otros miembros del grupo durante las actividades de manera positiva.	4	4	4	4	4
Obedece las reglas de comportamiento durante los proyectos.	4	4	4	3	4
Participa en trabajos en grupo.	4	4	4	4	3
Ayuda a los compañeros a resolver problemas	4	4	4	3	4
Promedio de los criterios de observación	4	4	4	3,5	3,75

Tabla 40. Puntuación de los criterios relacionados con la participación en el Protocolo de observación.

En promedio, según se desprende de la tabla, los alumnos tienen un rango de participación de 3,5 a 4. La escala del protocolo de observación se muestra en el Anexo E.

Apropiación del Pensamiento Computacional.

Se tomaron en cuenta los mismos criterios que en el estudio exploratorio. En la figura 23 se resume la evaluación de los conocimientos de los conceptos de PC evaluados en el Pos-test. No se tuvieron en cuenta los resultados del Pre-test porque los alumnos del grupo de control buscaron las respuestas en internet.

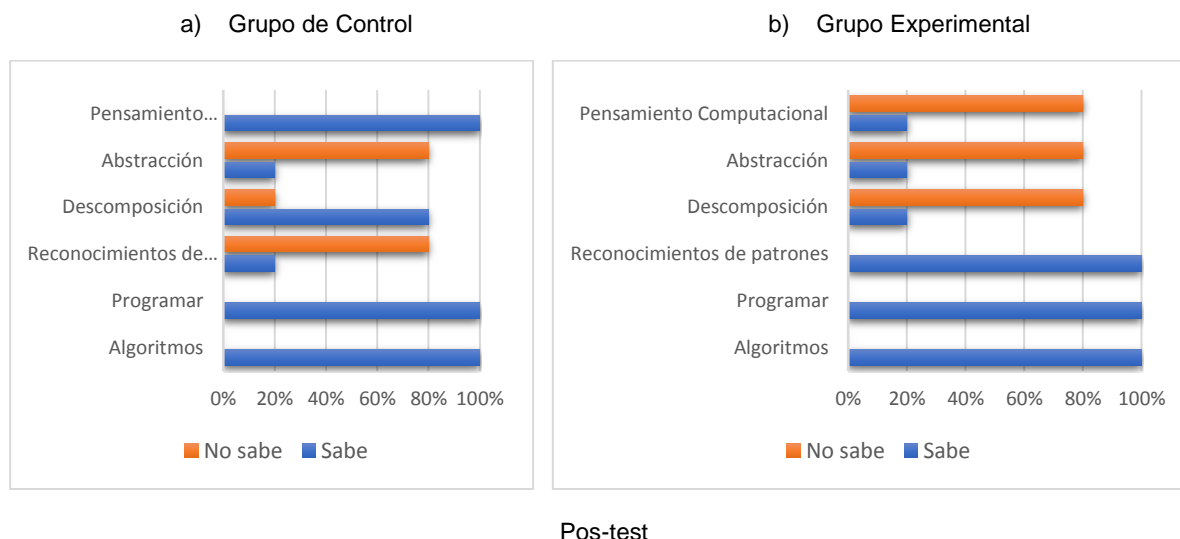


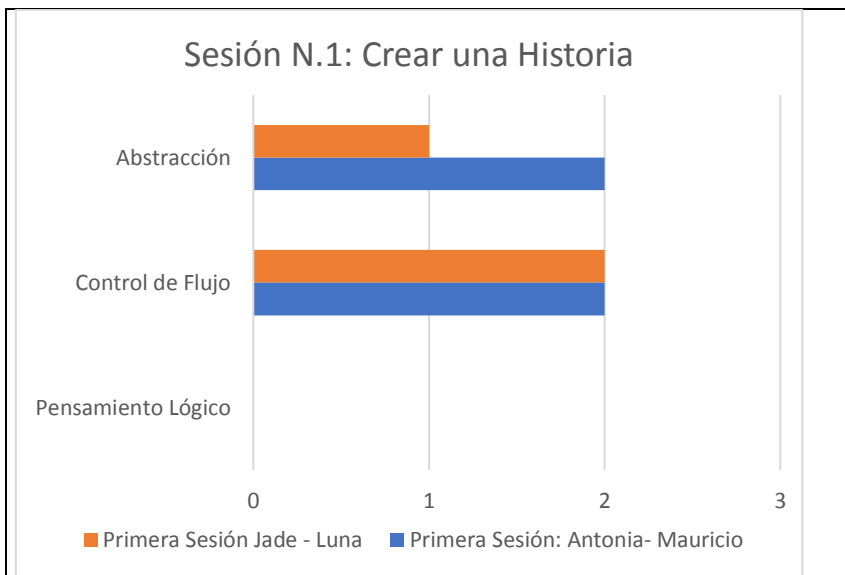
Figura 23. Comparativo Pos-test sobre la apropiación de los conceptos de PC en a) el grupo de control y b) el grupo experimental.

Las gráficas ponen de manifiesto que, al inicio del curso, tan solo la mitad de los niños dijo tener algún conocimiento sobre los conceptos de programación y reconocimiento de patrones, pero la totalidad de ellos desconocía los conceptos de algoritmos,

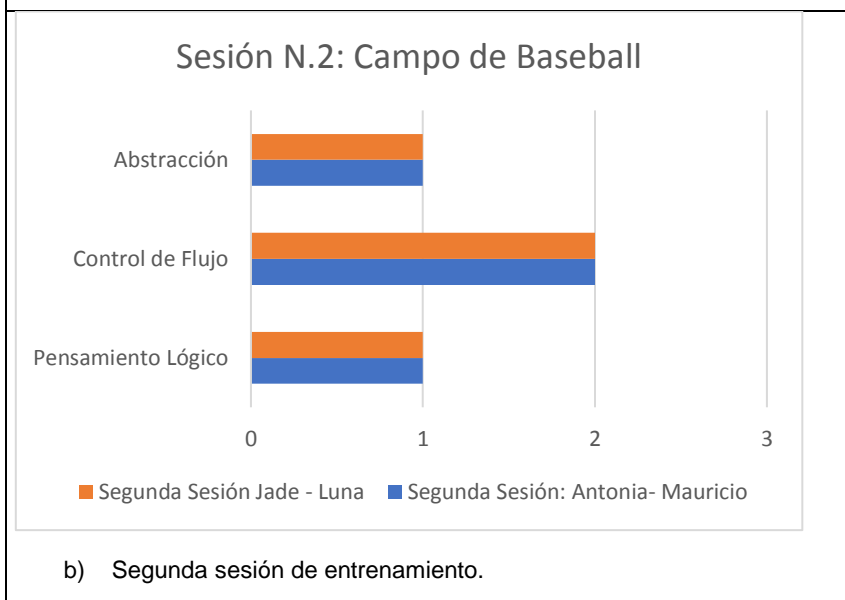
descomposición, abstracción y pensamiento computacional. Los resultados del Pos-test muestran una mejora en la apropiación de los demás conceptos del PC.

Del mismo modo que en el estudio exploratorio se usó la herramienta Dr. Scratch para validar los códigos generados por los equipos participantes, aquí se realizó en el estudio confirmatorio.

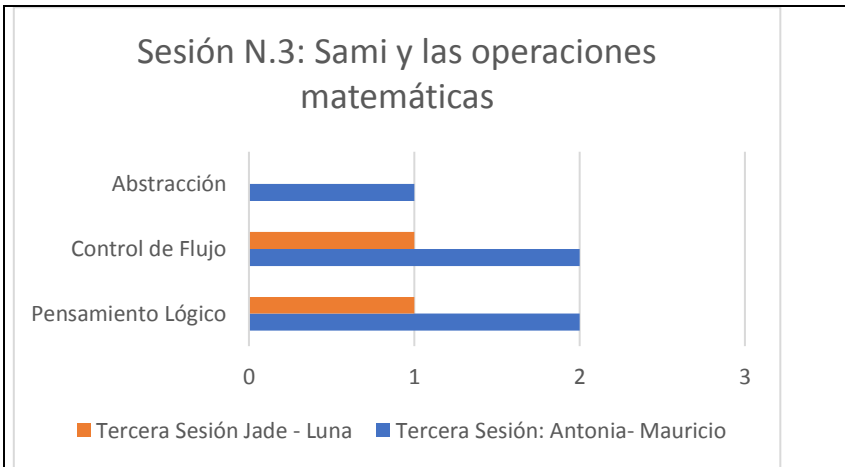
En el grupo de control se formaron dos equipos: “Jade-Luna” y “Antonia-Mauricio”, La comparación entre los códigos de los dos equipos, resultado de analizar los respectivos códigos con Dr. Scratch, se muestra en la figura 24 los resultados para el grupo de control y en la figura 25 para el grupo experimental.



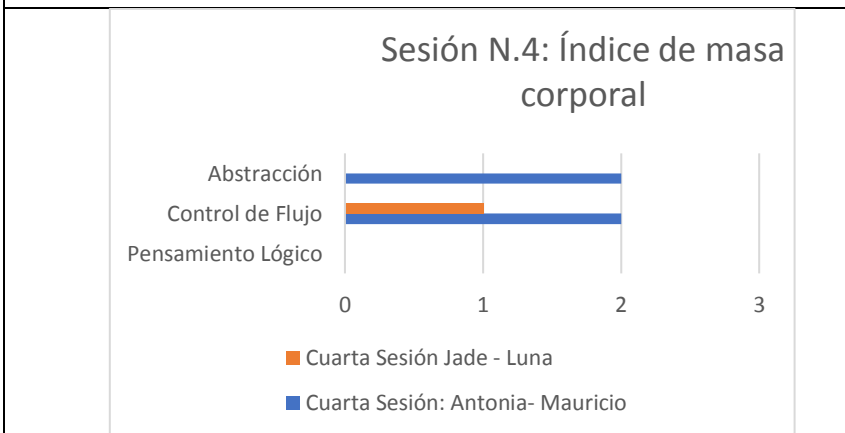
a) Primera sesión de entrenamiento.



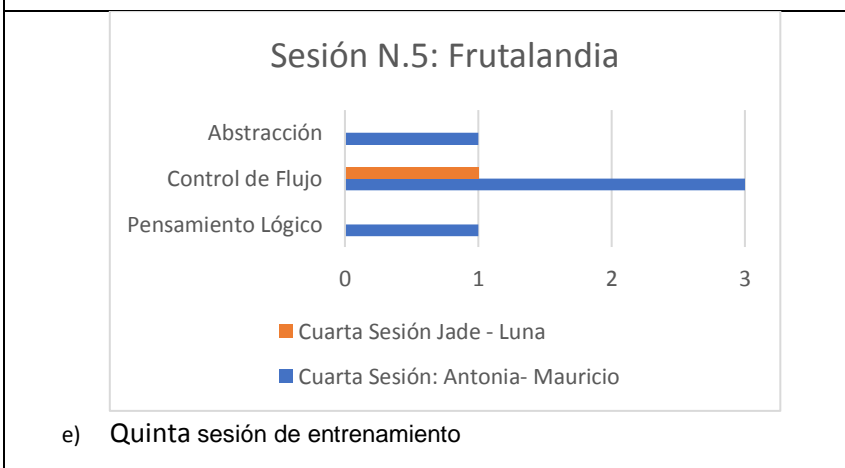
b) Segunda sesión de entrenamiento.



c) Tercera sesión de entrenamiento.

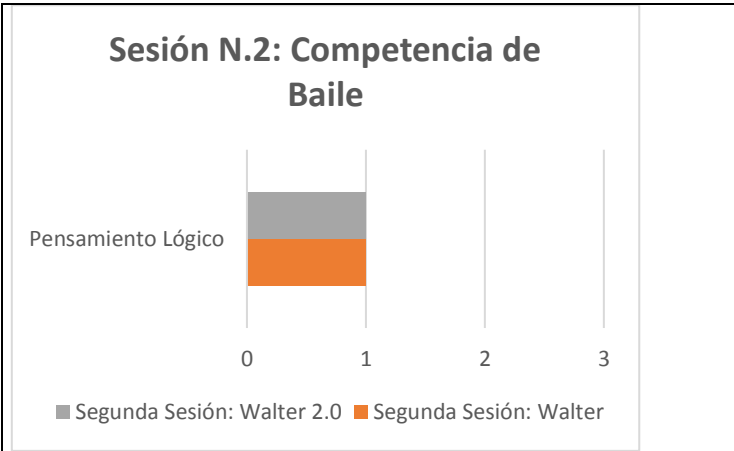


d) Cuarta sesión de entrenamiento.

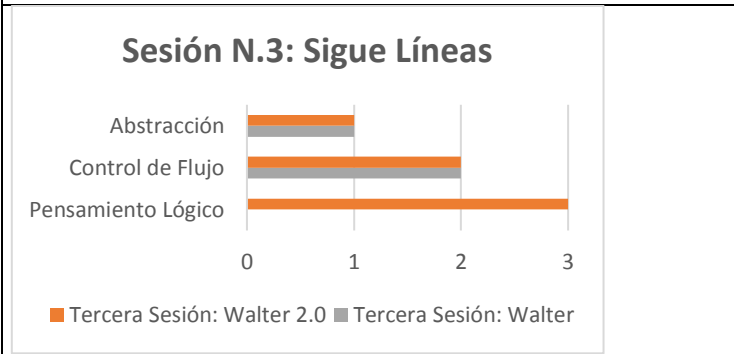


e) Quinta sesión de entrenamiento

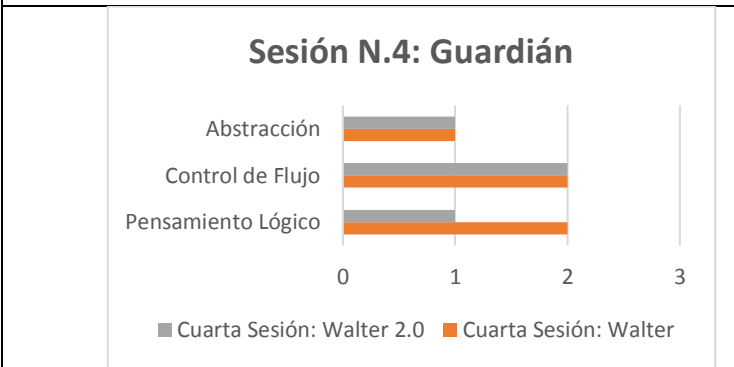
Figura 24. Resultados del análisis con Dr. Scratch del código de a) la primera sesión, b) la segunda sesión, c) la tercera sesión, d) cuarta sesión y e) quinta sesión de entrenamiento del grupo de control.



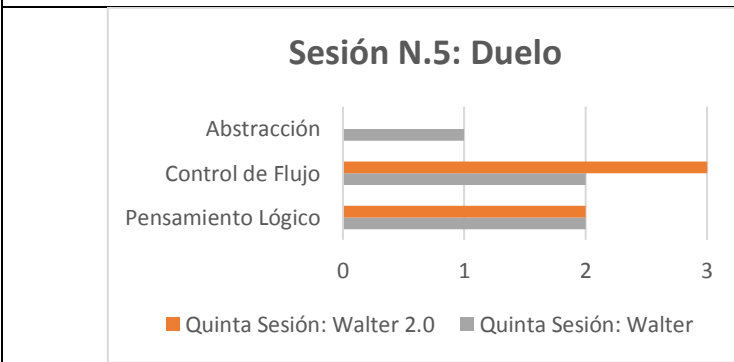
a) Segunda sesión de entrenamiento.



b) Tercera sesión de entrenamiento.



c) Cuarta sesión de entrenamiento.



d) Quinta sesión de entrenamiento

Figura 25. Resultados del análisis con Dr. Scratch del código de a) la segunda sesión, b) la tercera sesión, c) cuarta sesión y d) quinta sesión de entrenamiento del grupo experimental.

De acuerdo con los resultados de las figuras 24 y 25, fueron tres los aspectos del PC evaluados con la herramienta Dr. Scratch: pensamiento lógico, control de flujo y abstracción.

Resultados cualitativos.

En los resultados cualitativos tenemos las apreciaciones de los investigadores del experimento y las amenazas de validez

Apreciaciones de los investigadores durante el desarrollo del experimento exploratorio

Al hacer un contraste entre los resultados que miden la motivación de los participantes del grupo de control y el grupo experimental, no se observa modificación alguna al incluir los elementos de la robótica en el modelo de ChildProgramming. En ambos grupos la motivación fue alta durante el transcurso del experimento confirmatorio. En consecuencia, de acuerdo con los resultados obtenidos, no es posible validar la hipótesis N.1, que plantea un incremento significativo de la motivación en el desarrollo de CT en los niños que participan en el experimento.

Algo diferente ocurre en cuanto a la participación. En este aspecto se observa que la robótica, si mejora la participación tanto activa como en su gusto explícito por el trabajo en grupo. Además, es posible observar la descomposición que se hace de las piezas robóticas, tanto a la hora de ensamblar el robot como de su programación, esta observación se pudo evidenciar tanto en el estudio exploratorio como en el experimento confirmatorio. Por lo tanto, se da validez a la hipótesis N.2 planteada en el estudio exploratorio.

En cuanto a la apropiación de los conceptos de CT, se observó que los participantes del grupo experimental, aunque no eran capaces de expresar claramente un concepto mediante palabras, sí lo aplicaban de manera correcta durante el desarrollo de las misiones.

Amenazas de validez

Al igual que en el estudio exploratorio una de las amenazas de validez fue el muy reducido tamaño de participantes, tanto en el grupo experimental como en el de control.

La forma de realizar las sesiones de entrenamiento también puede influir en los resultados, con el grupo experimental se trabajó de forma presencial y con el de control, a distancia, de manera virtual. A causa de ello, en el grupo de control se presentaron algunos inconvenientes como fallas en las cámaras para poder observarlos, fallas en el audio y a veces problemas de conectividad.

4.4 ANÁLISIS

Dos partes muy importantes de la investigación fueron un estudio de tipo exploratorio y un experimento de tipo confirmatorio. El estudio exploratorio permitió definir las condiciones y los detalles del experimento confirmatorio. De ellos se pudo extraer como conclusión que la motivación permanece en niveles altos mientras no se presenten factores que la alteren, tales como las fallas que aparecieron en los robots durante el estudio exploratorio, causando que algunos de los participantes disminuyeran su participación y se sintieran desmotivados. En atención a lo anterior, no se validó la hipótesis N.1 planteada al inicio del estudio.

En cuanto a la hipótesis N.2, incluir la robótica educativa en el contexto de ChildProgramming como herramienta del desarrollo de pensamiento computacional, produce un incremento en la participación en los niños; esta hipótesis es válida teniendo en cuenta los resultados obtenidos en el experimento confirmatorio, dado que se notó que la participación de los niños aumentaba cuando programaban al robot. Además, una de las apreciaciones de los investigadores del proyecto es que realizar actividades como armar el robot ayuda a fomentar el trabajo colaborativo y participativo en los niños.

En cuanto a la hipótesis 3, formulada en la planeación del estudio, en relación con la apropiación de las habilidades del pensamiento computacional, puede decirse que, al incluir la robótica en el contexto de ChildProgramming se evidenció en el experimento confirmatorio un incremento en la apropiación de conceptos del PC. Aunque los niños no definían los conceptos por medio de palabras, sí manifestaban su realización práctica por medio de sus acciones y comportamiento; por ejemplo, la descomposición, el reconocimiento de patrones, y el diseño de algoritmos. En el experimento confirmatorio, tanto en el grupo de control como en el experimental se observa un incremento en la apropiación. De las gráficas, se nota que en el grupo de control los equipos no avanzan a la par, mientras que en el grupo experimental, los dos grupos avanzan de igual forma en la apropiación. En este sentido podemos validar la hipótesis 3, quedando para trabajos futuros poder contrastar la hipótesis 3 con equipos que tengan sesiones presenciales en ambos grupos tanto el control como el experimental.

5. CONCLUSIONES Y TRABAJO FUTURO

En este último capítulo se exponen, en primer término, las conclusiones a las que se ha llegado tras la elaboración de este trabajo; una segunda parte se encarga de dar recomendaciones, dirigidas a investigadores y personas en general interesadas en el tema, sobre particularidades a tener en cuenta al abordar éste u otro proyecto de características similares; finalmente, hay una orientación de lo que, a nuestro criterio, constituyen direcciones hacia las que podrían encaminarse futuros trabajos que pudieran tomar éste en consideración, ya sea como punto de partida, como complemento o como una referencia.

5.1 CONCLUSIONES

Durante la última década se ha visto un notorio incremento en el uso de la robótica como un vehículo tangible para impulsar el desarrollo de habilidades de Pensamiento Computacional. En general, existe coincidencia entre los investigadores al ubicar al construccionismo como la teoría dominante en la intersección entre la Robótica Educativa y el Pensamiento Computacional. La robótica educativa se presenta como un instrumento propicio para el aprendizaje mediante la filosofía construccionista ya que permite trasladar la experiencia obtenida mediante la interacción del robot con el entorno en un determinado contexto, en ideas que transforman las percepciones y conocimientos previos.

La principal contribución de la ER para el desarrollo del PC, radica en su capacidad de involucrar el uso de computadoras y el diseño, construcción y operación de robots, uniendo la abstracción y la realidad de una manera concreta, pasando de lo abstracto a lo tangible. Como consecuencia, las actividades de ER necesarias para ello, que incluyen montaje, programación y pruebas, resaltan las habilidades del PC; de otra parte, mientras brindan una oportunidad de desarrollar habilidades de PC, esas mismas características físicas y tangibles asociadas con los robots los convierten en un factor motivador que favorece el interés de los niños, motivándolos y animándolos a resolver problemas auténticos que son significativos para ellos. Por esto, puede decirse que la ER tiene al menos el potencial de producir una mejora significativa no sólo en el interés y la motivación sino también, dada su naturaleza, en la comunicación, el trabajo colaborativo, la creatividad, la autonomía y el liderazgo.

Por sí sola, la RE no afecta directamente el aprendizaje de los estudiantes; se requiere de una filosofía educacional, un ambiente de aprendizaje y una metodología apropiadas para su incorporación exitosa en la enseñanza. En esta vía, la extensión ChildProgramming-ER propuesta, estructurada en cuatro paquetes metodológicos, permite integrar los aspectos de la RE en el desarrollo del PC, al proporcionar un conjunto de directrices concretas para la utilización de aspectos de Robótica Educativa durante actividades encaminadas hacia el desarrollo del Pensamiento Computacional en niños entre los 10 y los 12 años de edad. La extensión incluye una guía de prácticas para orientar un curso de robótica, siguiendo la metodología de ChildProgramming.

En cuanto a los resultados del experimento confirmatorio se refiere, se concluye que, con el paso de las sesiones, se hizo notoria la mejora en las habilidades de Pensamiento Computacional de ambos grupos, sobre todo en aquellas medidas por Dr. Scratch. Tanto en el grupo experimental como en el de control, el interés y la motivación se mantuvieron en el nivel más alto posible; es decir, no se notaron diferencias entre el modelo ChildProgramming y la extensión ChildProgramming-RE.

Al incluir la robótica educativa en el contexto de ChildProgramming como herramienta del desarrollo del pensamiento computacional, se produce un incremento significativo en la participación en el desarrollo CT en los niños que participan en el experimento.

Al evaluar los grupos experimental, que trabajó con la extensión ChildProgramming-RE, y de control, que trabajó con ChildProgramming, con ayuda de la herramienta Dr. Scratch, no se encontraron entre ellos diferencias apreciables en la apropiación de los conceptos de Pensamiento Computacional relativos a algoritmos y abstracción; sin embargo, de acuerdo con los datos recopilados, el concepto de reconocimiento de patrones y el concepto mismo de Pensamiento Computacional fue mejor calificado, con marcada diferencia, en el grupo experimental, con respecto al grupo de control; caso contrario ocurrió con el concepto de descomposición, visiblemente mejor en el grupo de control con respecto al grupo experimental.

En referencia al grupo experimental, es necesario decir que demostró además evidencias de aplicación de los conceptos de Pensamiento Computacional estudiados al realizar actividades diferentes a de la programación, tales como diseñar, ensamblar y manipular el robot. Aun teniendo dificultades para expresar con palabras la definición de los conceptos, los participantes de este grupo los aplicaban en las misiones a las que se enfrentaban.

Finalmente, como es necesario recalcar que no es posible saber, a partir de este trabajo, cómo y hasta qué punto, ha influido la pandemia de COVID-19 en los resultados del estudio exploratorio y del experimento confirmatorio.

5.2 TRABAJO FUTURO

De esta monografía se derivan algunas posibles líneas de trabajos futuros que pueden ser una prolongación de, o que se encuentren relacionados con este; entre ellas, se destacan:

Realizar nuevos experimentos controlados que incluyan un número ostensiblemente mayor de estudiantes, que permitan subsanar las limitaciones debidas al pequeño tamaño de la muestra de este estudio, y poder así confirmar o refutar los resultados expuestos.

Aprovechando la metodología ChildProgramming propuesta, realizar un estudio comparativo entre la robótica tradicional y los nuevos entornos de robótica virtual, para determinar las diferencias existentes entre ambas cuando se habla de desarrollo de habilidades del Pensamiento Computacional, interés y motivación.

Investigar cuáles son las diferencias, en caso de existir, entre la utilización de diferentes kits robóticos para el desarrollo del Pensamiento Computacional.

Diseñar y crear un entorno de programación para algún kit robótico existente, dirigido al desarrollo y evaluación de las habilidades de Pensamiento Computacional.

Desarrollar herramientas evaluativas que permitan comparar con mayor facilidad las habilidades de Pensamiento Computacional obtenidas exclusivamente a través de la programación con las adquiridas a través de herramientas tangibles como la robótica.

Bibliografía

- [1] J. Wing, «Computational Thinking Benefits Society,» 10 01 2014. [En línea]. Available: <http://socialissues.cs.toronto.edu/2014/01/computational-thinking/>. [Último acceso: 07 02 2018].
- [2] J. M. Wing, «Computational thinking's influence on research and education for all,» *Italian Journal of Educational Technology*, vol. 25, nº 2, pp. 7-14, 2017.
- [3] Carnegie Mellon's Robotic Academy, Introduction to Programming LEGO Mindstorms EV3. Teacher's Guide, 2014.
- [4] A. Yadav, N. Zhou, C. Mayfield, S. Hambrusch y J. T. Korb, «Introducing Computational Thinking in Education Courses,» de *SIGCSE'11*, Dallas, Texas (USA), 2011.
- [5] M. Weigend, «Smartwalk: Computer Science on the Schoolyard,» de *Tomorrow's Learning: Involving Everyone. Learning with and about Technologies and Computing - 11th IFIP TC 3 World Conference on Computers in Education*, Dublín, Irlanda, 2017.
- [6] J. A. Hurtado, C. A. Collazos, S. T. Cruz y Ó. E. Rojas, «ChildProgramming: Una Estrategia de Aprendizaje y Construcción de Software Basada en la Lúdica, la Colaboración y la Agilidad,» *Revista Universitaria en Telecomunicaciones, Informática y Control*, vol. 1, nº 1, pp. 9-14, 2012.
- [7] A. M. Chimunja, C. A. Collazos y J. A. Hurtado, «ChildProgramming-C: como una mejora de la dimensión colaborativa del modelo ChildProgramming,» *Entre Ciencia e Ingeniería*, vol. 11, nº 22, pp. 67-75, 2017.
- [8] S. T. Cruz Sánchez y Ó. E. Rojas Ordóñez, «Un modelo para la enseñanza de la programación de software en niños a través de estrategias colaborativas,» Popayán, Colombia, 2013.
- [9] R. F. Zúñiga, J. A. Hurtado y C. A. Collazos, «Comprendiendo los procesos de abstracción computacional en los niños: un estudio de caso exploratorio,» *I+T+C. Investigación, Tecnología y Ciencia*, pp. 57-66, 2014.
- [10] M. Hacker, «Integrating computational thinking into technology and engineering education,» *Technology and Engineering Teacher*, pp. 8-14, Enero 2018.
- [11] I. Fronza, N. El Iioini y L. Corral, «Leveraging Robot Programming to Foster Computational Thinking,» de *Proceedings of the 9th International Conference on Computer Supported Education (CSEDU 2017)*, Porto, Portugal, 2017.
- [12] S. Ghosh, «Children should be taught computer science - not programming,» 27 02 2014. [En línea]. Available: <http://www.alphr.com/news/education/387319/children-should-be-taught-computer-science-not-programming>. [Último acceso: 08 03 2018].
- [13] Y. Tabesh, «Computational Thinking: A 21st Century Skill,» *Olympiads in Informatics*, vol. 11, pp. 65-70, 2017.

- [14] P. Román Graván, C. Hervás Gómez y J. L. Guisado Lizar, «Experiencia de innovación educativa con robótica en la Facultad de Ciencias de la Educación de la Universidad de Sevilla (España),» de *Innovación docente y uso de las TIC en educación*, Málaga (España), UMA Editorial, 2017.
- [15] D. Catlin y J. Woollard, «Educational Robots and Computational Thinking,» de *Proceedings of 4th International Workshop Teaching Robotics, Teaching with Robotics & 5th International Conference Robotics in Education*, Padova (Italia), 2014.
- [16] S.-C. Kong y C.-C. Lao, «Computational Thinking Development through Programmable Robotics Activities in STEM Education in Primary Schools,» de *Proceedings of the 25th International Conference on Computers in Education*, 2017.
- [17] S. Atmatzidou y S. Demetriadis, «How to Support Students' Computational Thinking Skills in Educational Robotics Activities,» de *Proceedings of 4th International Workshop Teaching Robotics, Teaching with Robotics & 5th International Conference Robotics in Education*, Padova (Italia), 2014.
- [18] C. Saygin, T. Yuen, H. Shipley, H.-d. Wan y D. Akopian, «Design, Development, and Implementation of Educational Robotics Activities for K-12 Students,» *American Society for Engineering Education*, 2012.
- [19] F. Á. Bravo Sánchez y A. Forero Guzmán, «La robótica como un recurso para facilitar el aprendizaje y desarrollo de competencias generales,» *Education in the Knowledge Society (EKS)*, pp. 120-136, 2017.
- [20] H. R. Sarmiento, C. A. S. Reis, V. Zaramella, L. D. A. Almeida y C. A. Tacla, «Supporting the Development of Computational Thinking: A Robotic Platform Controlled by Smartphone,» de *Learning and Collaboration Technologies - Second International Conference*, Los Angeles, CA (USA), 2015.
- [21] F. Benitti, «Exploring the educational potential of robotics in schools: A systematic review,» *Computers & Education*, pp. 978-988, 2012.
- [22] S. Atmatzidou y S. Demetriadis, «A Didactical Model for Educational Robotics Activities: A Study on Improving Skills Through Strong or Minimal Guidance,» de *Educational Robotics in the Makers Era*, D. Alimisis, M. Moro y E. Menegatti, Edits., Springer International Publishing, 2017, pp. 58-72.
- [23] T. Djambong y V. Freiman, «Task-Based Assessment of Students' Computational Thinking Skills Developed Through Visual Programming or Tangible Coding Environments,» de *13th International Conference on Cognition and Exploratory Learning in Digital Age (CELDA 2016)*, 2016.
- [24] J. M. Wing, «Computational Thinking,» *Communications on the ACM*, vol. 49, nº 3, pp. 33-35, Marzo 2006.
- [25] C. C. Selby y J. Woollard, «Computational Thinking: The Developing Definition,» 2013.
- [26] E. Segredo, G. Miranda y L. Coromoto, «Towards the Education of the Future: Computational Thinking as a Generative Learning Mechanism,» *Education in the Knowledge Society*, vol. 18, nº 2, pp. 33-58, 2017.

- [27] H. I. Haseski, U. Ilic y U. Tugtekin, «Defining a New 21st Century Skill-Computational Thinking: Concepts and Trends,» *International Education Studies*, vol. 11, nº 4, pp. 29-42, 2018.
- [28] K. Brennan y M. Resnick, «New frameworks for studying and assessing the development of computational thinking,» de *American Educational Research Association (AERA) 2012*, Vancouver, Canadá, 2012.
- [29] International Society for Technology in Education (ISTE) y Computer Science Teachers Association (CSTA), «Operational Definition of Computational Thinking for K-12 Education,» 2011. [En línea]. Available: <http://www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf?sfvrsn=2>. [Último acceso: 12 04 2018].
- [30] C. Mohtadi, M. Kim y J. Schlosser, «Why integrate computational thinking into a 21st century engineering curriculum?,» de *41st SEFI Conference*, Leuven (Bélgica), 2013.
- [31] «Exploring Computational Thinking,» [En línea]. Available: <https://edu.google.com/resources/programs/exploring-computational-thinking/#!ct-overview>. [Último acceso: 30 03 2018].
- [32] D. Alimisis, «Educational robotics: Open questions and new challenges,» *Themes in Science & Technology Education*, vol. 6, nº 1, pp. 63-71, 2013.
- [33] K.-Y. Chin, Z.-W. Hong y Y.-L. Chen, «Impact of Using an Educational Robot-Based Learning System on Student's Motivation in Elementary Education,» *IEEE Transactions on Learning Technologies*, vol. 7, nº 4, pp. 333-345, 2014.
- [34] D. Catlin y M. Blamires, «The Principles of Educational Robotics Applications (ERA): A framework for understanding and developing educational robots and their activities,» de *Proceedings of Constructionism*, París, Francia, 2010.
- [35] T. Phetsrikrant, W. Massagram y A. Harfield, «First steps in teaching computational thinking through mobile technology and robotics,» *Asian International Journal of Social Sciences*, vol. 17, nº 3, pp. 37-52, 2017.
- [36] H. F. Orejuela, A. A. García, J. A. Hurtado y C. A. Collazos, «Analizando y Aplicando la Gamificación del Proceso ChildProgramming,» *Revista colombiana de computación*, vol. 14, nº 2, pp. 7-23, 2013.
- [37] J. Turner, R. Baker y F. Kellner, «Theoretical Literature Review: Tracing the Life Cycle of a Theory and Its Verified and Falsified Statements,» *Human Resource Development Review*, vol. 17, nº 1, pp. 34-61, 2018.
- [38] A. Ioannou y E. Makridou, «Exploring the potentials of educational robotics in the development of computational thinking: A summary of current research and practical proposal for future work,» *Education and Information Technologies*, vol. 23, nº 1, 2018.
- [39] Istikomah y C. Budiyanto, «The Contribution of Educational Robotics and Constructivist Approach to Computational Thinking in the 21st Century,» de *ICCSET 2018. The 1st International Conference on Computer Science and Engineering*, Kudus, Indonesia, 2018.
- [40] Y. A. Caballero González, A. García Valcárcel y A. García Holgado, «Learning computational thinking and social skills development in young children through problem solving with educational robotics,»

de *TEEM'19: Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing*, 2019.

- [41] Y. A. Caballero González y A. García Valcárcel, «A robotics-based approach to foster programming skills and computational thinking: Pilot experience in the classroom of early childhood education,» de *TEEM'18: Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturalism*, Salamanca, España, 2018.
- [42] Y. Caballero González y A. García Valcárcel, «Robótica como Recurso Educativo para la Formación del Pensamiento Computacional en Educación Inicial,» de *Simposio Conectando redes. La relación entre la investigación y la práctica educativa. REUNI+D y RILME*, Santiago de Compostela, España, 2018.
- [43] F. A. Aristawati, C. Budiyo y R. A. Yuana, «Adopting Educational Robotics to Enhance Undergraduate Students' Self-Efficacy Levels of Computational Thinking,» *Journal of Turkish Science Education*, vol. 15, pp. 42-50, 2018.
- [44] Y.-A. Caballero González y A. García Valcárcel, «Fortaleciendo el pensamiento computacional y habilidades sociales mediante actividades de aprendizaje con robótica educativa en niveles escolares iniciales,» *Píxel-BIT Revista de Medios y Educación*, nº 58, pp. 117-142, 2020.
- [45] C. S. González González, «Estado del arte en la enseñanza del pensamiento computacional y la programación en la etapa infantil,» *Education in the Knowledge Society (EKS)*, vol. 20, pp. 15-28, 25 07 2019.
- [46] D. Alimisis, «Educational robotics: Open questions and new challenges,» *Science & Technology Education*, vol. 6, nº 1, pp. 63-71, 2013.
- [47] R. Isnaini y W. B. Cucuk, «The Influence of Educational Robotics to Computational Thinking Skill in Early Childhood Education,» de *ICCSET. The 1st International Conference on Computer Science and Engineering Technology Universitas Muria Kudus*, Kudus, Indonesia, 2018.
- [48] W.-Y. Huang, C.-F. Hu y C.-C. Wu, «The Use of Different Kinds of Robots to Spark Student Interest in Learning Computational Thinking,» de *International Conference on Learning and Teaching in Computing and Engineering (LaTICE)*, 2018.
- [49] Y. A. Caballero-González y A. García Valcárcel, «Learning with Robotics in Primary Education: A Means of Stimulating Computational Thinking,» *Education in the Knowledge Society (EKS)*, nº 21, 2020.
- [50] R. Isnaini, C. Budiyo y I. Widiastuti, «Robotics-based learning to support computational thinking skills in early childhood,» *AIP Conference Proceedings*, vol. 2194, nº 1, 2019.
- [51] A. García-Valcárcel y Y. A. Caballero-González, «Robótica para desarrollar el pensamiento computacional en educación infantil,» *Comunicar*, vol. 27, nº 59, pp. 63-72, 2019.
- [52] J. Haut, P. Bustos, M. Paoletti y N. Perales, «Code2Bot, a social robot for the classroom,» de *Actas de la XVI Conferencia CAEPIA*, Albacete, 2015.
- [53] S. Anwar, N. A. Basou y M. Menekse, «A Systematic Review of Studies on Educational Robotics,» *Journal of Pre-College Engineering Education Research (J-PEER)*, vol. 9, nº 2, 2019.

- [54] T. Phetsrikran, W. Massagram, T. Phoka y A. Harfield, «A Feasibility Study of Arduation Bot: An Educational Robotics and Mobile Application Kit for Computational Thinking Skills,» *22nd International Computer Science and Engineering Conference (ICSEC)*, pp. 1-6, 2018.
- [55] M. Vallance y P. Towndrow, «Mapping Computational Thinking for a Transformative Pedagogy,» de *Computational Thinking in the STEM Disciplines*, Springer, 2018, pp. 301-325.
- [56] S. Atmatzidou y S. Demetriadis, «Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences,» *Robotics and Autonomous Systems*, vol. 75, nº B, pp. 661-670, 2016.
- [57] A. García Valcárcel y Y. A. Caballero González, «Robotics to Develop Computational Thinking in Early Childhood Education,» *Comunicar*, vol. 27, nº 59, pp. 63-72, 2019.
- [58] Y. Liu y J. Rojas, «Evaluation of the ROOT Robot System and Curriculum to Improve Computational Thinking in Chinese Children,» de *IEEE Region 10 Humanitarian Technology Conference*, Depok, Indonesia, 2019.
- [59] c. y. d. d. G. d. E. Ministerio de educación, «Gosario de conceptos y términos robóticos,» [En línea]. Available: <http://el.uy/doc/robotica/mooc.educalab.es.%202017.%20Glosario%20de%20t%C3%A9rminos%20y%20conceptos%20rob%C3%B3ticos.pdf>. [Último acceso: 06 2021].
- [60] M. Tenorio Melenje y M. A. Trujillo Medina, *ChildDebugging*, Popayán, Colombia: Universidad del Cauca, 2018.
- [61] L. Daniela, R. Strods y I. France, «Activities with Educational Robotics: Research Model and Tools for Evaluation of Progress,» de *Smart Learning with Educational Robotics*, Springer, Cham, 2019, pp. 251-266.
- [62] J. A. Hurtado Alegría, «Toward a Scientific Method in Software Engineering,» 2017. [En línea]. Available: <http://artemisa.unicauca.edu.co/~ahurtado/MCIS.pdf>. [Último acceso: 28 02 2019].
- [63] M. Bunge, *Method, Model and Matter*, Springer Netherlands, 1973.
- [64] M. Shaw, «What Makes Good Research in Software Engineering?,» *International Journal of Software Tools for Technology Transfer*, vol. 4, nº 1, pp. 1-7, 2002.
- [65] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell y A. Wesslén, *Experimentation in Software Engineering*, Springer, 2012.
- [66] J. Ralyte, «Situational method engineering: combining assembly-based and roadmap-driven approaches,» *Requirements Engineering*, vol. 11, nº 1, pp. 58-78, 2006.
- [67] J. Moreno León, G. Robles y M. Román González, «Dr. Scratch: Automatic Analysis of Scratch Projects to Assess and Foster Computational Thinking,» *RED - Revista de Educación a Distancia*, 2015.

- [68] P. Tosato y M. Banzato, «Gender Difference in Handmade Robotics for Children,» de *Tomorrow's Learning: Involving Everyone. Learning with and about Technologies and Computing - 11th IFIP TC 3 World Conference on Computers in Education*, Dublín, Irlanda, 2017.
- [69] M. L. Wu, «Educational Game Design as Gateway for Operationalizing Computational Thinking Skills among Middle School Students,» *International Education Studies*, vol. 11, nº 4, pp. 15-28, 2018.
- [70] A. Pugnali, A. Sullivan y U. M. Bers, «The Impact of User Interface on Young Children's Computational Thinking,» vol. 16, pp. 171-193, 2017.
- [71] H. L. O'Brien y E. G. Toms, «The development and evaluation of a survey to measure user,» *Journal of the American Society for Information Science & Technology*, pp. 50-69, 2010.
- [72] P. Runeson, M. Host, A. Rainer y B. Regnell, *Case Study Research in Software Engineering - Guidelines and Examples*, 2012.
- [73] S. T. Cruz Sánchez y Ó. E. Rojas Ordóñez, *Un modelo para la enseñanza de la programación de software en niños a través de estrategias colaborativas*, Popayán, Colombia: Universidad del Cauca, 2013.