



Revista de Investigación en Tecnologías de la Información

ISSN: 2387-0893

editorial@riti.es

Universitat Politècnica de Catalunya

España

Inda Figueroa, Paulina; Miranda Silva, Daniela; Rangel Meza, Yahely; Peraza Polanco, Gustavo

Potenciando el aprendizaje de programación en estudiantes de educación básica: explorando aplicaciones innovadoras

Revista de Investigación en Tecnologías de la Información, vol. 12, núm. 25, 2024, enero-junio, pp. 11-25

Universitat Politècnica de Catalunya

España

DOI: <https://doi.org/10.36825/RITI.12.25.002>

- Número completo
- Más información del artículo
- Página de la revista en portal.amelica.org





Potenciando el aprendizaje de programación en estudiantes de educación básica: explorando aplicaciones innovadoras

Empowering programming learning in elementary school students: exploring innovative applications

Paulina Inda Figueroa

Universidad Autónoma de Sinaloa, Mazatlán, México

7paulinainda@gmail.com

ORCID: 0000-0001-8028-8187

Daniela Miranda Silva

Universidad Autónoma de Sinaloa, Mazatlán, México

llmiranda24ll@gmail.com

Yahely Rangel Meza

Universidad Autónoma de Sinaloa, Mazatlán, México

6yahelyrangel@gmail.com

Gustavo Peraza Polanco

Universidad Autónoma de Sinaloa, Mazatlán, México

gustavoep1234@gmail.com

ORCID: 0000-0003-4435-6060

doi: <https://doi.org/10.36825/RITI.12.25.002>

Recibido: Diciembre 20, 2023

Aceptado: Abril 15, 2024

Resumen: El objetivo de esta investigación documental es analizar y discutir la literatura de investigación publicada de 2018 a 2022 y aplicaciones comerciales encontradas en las tiendas virtuales de Android y iOS, liberadas de 2013 a 2021, orientadas al desarrollo del aprendizaje de la programación en estudiantes de educación básica. Se utilizó un enfoque cualitativo, de nivel exploratorio y método documental. Como principal resultado se presenta una clasificación de las aplicaciones seleccionadas, con nombre de la aplicación y del desarrollador y propósito. Además de una tabla con informes de desarrollo de aplicaciones, con título de la publicación, país y año. Después de analizar las publicaciones y aplicaciones seleccionadas, se concluye que la estimulación del aprendizaje de la programación en educación básica se estimula mediante herramientas interactivas que resuelven problemas orientados a la lógica de la programación.

Palabras clave: Programación Educativa, Aplicaciones Móviles, Aplicaciones de Programación, Programación Básica, Lógica, Educación Básica.

Abstract: This documentary research aims to analyze and discuss the literature published from 2018 to 2022 and commercial applications found in the Android and iOS virtual stores, released from 2013 to 2021, aimed at developing learning programming for basic education students. A qualitative approach, exploratory level, and

documentary method were used. The main result is a classification of the selected applications, with the application's name, developer, and purpose. In addition to a table with application development reports, publication titles, country, and year. After analyzing the selected publications and applications, it is concluded that programming learning in basic education is stimulated by interactive tools that solve problems oriented to programming logic.

Keywords: *Educational Programming, Mobile Applications, Programming Applications, Basic Programming, Logic, Basic Education.*

1. Introducción

Las aplicaciones móviles son programas diseñados para ejecutarse en dispositivos móviles como teléfonos inteligentes o tabletas. Estas aplicaciones pueden ser desarrolladas tanto para Android como para iOS y están disponibles para su descarga en las tiendas de aplicaciones, como *Google Play Store* para Android y *App Store* para iOS, que albergan millones de aplicaciones de diversas categorías. El año 2008 marca un hito en la historia de las aplicaciones, con la introducción inicial de Apple seguida poco después por Android. Este evento marcó el comienzo de un rápido crecimiento y evolución del mercado de aplicaciones, brindando a los desarrolladores la oportunidad de incursionar en el ámbito digital y ofrecer nuevas soluciones a los usuarios [1].

En 2015, se lanzó la aplicación *ScratchJr*, un lenguaje de programación visual derivado de *Scratch*, con el objetivo de introducir habilidades de codificación a niños de entre cinco y siete años. Esta plataforma permite a los niños crear proyectos y desarrollar habilidades creativas y de razonamiento lógico asociadas con la programación. Los creadores identificaron la necesidad de una herramienta que ofreciera una forma simplificada de codificar desde una edad temprana, sin requerir habilidades de lectura. Al ingresar al mundo de la programación, los niños experimentan un cambio en su percepción, ya que su aprendizaje se adapta a sus capacidades y necesidades individuales. Al ser herramientas lúdicas, los niños tienen la oportunidad de interactuar con ellas y modificarlas según sus preferencias [2].

En la educación actual, la implementación de tecnologías como las Tecnologías de la Información y Comunicación (TIC), Tecnologías de Aprendizaje y Conocimiento (TAC), y Tecnologías de Empoderamiento y Participación (TEP) proporciona una oportunidad para comprender el funcionamiento y la lógica subyacente de la programación. Estas herramientas facilitan el acceso a la codificación al encapsular la lógica de programación de manera técnica y comprensible para los jóvenes estudiantes. Este enfoque busca aumentar el interés en la programación al proporcionar a los estudiantes bases sólidas para comprender sus principios fundamentales. Al hacerlo, se busca superar el estigma de que solo unos pocos son capaces de programar, ya que se reconoce que todos pueden adquirir habilidades en esta área de la informática, especialmente cuando se enseña de manera interactiva. La interacción se presenta como la metodología principal de aprendizaje en este contexto. A través del entretenimiento digital, especialmente si las herramientas son atractivas y lúdicas, se puede captar el interés de los estudiantes y cumplir con los objetivos educativos planteados.

Investigaciones recientes sugieren que presentar a los estudiantes actividades cotidianas de manera algorítmica puede ser beneficioso para el desarrollo del pensamiento lógico. A través de estas actividades, se exploran diversas áreas del conocimiento, permitiendo a los estudiantes probar límites, experimentar con hipótesis alternativas y generar nuevas ideas. En este contexto, la programación se erige como una estrategia pedagógica efectiva para estimular la motivación, la innovación y la capacidad de investigación en los infantes, reviviendo así el innato deseo humano de pensar y crear [2]. Esto subraya la importancia de integrar la programación en el currículo escolar, especialmente en la educación básica, donde se considera una materia indispensable en los últimos años. Esta inclusión educativa proporciona a los niños y adolescentes la oportunidad de comprender de manera interactiva los fundamentos de la programación, permitiéndoles explorar este campo de conocimiento de manera más profunda y significativa.

En los siguientes apartados se presentarán publicaciones previas realizadas por otros investigadores. Estas abarcan revisiones sistemáticas de literatura, estados del arte, mapeos sistemáticos, *reviews* o encuestas sobre el campo de investigación tratado. A continuación, se describe la metodología utilizada en esta revisión sistemática de literatura. Posteriormente, se incluye una tabla de clasificación de los estudios analizados, presentada como resultados. Finalmente, se exponen las conclusiones derivadas de este trabajo de revisión.

2. Investigaciones previas

Se han publicado informes de investigación que abordan el aprendizaje de la lógica de programación a través de aplicaciones móviles interactivas. Este enfoque se ha vuelto cada vez más relevante en el contexto educativo, ya que las tecnologías móviles se han integrado ampliamente en la vida cotidiana de las personas, incluidos los estudiantes. En consecuencia, se ha generado un interés creciente en comprender cómo estas herramientas pueden utilizarse de manera efectiva para facilitar la adquisición de habilidades de programación desde edades tempranas.

La importancia de investigaciones previas sobre este tema radica en su capacidad para proporcionar *insights* valiosos sobre las mejores prácticas, los desafíos comunes y las oportunidades de mejora en el diseño y la implementación de aplicaciones móviles destinadas a enseñar programación. Al realizar una revisión exhaustiva del estado del arte en este campo, es posible identificar tendencias emergentes, identificar lagunas en el conocimiento y evaluar el impacto de intervenciones específicas en el aprendizaje de los estudiantes.

El objetivo del proyecto de investigación mencionado en [2] es analizar el impacto de la aplicación móvil "Bit by bit" en el desarrollo de habilidades lógico-matemáticas en niños de primaria. Este proyecto empleó un enfoque de investigación cuali-cuantitativo, con un diseño no experimental y un muestreo no probabilístico por conveniencia, contando con la participación de un total de 67 niños de entre cinco y seis años. Por otro lado, en la investigación llevada a cabo por los autores en [3], se examinaron las principales iniciativas realizadas en las escuelas en relación con el pensamiento computacional. Se analizó el uso de herramientas específicas, como los kits de robótica o los entornos de programación educativa, así como las estrategias de enseñanza y aprendizaje utilizadas en la educación infantil.

Asimismo, el estudio presentado en [4] tuvo como objetivo emplear aplicaciones móviles como recursos educativos para enseñar a niños menores de ocho años a programar y desarrollar sus habilidades de pensamiento computacional. En [5], el equipo detrás de *ScratchJr* comenzó a utilizar la herramienta *Google Analytics* para obtener una visión más profunda del comportamiento del usuario y a investigar los datos de análisis de aprendizaje que podrían explicar el pensamiento computacional en la infancia. Además, se proporcionó una descripción general de *ScratchJr* y las ideas informáticas que busca enseñar. Por otro lado, los autores en [6] ofrecieron una descripción completa de *ScratchJr* y las ideas informáticas que busca inculcar, así como un análisis de datos que revela tendencias de uso en Europa y comparaciones con el resto del mundo, destacando un alto uso en países con sólidas iniciativas informáticas como el Reino Unido y los países nórdicos.

El propósito planteado en [7] fue establecer un estándar de Ciencias de la Computación que fomente el interés de los niños pequeños en la programación y les proporcione ideas poderosas en esta área desde temprana edad. Por último, en [8] se presentó la introducción de técnicas de aprendizaje basadas en juegos para niños pequeños en Cabo Verde, utilizando la aplicación Code Karts, mientras que [9] introdujo un sistema web para generar juegos de programación para niños de primaria con el objetivo de enseñar conceptos básicos de programación. En [9], se presenta un sistema web para generar juegos de programación para niños de primaria, con el propósito de enseñar conceptos básicos de programación. Además, se analizan cinco juegos existentes con el mismo objetivo, que utilizan lenguajes de programación basados en bloques o basados en texto.

El objetivo general del estudio presentado en [10] fue explorar cómo los niños pequeños interactúan con dos juegos de codificación (Daisy the Dinosaur y Kodable) en una configuración de niño a tableta 1:1. Se encontró que el tipo de juego influía en las diferentes formas en que los niños se comportaban mientras jugaban. Por otro lado, la investigación realizada en [11] mostró que los participantes mejoraron su conocimiento de los comandos de Daisy (moverse, crecer, saltar) y el juego de Kodable (colocar flechas en la secuencia correcta para mover un personaje a través de un laberinto), pero no mejoraron su capacidad para explicar verbalmente qué es codificar. En [12], se propuso que, debido al papel fundamental del lenguaje y la alfabetización en los primeros años, la enseñanza de las ciencias de la computación puede complementarse con modelos de enseñanza de la alfabetización.

El propósito principal en [13] fue mejorar la comprensión de los niños mayores de 4 años en la Escuela Infantil Juana Alarco de Dammert en Trujillo en cuanto al uso de las vocales y los números. Este objetivo se logró mediante el uso de una aplicación móvil educativa que fue creada utilizando la plataforma de desarrollo de la unidad, Mono Desarrollo, Android Studio y Vuforia. La aplicación se desarrolló utilizando el lenguaje de programación C# y se basó en la metodología de desarrollo de programación extrema de *software*. En [14], se proporcionó información útil a los docentes interesados en enseñar habilidades de pensamiento computacional a sus alumnos, presentando diversas herramientas de software, lenguajes y entornos de programación que pueden

ser utilizados en el aula. El estudio incluyó información detallada sobre catorce herramientas de *software*, particularmente útiles para novatos en el tema de programación.

El objetivo de [15] fue encontrar y clasificar los diferentes enfoques y métodos para promover y aprender habilidades de codificación, así como analizar las iniciativas de codificación en Finlandia desde un punto de vista histórico y actual. Por último, en [16], se propuso un análisis de los antecedentes de la robótica, centrándose específicamente en la robótica educativa, examinando teorías didácticas relevantes a este tema, aspectos técnicos de la programación y el pensamiento computacional, y abordando la situación actual de la robótica educativa.

3. Metodología

La presente investigación adopta un enfoque cualitativo de nivel exploratorio y un diseño de campo, con el objetivo de explorar a fondo el uso de aplicaciones móviles en el aprendizaje de la programación. Se optó por el método documental para recopilar información relevante sobre este tema, aprovechando tanto investigaciones empíricas previas como las aplicaciones disponibles en las plataformas *Play Store* y *App Store*.

Para llevar a cabo este estudio, se diseñaron tres fases claramente definidas: la fase de planificación, la fase de ejecución y la fase de informe. En la fase inicial de planificación, se estableció un riguroso proceso de búsqueda de artículos utilizando *Google Académico*, priorizando las publicaciones de revistas especializadas y los repositorios universitarios reconocidos en el campo de la educación y la tecnología. Además de esta búsqueda bibliográfica exhaustiva, se definieron detalladamente los criterios de inclusión y exclusión que se aplicarían en la selección de los estudios relevantes, lo que garantizó la coherencia y la objetividad en la selección de los recursos utilizados en el análisis (ver Tabla 1).

Este enfoque metodológico proporciona un marco sólido para abordar la investigación de manera sistemática y exhaustiva, permitiendo una evaluación crítica de la literatura existente y una recopilación precisa de datos pertinentes para alcanzar los objetivos planteados. La división en fases claramente definidas facilita la gestión eficiente del proyecto y garantiza la coherencia en la recopilación y el análisis de datos.

Tabla 1. Criterios de inclusión y exclusión.

Inclusión	Exclusión
○ Investigaciones empíricas sobre el aprendizaje de la programación en aplicaciones móviles ○ Artículos, capítulos de libros que contengan investigación empírica ○ Publicaciones de revistas desde 2018 hasta 2022 ○ Investigaciones escritas en los idiomas español e inglés	○ Otro tipo de tecnologías aplicadas en el aprendizaje de programación. ○ Publicaciones de revistas no indizadas ○ Publicaciones anteriores a 2018 ○ Investigaciones escritas en otros idiomas

En la segunda fase, se procedió a llevar a cabo la búsqueda exhaustiva de publicaciones relevantes, aplicando rigurosamente los criterios predefinidos de inclusión y exclusión. Este proceso permitió identificar inicialmente un total de 160 informes de investigación empírica que cumplieran con los requisitos establecidos. Sin embargo, tras un análisis detallado, se seleccionaron cuidadosamente ocho documentos que mejor se ajustaban al alcance y los objetivos de nuestra investigación. Durante esta etapa, se descartaron documentos duplicados, aquellos considerados irrelevantes para nuestra indagación y aquellos que no proporcionaban evidencia empírica sólida. Además, se excluyeron aquellos informes con más de cinco años de antigüedad, asegurando así la relevancia y actualidad de los datos recopilados.

En la Tabla 2 se presenta un resumen detallado de los documentos seleccionados, donde se registran diversos aspectos clave como el título, el país de origen, el año de publicación, el método de investigación utilizado, los instrumentos empleados y las características de la muestra experimental. Esta recopilación sistemática de datos es fundamental para establecer una base sólida y confiable para nuestro análisis posterior.

Tabla 2. Publicaciones seleccionadas sobre aplicaciones móviles enfocadas al aprendizaje de la programación.

Título	País y año	Método	Instrumento	Muestra experimental
La robótica en la enseñanza de las ciencias en primaria, una experiencia con Bee-Bot [17].	España 2019	Semicuantitativo	Cuestionario	22 estudiantes de segundo de primaria
Effect of Mind Mapping on Creative Thinking of Children in Scratch Visual Programming Education [18].	China 2022	Cuantitativo	Cuestionario	84 estudiantes de quinto grado de primaria
Análisis de entornos de programación para el desarrollo de habilidades del pensamiento computacional y enseñanza de programación a principiante [19].	Perú 2021	Mixto	Observación	34 estudiantes
Incorporación de Pensamiento Computacional en Ingenierías como soporte a la competencia de Desarrollo de Problemas: jugando con Lightbot [20].	Perú 2021	Cuantitativo	Observación	80 estudiantes
Desarrollo de competencias digitales en programación de aplicaciones móviles en estudiantes de noveno grado a través de tres estrategias pedagógicas [21].	Colombia 2020	Mixto	---	20 estudiantes
La escuela en la era digital: smartphones, apps y programación en educación primaria y su repercusión en la competencia mediática del alumnado [22].	España 2018	Mixto	Cuestionario, Entrevista y Grupos de discusión	Grupo de alumnos de sexto de primaria
El desarrollo del pensamiento computacional en educación: valoración basada en una	España 2022	Cualitativo	Cuestionario	119 personas

experiencia con Scratch [23].				
Nuevos retos educativos en la enseñanza superior frente al desafío COVID-19 [24].	España 2021	Cualitativo	Cuestionario	80 estudiantes
Uso de lenguajes de programación para desarrollar el razonamiento lógico matemático en los niños [25].	Ecuador 2019	Mixto	---	---

La Tabla 2 proporciona una visión detallada de los métodos empleados en las publicaciones seleccionadas, destacando la diversidad de enfoques utilizados por los investigadores. Entre las metodologías identificadas, se observa la presencia de dos enfoques cualitativos, una metodología cuantitativa, una semicuantitativa y cuatro enfoques mixtos. Estos métodos se complementaron con una variedad de instrumentos de investigación, tales como cuestionarios, entrevistas, observación, experimentos, grupos focales y estudios de campo, adaptados según las necesidades y objetivos específicos de cada estudio. Además, se llevó a cabo un cuidadoso proceso de muestreo, utilizando métodos apropiados para cada contexto de investigación. Los países donde se desarrollaron estas investigaciones abarcan una amplia gama geográfica, incluyendo España, Perú, Ecuador y Colombia, lo que refleja una perspectiva internacional en el estudio del aprendizaje de la programación.

Asimismo, como parte de este estudio, se realizó una investigación exploratoria exhaustiva para identificar y recopilar información sobre 34 aplicaciones móviles dirigidas al aprendizaje de la programación. Estas aplicaciones fueron seleccionadas de las tiendas virtuales *Play Store* y *App Store*, y se detallan en la Tabla 3, donde se incluyen el nombre de la aplicación, el año de lanzamiento y el nombre de la empresa desarrolladora correspondiente. Este análisis preliminar proporcionó una comprensión más profunda del panorama actual de las aplicaciones móviles disponibles para la enseñanza de la programación, enriqueciendo así nuestro marco de investigación.

Tabla 3. Aplicaciones móviles orientadas al aprendizaje de la programación.

Nombre aplicación	Año lanzamiento	Nombre empresa desarrolladora
Lighbot: Code hour [26].	2013	SpriteBox LLC
CodeSpark Código de niños [27].	2014	CodeSpark
Lightbor Jr: Coding puzzles [28].	2014	SpriteBox LLC
ScratchJr [29].	2015	Scratch Foundation
Blockly para Dash & Dot [30].	2015	Wonder Workshop, inc
SpriteBox: Code Hour [31].	2016	SpriteBox LLC
Robotizen: Kid learn coding Ro [32].	2016	Mage studio
Human resource machine [33].	2016	Tomorrow corporation
Programación para niños [34].	2017	Mestal
Codificación para niños [35].	2017	IDZ
Code Aventures: Programación [36].	2017	Cyborc games
LEGO Boost [37].	2017	LEGO System A/S
SpriteBox coding [38].	2017	SpriteBox Coding
Ciudad Algoritmo: Juego de C [39].	2018	Musteren
PI Learn to Code [40].	2018	Publications international, Ltd
Algorithm City Pro [41].	2018	Musteren
Fox Factory – Código para niños [42].	2019	Studio Pango – Preschool and learning
Botzees Edu [43].	2019	Shangai Putao Technology Co., Ltd

Logic Code: flowchart [44].	2020	ProfeGarro
Programando con dinosaurios [45].	2020	Yateland – Learning games for kids
Code Karts [46].	2020	Edoki Academy
Coding & AI App – PictoBlox [47].	2020	STEMpedia
Code the Robot. Save the cat [48].	2020	Learny Land
While True: learn() [49].	2020	Nival International
Meoweb: The Puzzle Coding Game [50].	2020	Tapps Games
Bee-bot [51].	2020	TTS Group
Juegos motos niños de 8 años [52].	2020	SKIDOS: Cool math games 5,6,7,8,9,10 year old kids
Kids{Coding} skills [53].	2021	AppQuiz
Tynker Junior [54].	2021	Tynker
Code Land – Código para niños [55].	2021	Learny Land
Tynker – Learn to code [56].	2021	Tynker
Ciudad Algoritmo: El codific [57].	2021	Musteren
Programando con Dinosaurios 2 [58].	2021	Yateland – Learning games for kids

4. Resultados

A continuación, se resume la principal contribución de las nueve publicaciones analizadas en la Tabla 2, las cuales fueron seleccionadas luego de aplicar los criterios de inclusión y exclusión definidos la sección tres, Metodología.

En la investigación *La robótica en la enseñanza de las ciencias en primaria, una experiencia con Bee-Bot* [17], se menciona cómo la introducción de la robótica en el proceso de enseñanza ha mejorado el aprendizaje y ha aumentado el interés del alumnado de segundo grado de primaria por los contenidos de ciencias.

En *Effect of Mind Mapping on Creative Thinking of Children in Scratch Visual Programming Education* [18], los resultados muestran que el aprendizaje en el curso Scratch promovió el pensamiento creativo de los niños. La diferencia entre los dos grupos analizados indica que los mapas mentales fueron beneficiosos para mejorar el pensamiento creativo de los estudiantes.

Un cuestionario fue diseñado en *Análisis de entornos de programación para el desarrollo de habilidades del pensamiento computacional y enseñanza de programación a principiante* [19], para conocer la opinión de los estudiantes sobre su experiencia con herramientas de programación como *Code Studio*, *Lightbot*, *Scratch*, *Turtle* con *Python* y *PSInt*. Los resultados indican que la experiencia previa en *Code Studio* fue útil para programar funciones con *Python* y *Turtle*. *Scratch* ayudó a los estudiantes a aprender los conceptos fundamentales de programación, mientras que la mayoría encontró atractivos los programas que pueden desarrollarse con *Turtle*.

En *Incorporación de Pensamiento Computacional en Ingenierías como soporte a la competencia de Desarrollo de Problemas: jugando con Lightbot* [20], se propone utilizar el pensamiento computacional para apoyar el desarrollo de la competencia de resolución de problemas en estudiantes de Ingeniería. Se presenta una propuesta de actividades que utilizan *Lightbot* como herramienta y se divide en tres componentes para potenciar su uso en una o varias sesiones de trabajo. La experiencia lúdica podría ser incorporada como una herramienta de soporte para reflexionar sobre los aspectos implicados en la resolución de problemas.

El estudio *Desarrollo de competencias digitales en programación de aplicaciones móviles en estudiantes de noveno grado a través de tres estrategias pedagógicas* [21], se llevó a cabo con un enfoque mixto de tipo cuasiexperimental, con 20 estudiantes de grado noveno, para desarrollar siete competencias computacionales a través de tres estrategias pedagógicas: Moodle, página web y *app* inventor. Como resultado, los estudiantes aprendieron a programar aplicaciones móviles y desarrollaron habilidades digitales, liderazgo e innovación. El estudio destaca la necesidad de seguir utilizando las TIC como herramientas pedagógicas, ya que motivan a los estudiantes a aprender y facilitan la enseñanza para los docentes.

En *La escuela en la era digital: smartphones, apps y programación en educación primaria y su repercusión en la competencia mediática del alumnado* [22], los resultados indican que los estudiantes de nivel primaria son capaces de llevar a cabo diversas tareas utilizando las TIC, y que los diferentes agentes educativos valoran positivamente los proyectos que se desarrollan en la escuela.

En *El desarrollo del pensamiento computacional en educación: valoración basada en una experiencia con Scratch* [23] afirman que la experimentación con *Scratch* ha llevado al profesorado en formación a comprender la necesidad de fomentar el pensamiento computacional y sus habilidades en el aula. Por lo tanto, es importante seguir desarrollando modelos educativos que integren la enseñanza de la programación en las aulas universitarias para asegurar que los estudiantes sean activos, creativos y reflexivos en el mundo digital del siglo XXI.

Nuevos retos educativos en la enseñanza superior frente al desafío COVID-19 [24], es una propuesta didáctica que tiene como objetivo conocer la percepción de los futuros maestros de educación infantil sobre *ScratchJr* como herramienta educativa de PC y de competencia digital docente. Los resultados indican que los futuros docentes perciben positivamente la utilidad de *ScratchJr* para diseñar propuestas didácticas y expresan la intención de utilizarla en su enseñanza en el futuro. Además, se destaca la importancia de que los docentes sean capaces de incorporar herramientas tecnológicas en los procesos de enseñanza y aprendizaje.

El objetivo de *Uso de lenguajes de programación para desarrollar el razonamiento lógico matemático en los niños* [25], fue analizar el impacto del lenguaje de programación utilizado en Code.org en el desarrollo del razonamiento lógico-matemático en niños de nivel medio de educación básica, en Ecuador. Los resultados mostraron que la enseñanza del lenguaje de programación en Code.org mejora la capacidad lógico-matemática de los estudiantes en general y en cada uno de los indicadores estudiados.

Las investigaciones antes mencionadas dan a conocer un listado de aplicaciones que explican cómo el alumno de nivel básico se desarrolla con el uso de las aplicaciones móviles en el entorno educativo. Afirmando que el uso de las TIC es esencial en esa etapa educativa, donde a los alumnos se les facilitará el aprendizaje del pensamiento lógico y de programación mediante métodos lúdicos. También, en uno de los estudios se plantea que los profesores deberían utilizar estos métodos de aprendizaje para la enseñanza y aprendizaje en sí mismos.

Por otra parte, con base en la Tabla 3, sobre las aplicaciones móviles enfocadas en el aprendizaje de la lógica matemática y de programación, del cual se llevó a cabo una investigación exploratoria en las tiendas virtuales de *Google play* y *App store*, y también se analizaron las investigaciones previas, con el propósito de determinar si las aplicaciones ahí mencionadas, pudieran pasar a engrosar esta lista, que suma un total de 33 aplicaciones clasificadas como *Aprobado*, *Punto de corte* y *Descartado*, opciones mostradas en la Tabla 4. Aprobado significa que cumplen su objetivo, su funcionalidad, si está disponible y gratuito. La categoría Punto de corte, son aplicaciones que se encuentran en un punto intermedio, es decir, tiene criterios tanto para ser Aprobado o Descartado; por ejemplo, tener versión de prueba gratuita en la que se puede acceder a algunos cursos o niveles dentro de la aplicación, se encuentran anuncios o el diseño no es lo convencional ni llamativo que se busca. Por último, está el atributo Descartado, donde se clasificó de acuerdo a si la aplicación es de paga o presenta problemas de interfaz. Cabe mencionar que se descartaron aplicaciones que no fueron encontradas o eran aplicaciones web.

Tabla 4. Aplicaciones móviles orientadas al aprendizaje de la programación.

Nombre	Desarrolladores	Propósito de la aplicación
A P R O B A D O		
Scratch Jr [29]	Scratch Foundation.	Es un lenguaje de programación de iniciación que permite a niños entre cinco y siete años crear sus propias historias y juegos interactivos encajando bloques de programación gráfica con los que harán que sus personajes se muevan, salten, bailen o canten.
Programación para niños / Kids {Coding} skills [53]	AppQuiz.	Su propósito es conocer los fundamentos básicos de la programación, como la ejecución secuencial, los bucles y las funciones de la forma más fácil y efectiva. Además, los niños podrán mejorar sus habilidades para resolver problemas, desarrollar el pensamiento lógico y estimular la memoria.
SpriteBox: Code Hour [31]	SpriteBox LLC	SpriteBox es un rompecabezas de plataformas único; una mezcla de exploración y el aprendizaje de código.
Robotizen: Kid leran coding Ro [32]	Mage studio	El objetivo es aprender a programar el código para los niños de cuatro a nueve años. Enseña los fundamentos de la programación informática y la robótica a través de una variedad de actividades de aprendizaje interactivo en una historia interesante. Ayudaría a los niños a mejorar su

		resolución de problemas, el pensamiento lógico y la creatividad.
Ciudad Algoritmo: El codific [57]	Musteren	Enseña a los niños los principios fundamentales de la programación y los algoritmos de una forma divertida e innovadora. A través de una serie de juegos con animales, los niños aprenden los conceptos básicos de programación, como la secuencia de comandos, las funciones y los bucles, y deben guiar a un personaje a través de la recopilación de oro y la superación de niveles.
Bee-bot [51]	TTS Group	Se sirve de la funcionalidad principal de Bee-Bot para ayudar a los niños a mejorar sus habilidades en el lenguaje direccional y programar secuencias de giros de 90 grados.
Meoweb: The Puzzle Coding Game [50]	Tapps Games	Aprende CSS mientras ayudas a Agatha a resolver divertidos desafíos de juegos de rompecabezas de plataformas para ayudar a salvar a los gatitos perdidos en apuros.
Lighbot: Code hour [26]	SpriteBox LLC	Juego de puzzle que incorpora conceptos de programación; al utilizarlo, los usuarios aprenden de manera práctica conceptos fundamentales como la secuenciación de instrucciones, procedimientos y bucles, mientras guían a un robot para iluminar azulejos y resolver niveles.
PI Learn to Code [40]	Publications international, Ltd	Una aplicación de codificación donde los niños diseñan y codifican historias animadas. Los niños diseñan personajes, crean escenas de fondo, agregan animación y convierten sus ideas en historias animadas completas con diálogo y efectos de sonido. Creador de personajes para hacer sprites animados. Creador de fondos para hacer mosaicos y dibujar escenas. La sección de codificación tiene comandos de arrastrar y soltar para mover, decir, emocionar y más. Reproduce historias en una linda interfaz de "sala de cine".

PUNTO DE CORTE

Codificación para niños [35]	IDZ	Un divertido juego de codificación para enseñar a los niños los conceptos básicos de la programación, una habilidad muy esencial en el mundo actual. La codificación ayuda a los niños a mejorar sus habilidades de resolución de problemas, aumenta la memoria y aumenta las habilidades de pensamiento lógico.
Programación para niños [34]	Mestal	Arrastra y suelta movimientos a las listas y presiona el botón de inicio. El robot se moverá de acuerdo con los movimientos listados y tomará las baterías.
Tynker – Learn to code [56]	Tynker	Es un sistema de aprendizaje completo para que los niños aprendan a codificar. Los niños comienzan a experimentar con bloques visuales, luego progresan a JavaScript, Swift y Python mientras diseñan juegos, crean aplicaciones y realizan proyectos increíbles.
Logic Code: flowchart [44]	ProfeGarro	Proporciona una forma divertida de aprender acerca del uso de los diagramas de flujo en la programación. El juego también ayuda a fortalecer las habilidades de pensamiento computacional, como el pensamiento algorítmico, la identificación de patrones, la abstracción y la descomposición.
Tynker Junior [54]	Tynker	Tynker Junior se inspiró en el galardonado lenguaje de programación Tynker (tynker.com), utilizado por 60 millones de niños y en más de 90.000 escuelas en todo el mundo. El lenguaje gráfico y la interfaz de usuario se han

		rediseñado para que sea más fácil para los prelectores, con bloques de imágenes sin palabras, una interfaz basada en toques, voces en off amigables, sugerencias útiles, así como una progresión suave de dificultad para motivar la finalización.
--	--	--

DESCARTADO

Ciudad Algoritmo: Juego de C [39]	Musteren	El juego es una herramienta educativa que utiliza juegos divertidos con animales para enseñar a los niños los fundamentos de la programación y algoritmos.
Code Land – Código para niños [55]	Learny Land	Dirigida a niños y niñas de 4 a 10 años y les enseña a programar de una manera visual y divertida, utilizando juegos para adquirir habilidades fundamentales del siglo XXI como la ciencia, programación, lógica y algoritmos.
CodeSpark Código de niños [27]	CodeSpark	Aprendizaje de programación para niños de 5 a 10 años, con múltiples actividades y juegos que tienen como objetivo enseñar los fundamentos de la informática y acercarlos al mundo de STEM.
Code Aventures: Programación [36]	Cyborc games	Sirve para motivar y fomentar en los niños el interés por la codificación y la ciencia. Desarrollado en colaboración con educadores y probado en escuelas, lo que lo hace efectivo para enseñar no solo los fundamentos de la programación, sino también otras habilidades valiosas como el pensamiento lógico, la resolución de problemas, la paciencia, la persistencia y la confianza en sí mismos.
Programando con dinosaurios [45]	Yateland - Learning Games For Kids	Los patrones lógicos simples en los bloques facilitan la programación incluso para los niños que aún no saben leer. El programa ayuda a comprender los conceptos básicos de la programación y desarrollar habilidades de pensamiento lógico y resolución de problemas informáticos.
Code Karts [46]	EDOKI ACADEMY	Esta aplicación presenta una serie de puzzles lógicos en forma de carrera, diseñados para introducir a los niños a partir de los 4 años en el mundo de la preprogramación.
Lightbot Jr: Coding puzzles [28]	SpriteBox LLC	Creado para niños de 4 a 8 años, es una versión simplificada del popular juego <i>Lightbot</i> . Tiene un nivel de dificultad más bajo y puzzles menos complejos. Los niños pueden elegir jugar como <i>Boybot</i> o <i>Girlbot</i> y deben resolver puzzles basados en conceptos de Ciencias de la Computación.
Code the Robot. Save the cat [48]	Learny Land	Es un juego educativo y divertido para aprender los conceptos básicos de programación y lógica. Explora las acciones, los bucles, las funciones y las condiciones.
While True: learn() [49]	Nival International	El juego es una combinación de ingenio y simulación que aborda temas complejos como el aprendizaje automático, redes neuronales, macrodatos e inteligencia artificial, todo esto en el contexto de comprender a tu gato.
Human resource machine [33]	Tomorrow corporation	Rompecabezas dirigido a personas interesadas en la informática. En cada nivel, el jugador recibe una tarea de su jefe y debe resolverla automatizando la solución mediante la programación de un asistente virtual. Si se completa la tarea, se avanza al siguiente nivel y se puede seguir trabajando un año más en el gran edificio de oficinas.

SpriteBox coding [38]	SpriteBox LLC	Aprender a código desde cero, comenzando con el icono de codificación y luego avanzar lentamente a la utilización de la sintaxis Java real.
Coding & AI App – PictoBlox [47]	STEMpedia 2020	Es una aplicación de codificación educativa basada en bloques para principiantes con capacidades mejoradas de interacción de hardware y tecnologías emergentes como robótica, IA y aprendizaje automático que hacen que aprender a codificar sea divertido y atractivo.
LEGO Boost [37]	LEGO System A/S	Mejora habilidades de programación jugando y superando desafíos propuestos por la aplicación. Conforme se avanza, desbloquea niveles y bloques de desarrollo más complejos para seguir mejorando. Además, es posible incorporar BOOST a los modelos y creaciones.
Blockly para Dash & Dot [30]	Wonder Workshop, inc	Es una herramienta visual de programación tipo Drag-and-Drop desarrollada por Google que permite que los niños conecten elementos como piezas de rompecabezas.
Programando con Dinosaurios 2 [58]	Yateland - Learning Games For Kids	No hay que escribir códigos; los niños pueden programar simplemente arrastrando los bloques. Con iconos dinámicos y bien diseñados, sin texto, fáciles para que los niños los entiendan
Juegos motos niños de 8 años [52]	SKIDOS: Cool Math Games 5,6,7,8,9,10 Year Old Kids	Las aplicaciones SKIDOS están diseñados como juegos divertidos que a los niños les encanta jugar. Luego, integramos el contenido de aprendizaje interactivo.
Fox Factory – Código para niños [42]	Studio Pango - Preschool apps and Learning games	Crea combinaciones de bloques con instrucciones simples como cortar, girar, pintar, pegar, entre otros, para diseñar líneas de producción que generen las piezas necesarias para reparar tangrams en el juego.
Algorithm City Pro [41]	Musteren	Aplicación de codificación que utiliza juegos divertidos con animales para enseñar a los niños los conceptos básicos de programación y algoritmos, lo que hace que el aprendizaje sea divertido e innovador.
Botzees Edu [43]	Shanghai Putao Technology Co., Ltd.	Botzees incluye más de 60 niveles de programación interactivos que inspiran a los estudiantes a explorar conceptos básicos de codificación como secuenciación, bucles y condiciones a través de un lenguaje de programación visual e intuitivo.

Con base en la información de la Tabla 4, se resume que el total de aplicaciones fueron divididas en tres categorías: Aprobado, que son las más importantes y que cumplen con su propósito, relacionadas al tema de la programación, que es donde entra la lógica-matemática, con un total de 9 aplicaciones. Después está las catalogadas como Punto de corte, que se refieren a las aplicaciones que tienen aspectos positivos y negativos; son de un total de 5. Por último, está las clasificadas como Descartado, y que son un total de 19 aplicaciones, las cuales son de pago, tienen una interfaz que no va de acuerdo con el objetivo que pretenden cubrir. Es importante decir que se dejó de lado las aplicaciones mencionadas en investigaciones previas que están enfocadas a ejecutarse en la web; también se descartaron otras de las tiendas virtuales, que no estaban enfocadas en niños o su propósito no cubre el objetivo de este trabajo.

En cuanto a su ambiente gráfico se observa en la Figura 1 que *ScratchJr* tiene una interfaz con opciones de crear, editar al personaje y el fondo. La opción se refiere al aprendizaje por bloques, donde se dan instrucciones para enviar mensajes, movimiento, sonido, ciclos de tiempo y repetición. Esto para la creación de escenarios de videojuegos poniendo en práctica la lógica. Hay que destacar que, así como *ScratchJr*, se encuentran otras aplicaciones similares en interfaz y lógica, como *Tynker*, *TynkerJr*, *PI learn to code*, *bee-bot*, entre otras.



Figura 1. Captura desde la aplicación móvil Scratch Jr.

También, están las aplicaciones del tipo *puzles* (rompecabezas), que incrementan su nivel conforme se avanza en el juego, hasta llegar a etapas con bloques de instrucciones como en *Scratch Jr*. Por mencionar a algunos: Programación para niños / *Kids {Coding} skills*, Codificación para niños, *Lighbot: Code hour* (Figura 2), Programación para niños, *SpriteBox: Code Hour Robotizen: Kid leran coding Ro*, *Logic Code: flowchart* y Ciudad Algoritmo: El *codific*.

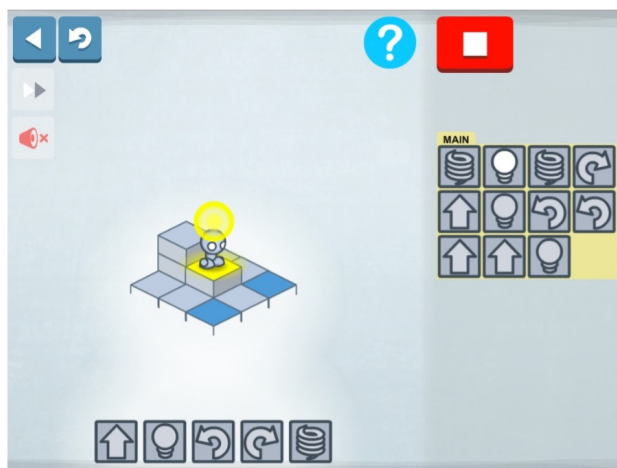


Figura 2. Captura desde la aplicación móvil *Code hour*.

Todas estas aplicaciones por similitudes contienen casi los mismos procedimientos en donde en cada nivel se les enseña un concepto básico de la programación, empezando primero por lo sencillo como sería ejecutar *Hello world*, luego se enseña las secuencias, los bucles y por finalizar las funciones en donde se pone en práctica la recursividad, todo esto ejecutándolo en un *main*, dando a entender los conceptos básicos de la programación.

5. Conclusiones

Es relevante destacar que, durante la búsqueda de las aplicaciones analizadas en investigaciones previas, se encontró un artículo que examinaba la aplicación bit by bit, la cual lamentablemente ya no está disponible en las tiendas virtuales. A pesar de este contratiempo, el propósito de este estudio sigue siendo explorar el aprendizaje lógico de la programación en el ámbito educativo.

Al evaluar las 33 aplicaciones, se pretendía que cumplieran con los criterios mencionados en este trabajo. Sin embargo, este proceso resultó ser tedioso y mecánico, especialmente al determinar la calidad de la interfaz, el correcto funcionamiento y el precio. No obstante, esta tarea permitió a los autores adentrarse en la didáctica de la programación en los niveles básicos de la educación, comprendiendo mejor las necesidades y desafíos de los estudiantes en este campo.

Se observó que la mayoría de las aplicaciones ofrecían membresías opcionales para avanzar en el juego, lo que complicaba el proceso de aprendizaje al introducir elementos de pago. Este aspecto se consideró significativo en la selección final de las aplicaciones, ya que se priorizó aquellas que ofrecían una experiencia de aprendizaje completa sin requerir costos adicionales.

Durante la búsqueda de aplicaciones móviles orientadas a la lógica de la programación para el aprendizaje en estudiantes de educación básica, se llevó a cabo una revisión exhaustiva de investigaciones previas para analizar las aplicaciones mencionadas en esos estudios. Posteriormente, se estableció una conexión con investigaciones empíricas donde se aplican en un contexto exploratorio, enriqueciendo así el análisis con evidencia empírica relevante.

Este proceso combinado de revisión bibliográfica y análisis empírico permitió seleccionar cuidadosamente las aplicaciones recopiladas de las tiendas virtuales. Estas aplicaciones fueron elegidas por su utilidad en el entorno de la programación y su capacidad para ser comprensibles para los alumnos de educación básica. El objetivo final fue analizar y discutir la literatura de investigación publicada de 2018 a 2022 y aplicaciones comerciales encontradas en las tiendas virtuales de Android y iOS, liberadas de 2013 a 2021, orientadas al desarrollo del aprendizaje de la programación en estudiantes de educación básica.

6. Referencias

- [1] Software. (2019). *Aplicaciones móviles: Historia y desarrollo*. Duacode. <https://www.duacode.com/es/blog-noticia/aplicaciones-moviles-desarrollo-historia>
- [2] Sampedro Gualotuña, M. E., Pabón Peralvo, D. M., Analuisa Maiguashca, J. C., Guerrón Varela, E. R. (2022). Programación infantil y desarrollo del ámbito de relaciones lógico-matemáticas en niños de Educación Primaria: Enseñanza con Bit by Bit. *Revista Cognosis*, 7 (1), 39-54. <https://doi.org/10.33936/cognosis.v6i4.3577>
- [3] González-González, C. S. (2019). Estado del arte en la enseñanza del pensamiento computacional y la programación en la etapa infantil. *Education in the Knowledge Society (EKS)*, 20, 1-15. https://doi.org/10.14201/eks2019_20_a17
- [4] Terzopoulos, G., Satratzemi, M., Tsompanoudi, D. (2019). *Educational mobile applications on computational thinking and programming for children under 8 years old*. 13th Internet of Things, Infrastructures and Mobile Applications. Thessaloniki, Greece. https://doi.org/10.1007/978-3-030-49932-7_50
- [5] Leidl, K. D., Bers, M. U., Mihm, C. (2017). *Programming with ScratchJr: a review of the first year of user analytics*. Conference Proceedings of International Conference on Computational Thinking Education. Wanchai, Hong Kong. https://sites.bc.edu/devtech/wp-content/uploads/sites/181/2018/02/Leidl_Bers_Mihm_ScratchJrAnalyticsHongKong.pdf
- [6] Bers, M. U. (2018). Coding and computational thinking in early childhood: The impact of ScratchJr in Europe. *European Journal of STEM Education*, 3 (3), 8. <https://doi.org/10.20897/ejsteme/3868>
- [7] Sullivan, A., Bers, M. U. (2019). Computer science education in early childhood: the case of ScratchJr. *Journal of Information Technology Education. Innovations in Practice*, 18, 113-138. <https://doi.org/10.28945/4437>
- [8] Pombo, N., Lamas, D. (2022). *Game-Based Learning for Young Children: A Case Study*. IEEE Global Engineering Education Conference (EDUCON) Tunis, Tunisia. <https://doi.org/10.1109/EDUCON52537.2022.9766723>
- [9] Erazo-Palacios, J., Jaimez-González, C. R., García-Mendoza, B. (2022). Towards a Web Generator of Programming Games for Primary School Children. *International Journal of Engineering Pedagogy (iJEP)*, 12 (4), 98–114. <https://doi.org/10.3991/ijep.v12i4.17335>
- [10] Lennon, M., Pila, S., Flynn, R., Wartella, E. A. (2022). Young children's social and independent behavior during play with a coding app: Digital game features matter in a 1: 1 child to tablet setting. *Computers & Education*, 190, 1-14. <https://doi.org/10.1016/j.compedu.2022.104608>
- [11] Pila, S., Aladé, F., Sheehan, K. J., Lauricella, A. R., Wartella, E. A. (2019). Learning to code via tablet applications: An evaluation of Daisy the Dinosaur and Kodable as learning tools for young children. *Computers & Education*, 128, 52-62. <https://doi.org/10.1016/j.compedu.2018.09.006>

- [12] Bers, M. U. (2019). Coding as another language: A pedagogical approach for teaching computer science in early childhood. *Journal of Computers in Education*, 6 (4), 499-528. <https://doi.org/10.1007/s40692-019-00147-3>
- [13] Cieza, E., Lujan, D. (2018). Educational mobile application of augmented reality based on markers to improve the learning of vowel usage and numbers for children of a kindergarten in Trujillo. *Procedia computer science*, 130, 352-358. <https://doi.org/10.1016/j.procs.2018.04.051>
- [14] Sarmiento Bolívar, M. I. (2021). Lenguajes y Entornos de Programación para Fortalecer El Desarrollo de Competencias Concernientes al Pensamiento Computacional. *Revista de divulgación científica de la Universidad Alas Peruanas (Hamut'ay)*, 7 (3), 86-97. <http://dx.doi.org/10.21503/hamu.v7i3.2200>
- [15] Tuomi, P., Multisilta, J., Saarikoski, P., Suominen, J. (2017). Coding skills as a success factor for a society. *Education and Information Technolies*, 23, 419–434. <https://doi.org/10.1007/s10639-017-9611-4>
- [16] Ruiz Rey, F. J., Hernández Hernández, P., Cebrian-de-la-Serna, M. (2018). *Programación y robótica educativa: enfoque didáctico-técnico y experiencias de aula*. Jornada de Programación, Robótica e Impresión 3D. Madrid. <https://hdl.handle.net/10630/15784>
- [17] Hurtado Soler, A., Santamaria Pérís, N. (2019). La robótica en la enseñanza de las ciencias en primaria, una experiencia con Bee-Bot. *Creativity and Educational Innovation Review*, (3), 104-119. <https://doi.org/10.7203/CREATIVITY.3.15977>
- [18] Su, Y. S., Shao, M., Zhao, L. (2022). Effect of mind mapping on creative thinking of children in scratch visual programming education. *Journal of Educational Computing Research*, 60 (4), 906-929. <https://doi.org/10.1177/07356331211053383>
- [19] Laura-Ochoa, L., Bedregal-Alpaca, N. (2021). Análisis de entornos de programación para el desarrollo de habilidades del pensamiento computacional y enseñanza de programación a principiantes. *Revista Ibérica de Sistemas e Tecnologías de Informação*, (E43), 533-548. <https://www.proquest.com/scholarly-journals/análisis-de-entornos-programación-para-el/docview/2562271519/se-2>
- [20] Urquiza, G., Vidal, E., Castro, E. (2021). Incorporación de Pensamiento Computacional en Ingenierías como soporte a la competencia de Desarrollo de Problemas: jugando con Lightbot. *Revista Ibérica de Sistemas e Tecnologías de Informação*, (E42), 199-207. <https://www.proquest.com/scholarly-journals/incorporación-de-pensamiento-computacional-en/docview/2493869400/se-2>
- [21] Fonseca Barrera, C. C., Niño Vega, J. A., Fernández Morales, F. H. (2020). Desarrollo de competencias digitales en programación de aplicaciones móviles en estudiantes de noveno grado a través de tres estrategias pedagógicas. *Boletín Redipe*, 9 (4), 179-191. <https://dialnet.unirioja.es/servlet/articulo?codigo=7528414>
- [22] Bonilla-del-Río, M., Aguaded Gómez, J. I. (2018). La escuela en la era digital: smartphones, apps y programación en Educación Primaria y su repercusión en la competencia mediática del alumnado. *Pixel-Bit: Revista de Medios y Educación*, 53, 151-163. <https://doi.org/10.12795/pixelbit.2018.i53.10>
- [23] Gamito, R., Aristizabal, P., Basasoro, M., León, I. (2022). El desarrollo del pensamiento computacional en educación: valoración basada en una experiencia con Scratch. Innoeduca. *International Journal of Technology and Educational Innovation*, 8 (1), 59-74. <https://doi.org/10.24310/innoeduca.2022.v8i1.12093>
- [24] Satorre Cuerda, R. (2021). *Nuevos retos educativos en la enseñanza superior frente al desafío COVID-19*. Editorial Octaedro. <http://hdl.handle.net/10045/119194>
- [25] Aguilar Enríquez, F. D. (2019). Uso de lenguajes de programación para desarrollar el razonamiento lógico matemático en los niños. *Revista Científica UISRAEL*, 6 (2), 64-72. <https://doi.org/10.35290/rcui.v6n2.2019.114>
- [26] SpriteBox LLC. (2013). *Lightbot: Code hour*. <https://lightbot.com/hour-of-code.html>
- [27] CodeSpark .(2014). *CodeSpark: Código de niños* (3.13.00). https://play.google.com/store/apps/details?id=org.codespark.thefoos&hl=es_MX&gl=US
- [28] SpriteBox LLC. (2014). *Lightbot Jr: Coding puzzles*. <https://play.google.com/store/apps/details?id=com.lightbot.lightbotjr&hl=es&gl=US> / <https://apps.apple.com/us/app/lightbot-jr-coding-puzzles-for-ages-4/id858640629>
- [29] Scratch Foundation. (2015). *Scratch Jr*. <https://www.scratchjr.org/>
- [30] Wonder Workshop, Inc. (2015). *Blockly para Dash & Dot*. https://play.google.com/store/apps/details?id=com.makewonder.blockly&hl=es_MX&gl=US
- [31] SpriteBox LLC. (2016). *SpriteBox: Code hour*. <https://spritebox.com/hour.html>

- [32] Mage studio. (2016). *Robotizen: Kid learn coding Ro*. <https://mage.com.vn/robotizen>
- [33] Tomorrow corporation (2016). *Human resource machine*.
<https://tomorrowcorporation.com/humanresourcemachine#getitnow>
- [34] Mestal. (2017). *Programación para niños* (3.2.4).
https://play.google.com/store/search?q=programacion%20para%20ni%C3%B1os&c=apps&hl=es_MX
- [35] IDZ. (2017). *Codificación para niños*. <https://civat.es/app/codificacion-para-ninos/>
- [36] Cyborc games. (2017). *Code Aventures: Programacion* (3.1).
https://play.google.com/store/apps/details?id=com.cyborc.codeadventures&hl=es_MX&gl=US
- [37] LEGO System A/S. (2017). *LEGO Boost*. <https://www.lego.com/es-mx/service/device-guide/boost>
- [38] SpriteBox Coding. (2017). *SpriteBox Coding*. <https://spritebox.com/>
- [39] Musteren. (2018). *Ciudad Algoritmo: Juego de C* (1.0.5).
https://play.google.com/store/apps/details?id=air.MusterenGames.ElHarezmiCoding&hl=es_MX&gl=US
- [40] Publications international, Ltd. (2018). *PI Learn to code*.
<https://play.google.com/store/apps/details?id=com.pubint.codingkit&hl=en&gl=US>
- [41] Musteren. (2018). *Algorithm City Pro* (1.0.2).
https://play.google.com/store/apps/details?id=air.MusterenGames.AlgorithmCityPro&hl=es_MX&gl=US
- [42] Studio Pango – Preschool and learning. (2019). *Fox Factory – Código para niños*.
https://play.google.com/store/apps/details?id=com.studiopango.foxfactory&hl=es_MX&gl=US
- [43] Shangai Putao Technology Co., Ltd (2019). *Botzees edu*.
https://play.google.com/store/apps/details?id=com.pai.tots.edu&hl=es_MX&gl=US
- [44] ProfeGarro. (2020). *Logic Code: flowchart* (1.02).
https://play.google.com/store/apps/details?id=com.profegarro.LogicCode&hl=es_MX&gl=US
- [45] Yateland – Learning games for kids. (2020). *Programando con dinosaurios* (1.0.6).
<https://es.vnmod.net/app/programando-con-dinosaurios-mod-hack/>
- [46] EDOKI ACADEMY. (2020). *Code Karts* (4.2).
https://play.google.com/store/apps/details?id=com.edokiacademy.babycoding&hl=es_MX&gl=US
- [47] STEMpedia. (2020). *Coding & AI App – PictoBlox*. <https://thetempedia.com/product/pictoblox/>
- [48] Learnly Land. (2020). *Code the Robot. Save the cat*. <https://learnlyland.com/apps/>
- [49] Nival International. (2020). *While True: learn()*. https://luden-io.translate.google.wtl/?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es-419&_x_tr_pto=sc
- [50] Tapps Games. (2020). *Meoweb: The Puzzle Coding Game* (1.0).
<https://play.google.com/store/apps/details?id=br.com.tapps.meoweb&hl=en&gl=US>
- [51] TTS Group. (2020). *Bee-bot*. <https://play.google.com/store/apps/details?id=com.tts.beebot&hl=en&gl=US>
- [52] SKIDOS: Cool math games 5,6,7,8,9,10-year-old kids. (2020). *Juegos motos niños de 8 años* (2.0).
<https://play.google.com/store/apps/details?id=skidos.kids.cool.math.bike.racing&pli=1>
- [53] AppQuiz. (2021). *Kids{Coding} skills* (2.9).
<https://play.google.com/store/apps/developer?id=AppQuiz&hl=es&gl=US>
- [54] Tynker. (2021). *Tynker Junior*. <https://www.tynker.com/mobile/apps/tynker-junior>
- [55] Learnly Land. (2021). *Code Land – Código para niños*. <https://learnlyland.com/code-land/>
- [56] Tynker. (2021). *Tynker – Learn to code*. <https://www.tynker.com/mobile/apps/tynker>
- [57] Musteren. (2021). *Ciudad Algoritmo: El codific*. <http://www.musterengames.com/2020/07/19/algorithm-city-coding-game/>
- [58] Yateland – Learning games for kids. (2021). *Programando con Dinosaurios 2*.
<https://play.google.com/store/apps/details?id=com.imayi.dinocode2>