

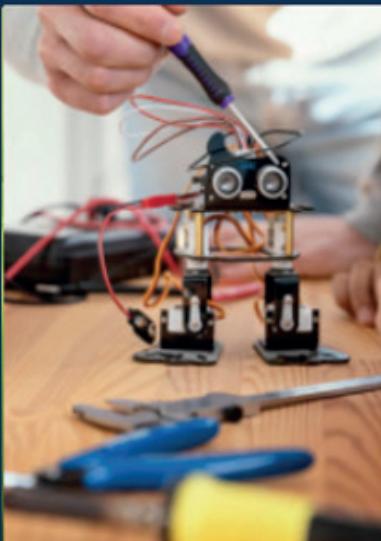


uni

Universidad
Nacional
de Loja

1859

Herramientas digitales para la enseñanza de la programación en el ámbito educativo



Milton Labanda Jaramillo
María Coloma Andrade
Gloria Michay Caraguay



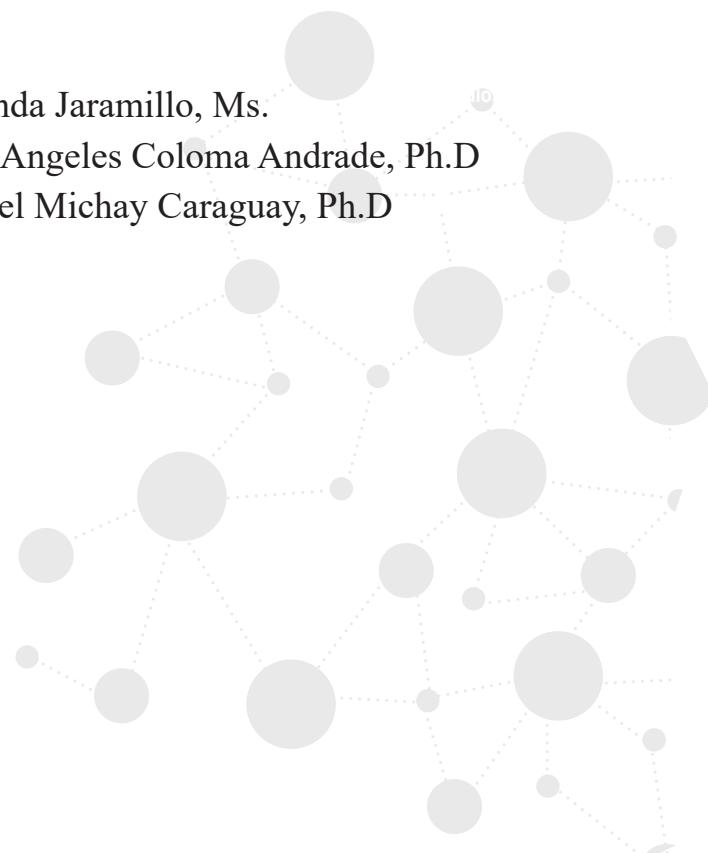
Herramientas digitales para la enseñanza de la programación en el ámbito educativo

Herramientas digitales para la enseñanza de la programación en el ámbito educativo

Milton Labanda Jaramillo, Ms.

María de los Angeles Coloma Andrade, Ph.D

Gloria Cecibel Michay Caraguay, Ph.D



UNL



Universidad
Nacional
de Loja

Nikolay Aguirre, Ph. D.

Rector UNL

Elvia Zhapa Amay, Ph. D.

Vicerrectora académica

Max Encalada, Ph. D.

Director de Investigación

Herramientas digitales para la enseñanza de la programación en el ámbito educativo

Autores:

Milton Labanda Jaramillo, Ms.

María de los Angeles Coloma Andrade, Ph.D

Gloria Cecibel Michay Caraguay, Ph.D

Revisión Par Académico:

Janio Jadán, Ph.D

Rosemary Samaniego, Ph.D

ISBN: 978-9942-671-08-0

Editorial Universitaria

Contacto: comision.editorial@unl.edu.ec

Universidad Nacional de Loja

Tel: +59372593550

Ciudad Universitaria Guillermo Falconí, Loja - Ecuador

www.unl.edu.ec

Noviembre, 2024

Loja, Ecuador



Índice General

Prólogo.....	XIII
Prefacio	XV
Acrónimos.....	XVII
Introducción	19
Capítulo 1.....	21
La programación como fuente motivadora para el desarrollo de habilidades del pensamiento mediante el legado de Seymour Papert	21
Introducción.....	21
La programación.....	22
El pensamiento computacional	24
Elementos del pensamiento computacional.....	25
Fomentar el pensamiento computacional en el aula.....	26
Aprender a programar a través de la acción, el legado de Seymour Papert.....	27
Programación Logo	28
Aprendizaje utilizando la programación FMSLogo	29
Instrucciones de trabajo de FMSLogo.....	30
Experimentación	31
Ejemplo 1.....	31
Ejemplo 2.....	32
Reflexiones	33
Bibliografía	35
Capítulo 2.....	38
El aprendizaje basado en simulación para aprender la programación de computadoras	38
Introducción.....	38
El aprendizaje basado en la simulación	39
Ventajas y desventajas de las simulaciones en el proceso educativo	40
Programación de computadoras.....	42
Programación de computadoras en base a simulaciones	44
Caso práctico: PSeInt.....	45
Reflexiones	49
Bibliografía	51

Capítulo 3.....	54
Herramientas de codigofabetización y Code.org para la enseñanza de la programación	54
Introducción.....	54
La codigofabetización en el siglo XXI	55
Recursos para la enseñanza-aprendizaje del coding.....	56
Code.org.....	65
Reflexiones	68
Bibliografía	70
Capítulo 4.....	74
Enseñanza de la programación orientada a objetos con la herramienta Alice.....	74
Introducción.....	74
Programación.....	74
Paradigmas de Programación.....	76
Paradigmas procedimentales	76
Paradigmas declarativos	77
Programación Pedagógica	77
Entorno de programación Alice	78
¿Cómo se usa Alice?.....	79
Programación orientada a objetos con Alice	81
Herramienta Alice para crear entornos dinámicos de aprendizaje	82
Reflexiones	84
Bibliografía	85
Capítulo 5.....	88
Fundamentos de programación en Scratch para el desarrollo de habilidades cognitivas en niños de Educación primaria	88
Introducción.....	88
La tecnología en la educación.....	89
Fundamentos Generales de la Programación.....	89
Habilidades cognitivas	90
Programación en bloques mediante la herramienta Scratch para el desarrollo de habilidades cognitivas	92
Experiencias de usuarios al utilizar la herramienta de programación Scratch	94
Presentación de la herramienta Scratch	95
Reflexiones	100
Bibliografía	102

Capítulo 6.....	105
mBlock: bases técnicas y metodológicas para iniciar en el mundo de la programación a los estudiantes de Educación Básica.....	105
Introducción.....	105
Programación Orientada a Objetos.....	106
¿Qué es mBlock?	106
Educación STEAM	108
Tecnologías que integra mBlock.....	108
Scratch	108
Robótica Educativa.....	109
Inteligencia Artificial (IA)	109
Python.....	110
Arduino.....	110
Recurso educativo “El rango y combinaciones simples” desarrollado en mblock.....	111
Reflexiones	114
Bibliografía	115
Capítulo 7.....	117
Programación por bloques: Caso de estudio en la herramienta Blockly.....	117
Introducción.....	117
Importancia de programar en el siglo XXI	117
Programación en Educación	119
Programación por bloques	122
Herramientas para programar por bloques	123
Programación por bloques utilizando Blockly Games	125
Reflexiones	131
Bibliografía	133
Capítulo 8.....	136
Tynker y el mundo de la programación basada en bloques en la educación primaria	136
Introducción.....	136
Educación primaria en el contexto ecuatoriano.....	137
Programación basada en bloques.....	138
Tynker	139
Tynker en la educación primaria	142

Reflexión.....	144
Bibliografía	145
Capítulo 9.....	148
Educación en inteligencia artificial y codificación en PictoBlox ...	148
Introducción.....	148
Pensamiento computacional	149
Gamificación.....	150
Metodología STEAM	150
Programación por bloques	151
PictoBlox	152
Reflexiones	156
Bibliografía	157
Capítulo 10	159
Desarrollo de la lógica de programación en edades tempranas con Bee-Boot.....	159
Introducción.....	159
Desarrollo de la lógica en edades tempranas	160
Desarrollo de la lógica en programación	161
Desarrollo de la lógica en programación con Bee-Bot	162
Casos de Estudio y Experiencias Reales con Bee-Boot ..	165
Aplicación digital Bee-Boot.....	166
Reflexión.....	167
Bibliografía	168
Capítulo 11	170
Aprendizaje de la programación a través de la gamificación en la herramienta Grasshopper.....	170
Introducción.....	170
Programación en la educación	171
Gamificación.....	172
Gamificación aplicada a la enseñanza de programación	173
Programando con Juegos.....	174
La herramienta Grasshopper.....	174
Características de la aplicación Grasshopper	175
Estructura de la herramienta Grasshopper	177
Niveles	178

Funciones	179
Experimentación de la herramienta Grasshopper.....	180
Reflexiones	181
Bibliografía	183
Capítulo 12.....	186
SoloLearn: plataforma móvil para aprender a programar en varios lenguajes.....	186
Introducción.....	186
Introducción a las tecnologías móviles.....	187
Clasificación de los dispositivos móviles	187
¿Por qué son necesarias las tecnologías móviles en la educación?	188
Eficiencia en el aprendizaje con tecnología.....	190
Inclusión de la tecnología móvil en el aprendizaje	191
Implementación de la programación móvil con SoloLearn.....	192
Aprende a programar con SoloLearn	192
Métodos de enseñanza.....	193
Cursos disponibles.....	194
Aprendizaje constante	195
Comunidad de SoloLearn	195
Código desarrollado en SoloLearn	196
Reflexiones	200
Bibliografía	201
Capítulo 13.....	203
Experiencia de Enseñanza de Programación con Realidad Aumentada	203
Introducción	203
Enseñanza del lenguaje de programación en la educación	203
Beneficios de la enseñanza de lenguajes de programación	206
Realidad Aumentada en la educación	206
Beneficios de la RA	208
Herramienta Metaverso	209
Reflexiones	210
Bibliografía	212

Prólogo

La investigación es un eje al que toda sociedad debe inclinarse a fin de descubrir nuevos horizontes, siendo así el presente libro es el resultado de la investigación bibliográfica que surge a partir de la asignatura de Programación Web, trabajado colaborativamente entre los estudiantes del ciclo seis y el docente responsable de la asignatura, en la carrera de Pedagogía de las Ciencias Experimentales Informática de la Facultad de la Educación el Arte y la Comunicación de la Universidad Nacional de Loja.

En esta obra se incluyen investigaciones inéditas sobre diferentes herramientas digitales para la enseñanza de la programación en el ámbito educativo, considerando que aprender a programar mejora el razonamiento, facilita el aprendizaje de las matemáticas y la capacidad de resolver problemas, así como también desarrolla competencias, favorece la creatividad y aumenta la confianza y la autoestima.

De esta manera este libro es destinado a todos quienes se dedican al maravilloso mundo de la educación de manera específica a la enseñanza de la programación en el ámbito educativo, consideramos entonces que será posible llegar a mucha gente por la importancia del mismo, la significación del mismo no solo se debe a su riqueza intelectual, sino también a su impacto y el apoyo que brindará de manera directa en la educación.

Mg. Gloria Cecibel Michay Caraguay.

Prefacio

Las tecnologías aplicadas a la educación han influido en los procesos de enseñanza aprendizaje tanto en modalidad presencial, así como en la educación en línea, y también en las formas de comunicación entre los actores educativos. Estos hechos permiten nuevas formas de interacción, que representan una oportunidad de responder a las necesidades que demanda la sociedad, mediante su inclusión en la planificación curricular; definiendo una nueva forma de comunicación donde las audiencias digitales exigen contenidos interactivos capaces de adaptarse a un nuevo patrón de consumo mediático.

En la actualidad los estudios superiores se encuentran con un gran desarrollo para la educación, debido a los magnos avances tecnológicos, esto trae ventajas para los estudiantes y docentes, ya que tienen que acoplar sus actividades a las nuevas tecnologías fomentando la producción del conocimiento sobre su aplicación, empezando a usar herramientas más accesibles y fáciles para lograr relacionarse de una manera más completa. En este contexto la asignatura de Programación Web en el contexto educativo permite fusionar áreas de sistemas cognitivos aplicados a la tecnología educativa que generan un ambiente de aprendizaje alternativo.

El fin aquí, es generar una experiencia de aprendizaje en la cual se usa como modelo el razonamiento humano para resolver problemas y formular diagnósticos, que provee explicaciones a los estudiantes, ya que todos los contenidos están estructurados bajo un banco de lenguajes de programación por bloques y lenguajes de código, existiendo estrategias de instrucción y de resolución asociadas a la capacidad de moldearse al desarrollo del estudiante a través de una enseñanza individual y colaborativa que permiten implementar técnicas y recursos educativos en el sistema de enseñanza/aprendizaje.

Ph.D. María de los Angeles Coloma Andrade.

Acrónimos

CMU	Universidad Carnegie Mellon
CSTA	Asociación de Docentes en Ciencias de la Computación.
EGB	Educación General Básica
IA	Inteligencia Artificial.
IDE	Entorno de Desarrollo Integrado
INTEF	Instituto Nacional de Tecnologías Educativas y Formación del Profesorado
ISDI	Instituto Superior para el Desarrollo de Internet
ISTE	Sociedad Internacional para la Tecnología en Educación
JS	JavaScript
MIT	Instituto de Tecnología de Massachusetts
MLEARNING	Aprendizaje móvil
PC	Pensamiento Computacional
PRISMA	Preferred Reporting Items for Systematic reviews and Meta-Analyses.
POO	Programación Orientada a Objetos.
RA	Realidad Aumentada
RITA	Inventor de robot para enseñar algoritmos
SE	Sistema Embebido.
STEAM	Science, Technology, Engineering Arts & Mathematics
STEM	Science, Technology, Engineering and Mathematics
TIC	Tecnologías de la Información y la Comunicación
UNIR	Universidad Internacional de La Rioja

Introducción

El siglo XXI demanda del ciudadano, una comprensión cada vez más profunda del mundo digital (ciencias de la computación, pensamiento computacional, programación) con el fin de que las generaciones jóvenes se incorporen de forma activa y flexible dentro de la fuerza laboral que opera cada vez más con el apoyo de la tecnología, por ello necesitamos desarrollar nuevas competencias relacionadas con aquello en el entorno escolar. Este escenario nos demanda un cambio de paradigma y de mentalidad en la formación del talento humano para las profesiones del futuro, pasar de reproducir contenidos a generar conocimiento, crear tecnología y aplicar ese conocimiento para resolver problemas reales, pues según el Foro Económico Mundial alrededor de 100 millones de nuevos puestos de empleo divididos entre humanos, máquinas y algoritmos serán creados al 2025, por ello la necesidad de preparar a las nuevas generaciones para afrontar estos retos laborales es inminente.

En este contexto la enseñanza de la programación surge como elemento del cambio de paradigma y debe contemplar entre sus objetivos esenciales, la formación y desarrollo de habilidades por parte de los estudiantes, que posibiliten la resolución de problemas del ámbito escolar, profesional o de la vida práctica, teniendo en cuenta los recursos que brindan los diferentes lenguajes y herramientas de programación. A lo largo de la historia, diferentes tecnologías han posibilitado múltiples aprendizajes ya que la mediatización en el devenir histórico es una característica universal de todas las sociedades humanas, por ello en las últimas décadas se han creado entornos visuales de programación que eximen a los usuarios de responsabilidades otras complejas como el dominar la sintaxis de lenguajes de programación, pudiendo centrarse en la semántica de los programas concretamente.

Esta obra tiene la intención de ampliar el panorama de docentes y estudiantes de escuelas y colegios con respecto a la programación de computadores desde una visión didáctica y técnica. Erigida como uno de los productos de la asignatura denominada Programación Web parte del plan de estudios de la Carrera Pedagogía de las Ciencias Experimentales Informática, busca desde la revisión bibliográfica de un conjunto de lenguajes y plataformas, contribuir a la literatura científica en el campo de la programación de computadores desde un enfoque pedagógico orientado hacia el uso de la programación en las aulas escolares.

Ms. Milton Leonardo Labanda Jaramillo.

Capítulo 1

La programación como fuente motivadora para el desarrollo de habilidades del pensamiento mediante el legado de Seymour Papert

Naranjo Correa Guissella Yazmin, Villamagua Vicente Sandra del Carmen.

Universidad Nacional de Loja.

guissella.naranjo@unl.edu.ec, sandra.villamagua@unl.edu.ec

Introducción

El presente trabajo muestra cómo la programación hoy en día permite construir el conocimiento a través de la implementación de diferentes paradigmas o teorías del aprendizaje, debido a que existen muchos mecanismos y herramientas adaptadas como apoyo a la educación, y su relación con el desarrollo de las habilidades del pensamiento computacional, creativo y crítico. Por ello, al momento de aplicar un lenguaje de programación se debe tener en cuenta la metodología que se va a llevar a cabo durante el proceso dirigido a la enseñanza del estudiante.

En los últimos tiempos ha surgido el interés por la enseñanza de lenguajes computacionales en las escuelas, que emplea el uso de Internet, redes sociales digitales, telefonía móvil y la importancia de las ciencias computacionales, las mismas, han conducido hacia nuevas perspectivas para entender el rol de la programación en la educación. Hoy existe la alternativa de introducir la enseñanza de lenguajes de programación desde la escuela, los mismo que generan impactos cognitivos, que faciliten el desarrollo de habilidades en la resolución de problemas y a su vez promueve el pensamiento lógico.

La programación, comenzó a dar un giro positivo hacia la pedagogía, y surge a inicios de los años 70, cuando se da el primer uso en el ámbito educativo. Seymour Papert llevó a cabo esta propuesta con su lenguaje de programación Logo e introdujo la idea de que la programación a través de la computadora fomenta la capacidad de los niños para pensar y razonar de manera crítica. Además, se puede utilizar en una variedad de escenarios educativos, ya que permite a los estudiantes programar sin necesidad de aprender lenguajes de programación tradicionales.

La programación

La programación cobra relevancia en la enseñanza y aprendizaje de los contenidos que se desea impartir en la educación. “Los estudiantes deben primero desarrollar el pensamiento computacional para luego abordar cursos de programación, la mejor manera de desarrollar el pensamiento computacional es a través del aprendizaje de la programación” (Brugés y Camperos, 2022, p. 5). Por lo tanto, se debe fomentar la enseñanza de la programación como medio en los estudiantes para lograr un aprendizaje continuo mediante el juego y el uso de las tecnologías.

La programación se guía por una serie de normas, instrucciones y expresiones que tienden a ser semejantes a una lengua natural acotada. Por lo cual recibe el nombre de lenguaje de programación. De modo que, en la informática, la programación es pieza clave en la relación entre los ordenadores y los usuarios. La programación es la escritura de programas para computador, mediante los cuales sea capaz de sistematizar la información e ideas ya sea para simular sus interpretaciones e hipótesis o para crear soluciones informáticas (Adell et al., 2019). Así, el principal motivo de la programación es utilizar el ordenador como medio de conocimientos en la ejecución de programas y aplicaciones que sean accesibles para el usuario y poder desempeñar diversas funciones para las cuales fue elaborado.

Aprender a programar puede ser considerado como un gran reto que implica explorar nuevos conocimientos, para Gómez (2020), es similar a aprender un nuevo idioma, es necesario aprender nuevo vocabulario y conceptos, con los cuales interactuar en este caso con el computador o dispositivo tecnológico y comunicar las instrucciones de tal forma que la máquina entienda la información y la procese según los requerimientos establecidos.

La programación para niños al ser una disciplina nueva, no cuenta con grandes estudios de investigación relacionados a su didáctica, por lo cual el autor diseñó, un entorno para favorecer la expresividad en el lenguaje, para detectar errores y proponer alternativas o métodos para adquirir el lenguaje de programación basado en el aprendizaje del lenguaje natural (Gómez, 2020); por lo cual es posible enseñar lenguaje de programación a niños de cualquier edad, incluso si aún no saben leer y escribir.

Por ende, la programación aplicada al proceso educativo permite desarrollar habilidades y crear un formato atractivo de clase, que permitirá un aprendizaje significativo de conceptos teóricos, a la vez que los niños realicen actividades vivenciales (Manches y Ploughma citado en González et al., 2023). Sin embargo, hay algunos aspectos que dificultan su amplia implementación: el currículum actual de educación Infantil no contempla el desarrollo de la competencia digital.

Teniendo en cuenta lo anterior, es importante abordar la programación desde edades tempranas, siendo la Educación Primaria la primera etapa en la que se debe contemplar el enseñar programación (Vera, 2021). De igual manera como menciona Álvarez (2021), no existe una materia específica para profesorado en formación y muchos docentes deben autoformarse; en algunos casos la autoformación genera carencias lo que conlleva a propuestas vacías de contenido, de uso esporádico y con un objetivo lúdico y de entretenimiento.

El pensamiento computacional

La incorporación de la tecnología en los centros educativos continúa ampliando los procesos de enseñanza-aprendizaje y a su vez fortalece los diferentes servicios que se utilizan para la educación, por ello aprender a programar, codificar o escribir un conjunto de instrucciones que realicen una determinada tarea y las ejecute una computadora (Polanco et al., 2021); en consecuencia, permitirá al estudiante el desarrollo de diversas habilidades del pensamiento.

En esta misma línea de investigación acerca de las repercusiones que tiene la programación en el desarrollo de habilidades, Valverde et al. (como se cita en Roig y Moreno, 2020) entiende al pensamiento computación como un concepto complejo en sí mismo, ya que permite que se le relacione con una competencia completa, de un grado de complejidad alto que, al mismo tiempo, se puede relacionar con niveles de pensamiento abstracto, matemático y pragmático aplicados en diferentes momentos de nuestra vida cotidiana.

Asimismo, considerando el aporte que brinda otros autores a lo mencionado con anterioridad, Wing (como se citó en Polanco et al., 2021) define el término como una “forma de pensar que no se restringe en exclusiva hacia programadores de sistemas ni científicos en computación, sino como un grupo de habilidades útiles para todas las personas” (p.3). Además, estos procesos son indispensables para resolver problemas complejos mediante estrategias de aprendizaje, para alcanzar los procesos cognitivos necesarios para su resolución, donde, se refleje el potencial individual y colectivo.

A estos conceptos sobre el desarrollo de habilidades cognitivas, se suma Pérez y Roig (citado en Polanco et al., 2021), al indicar que el pensamiento computacional está relacionado con otros tipos de

pensamiento, como el matemático, lógico y crítico, dado que comparten habilidades cognitivas comunes tales como el reconocimiento de patrones, abstracción y modelación. Además, indican que la finalidad del pensamiento computacionales que, a partir del reconocimiento de los problemas reales de las actividades diarias, se propongan soluciones con la aplicación de herramientas informáticas.

Por ello, la concepción del pensamiento computacional es denominada como un “grupo de herramientas mentales con las que toda persona debería contar y aplicar para solventar situaciones, es un punto bastante natural, pero adiciona el hecho de diseñar sistemas y comprender el comportamiento humano” (Polanco et al., 2021, p.57). Asimismo, el fin del pensamiento computacional en la escuela, son las competencias de alfabetización digital enfocadas en promover el surgimiento de nuevos conocimientos con respecto al mismo.

Elementos del pensamiento computacional

El Pensamiento Computacional (PC) radica en resolver problemas de la vida diaria mediante el empleo de los conceptos elementales de la programación informática donde las soluciones pueden ser interpretadas mediante una secuencia de pasos o instrucciones. Debido a lo mencionado anteriormente el PC debería formar parte de la educación de todo ser humano. De modo que, según Rodríguez (2021) los elementos o principios son:

- Descomposición: es un principio del pensamiento computacional que tiene como objetivo dividir la problemática en secciones pequeñas, para así analizar cada sección parte por parte.
- Abstracción: permite destacar los elementos importantes del problema o situación para así establecer relaciones entre sí y trabajar en la construcción de solución o del elemento que solicite el contexto en el que se desarrolla la problemática.

- Reconocimiento de patrones: es analizar y establecer los patrones o elementos que surgen de las actividades y que pueden ser funcionales para dichas situaciones.
- Escritura del Algoritmo: permite tener un orden en las funciones y órdenes de los pasos a seguir y las decisiones que se toman en el proceso y así determinar la mejor manera de abordar la problemática o la solución necesaria.
- Evaluación y depuración: se detectan errores o anomalías para ajustar el proceso de resolución.
- Finalmente, la Solución, está resuelve el problema mediante la aplicación del sistema y los pasos elegidos. Las seis fases expuestas anteriormente representan las principales habilidades que demanda una persona para resolver problemas utilizando el PC.

Fomentar el pensamiento computacional en el aula

La educación, al igual que la sociedad en la actualidad, cada vez está más enlazada con la tecnología y la digitalización, procesos que evolucionan rápidamente transformando nuestro diario vivir. Sin embargo, no avanza al mismo tiempo cómo evoluciona la sociedad. En consecuencia, Montes et al. (2020) mencionan que en varias partes del mundo se está incorporando el pensamiento computacional en los currículos de la enseñanza como un medio para desarrollar las habilidades de resolución de problemas.

Las dos principales justificaciones para desarrollar las habilidades de PC en niños y jóvenes, según Montes et al. (2020), corresponde a que puedan pensar de manera diferente, expresarse a través de una variedad de medios, resolver problemas del mundo real y analizar temas cotidianos desde una perspectiva diferente y el fomento del PC para impulsar el crecimiento económico, cubrir puestos de trabajo y prepararse para futuros empleos.

Por consiguiente, el primer paso para el desarrollo del PC requerirá que los docentes trabajen diversas actitudes, tales como: la persistencia ante problemas difíciles de solucionar, la confianza a la hora de manejar situaciones complicadas, la tolerancia a la ambigüedad, la capacidad para enfrentar un problema no estructurado (Calvo, 2022); siendo capaces organizar y trabajar en equipo situaciones que conlleve un reto por resolver.

Además, implica que los docentes se encuentren en constante capacitación; una propuesta es formar comunidades o redes de aprendizaje o bien incorporarse a una ya existente (Montes et al., 2020); donde la interacción con otras personas, profesionales, estudiantes o aficionados permita tanto adquirir como compartir nuevos conocimientos y experiencias que den solución a problemas del entorno.

Aprender a programar a través de la acción, el legado de Seymour Papert

El aprendizaje de un lenguaje de programación permitirá que cualquier persona se relacione con su mente y además con la tecnología, logrando así tener una mejor comprensión de lo que implica la tecnología y su integración en nuestras vidas (Otavo, 2022). En cuanto a los niños esto les permitirá desarrollar habilidades que les contribuirá al proceso de aprendizaje, además, a que los mismos logren poder enfrentarse a las diferentes situaciones que se les presenten en la vida.

De modo que, la mejor manera de aprender es construyendo, así como menciona Seymour Papert (citado en Otavo, 2022) “Cuando un niño aprende a programar, el proceso de aprendizaje es transformado. Se vuelve más activo y autodirigido. El nuevo conocimiento es una fuente de poder y se experimenta como tal” (p. 4); así Papert asegura que el aprendizaje debe realizarse a través de la acción, confiando plenamente en el poder activo del proceso de aprendizaje.

Programación Logo

Uno de los primeros programas de programación para niños, se destacó Logo, se introdujo en 1967, era un lenguaje de programación con un marco integrado de ideas pedagógicas, tecnológicas y educativas diseñado por Seymour Papert basado en la forma de aprendizaje de los niños. Así, su libro Mindstorms (1980), escrito después de una década de investigación y experimentación con Logo, fue un hito para la educación informática y la enseñanza del pensamiento computacional bajo la premisa de pensamiento procedimental, Papert (citado en Otavo, 2022) argumentó que el aprendizaje es más efectivo cuando los alumnos “construyen conocimiento haciendo”; es decir, construyen conocimientos inmersos en el mundo de la práctica.

Como se menciona en el libro Construcción de conocimiento: Papert y su visión, Saldarriaga et al. (2020) aluden que “la programación como el Logo, el proceso de construcción del conocimiento programático puede ser imaginado y representado de manera más natural, lúdica, con pensamientos propios, con esquemas de secuenciación, de propiedad, de condición, de repetición, de formas estructuradas, organizadas, autoorganizadas” (p.52). De igual forma Logo muestra un entorno gráfico en el que un objeto con forma de “tortuga” se mueve por toda la pantalla, tiene como característica que las instrucciones se pueden comprender en diferentes idiomas, por lo tal razón, esto hace que Logo sea un lenguaje de programación fácil de aprender.

Teniendo en cuenta los aportes pedagógicos que presenta la programación, Papert (citado De la Rosa et al., 2020) expresó “veo Logo como un medio que puede, en principio, ser usado por educadores para apoyar el desarrollo de nuevas formas de pensamiento y aprendizaje” (p. 3). De igual forma Harvey (citado De la Rosa et al., 2020) menciona que es un programa interactivo, flexible, con procesos cortos y simples, recursivo y amigable con el usuario. En definitiva, se puede decir que,

para trabajar en este lenguaje, lo único que debemos saber es leer y escribir para poder interactuar con la tortuga.

Aprendizaje utilizando la programación FMSLogo

FMSLogo es un lenguaje que nos permite construir comandos que pueden ser ejecutados por el ordenador, es una implementación gratuita de un entorno informático llamado Logo, admite múltiples tortugas y gráficos 3D (FMSLogo, (s.f.)). De igual forma, FMSLogo admite una interfaz de Windows, además, este es un programa de fácil aprendizaje, razón por la cual este debe ser el lenguaje de programación para trabajar en entornos educativos. En la Figura 1.1, se puede observar la pantalla principal que se utiliza en la programación de FMSLogo.



Figura 1.1. Pantalla principal de FMSLogo. Fuente: Adaptado de FMSLogo (s.f.).

La ventana de gráficos, es la que escribe las órdenes de manera adecuada, se mueve el triángulo (tortuga), se puede realizar dibujos, animaciones, etc. En FMSLogo (s.f.), la caja de entrada, se dan las instrucciones a ejecutar; la ventana de Trabajo, son las órdenes que se realizan en la caja de entrada, los errores y resultados de las instrucciones que se utilizan; los botones, se encuentran en la zona inferior derecha; el botón

Alto finaliza la ejecución de un programa; Reiniciar borra la pantalla; y EdTodo abre un editor de texto.

Instrucciones de trabajo de FMSLogo

A continuación, se presentará las herramientas que se pueden utilizar en un entorno informático para programar en FMSLogo, ver Tabla 1.1; en la misma, se podrá apreciar todo lo referente a los comandos de la aplicación, mediante la utilización de múltiples tortugas y gráficos 3D, las cuales, los docentes pueden incorporar en sus clases para la enseñanza-aprendizaje de los estudiantes a través de FMSLogo.

Tabla 1.1. Principales primitivas o instrucciones de trabajo.

Primitiva	Descripción
av o fd / avanza	Avanza Tortuga.
re / retrocede	Retroceder Tortuga.
gd o rt 90 / giraderecha	Girar Derecha.
gd o lt 90 / giraizquierda	Girar Izquierda.
bp / borrapantalla	Borrar Pantalla.
sl / subrlapiz	Sube Lápiz (la tortuga no dibuja al moverse).
bl / bajalapiz	Bajar Lápiz.
Goma	La tortuga borrar al moverse.
centro	La tortuga se ubica en el centro de la pantalla.
lapiznormal / ponlapiz	Lapiznormal / ponlapiz al moverse.
ot / ocultar tortuga	Oculta la tortuga en la pantalla de gráficas.
mt / mostrar tortuga	Muestra la tortuga en la pantalla de gráficas.
repite n [instrucciones]	Instrucciones.

Fuente: Adaptado de comandos en FMSLogo.

En la Tabla 1.1, se perciben las instrucciones que se deben seguir para poner en práctica en programación para FMSLogo, que consta de dos apartados. En primer lugar, tenemos la primitiva en la que se observa las abreviaturas que se utilizan para emplearlas en la plataforma antes mencionadas. En segundo lugar, se presenta la descripción de las

abreviaturas del programa la misma servirá como base en la utilización de las herramientas en el aula de clases.

Experimentación

Ejemplo 1

En la Figura 1.2, se visualizará varias imágenes geométricas, las cuales se las ha desarrollado en el programa FMSLogo, el cual, es una herramienta gratuita de un entorno informático llamado Logo. El mismo, fomenta el aprendizaje de los estudiantes, además, brinda apoyo para el estudio de las distintas materias, incluidas las matemáticas, la ingeniería, el arte, la música y la robótica.

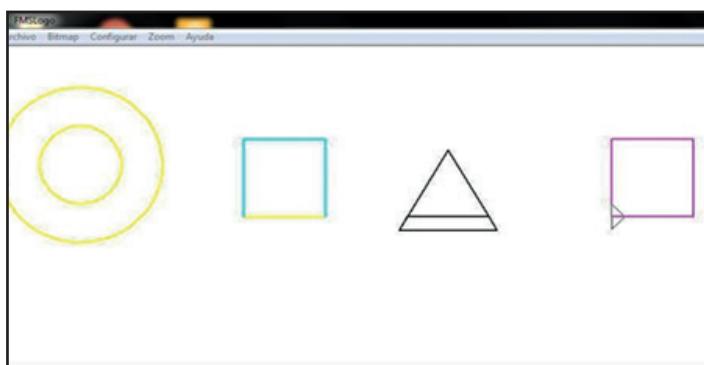


Figura 1.2. Figuras realizadas en FMSLogo. Fuente: Elaboración propia.

La Figura 1.2, da a conocer el paso a paso de la construcción de gráficas en FMSLogo, por consiguiente, para realizar un cuadrado empezamos moviendo la tortuga a la izquierda, escribimos avanza 35 y avanzará 35 espacios, escribimos el código gira izquierda 90, sube lápiz, avanza 300, gira izquierda 90, avanza 100, giro izquierda 90, escribimos el comando baja lápiz, avanza 100, giro izquierdo 90, avanza 100, giro izquierda 90, avanza 100, gira izquierda 90 y avanza 100 espacios más y así se realiza un cuadrado.

Para realizar un triángulo empezamos moviendo la tortuga a la izquierda, baja lápiz, avanza 100, giro a la izquierda 90 y después giro a la izquierda 30, avanza 100, gira izquierda 120, avanza 100 y así es como obtenemos un triángulo perfecto. Para realizar un círculo escribimos el comando círculo 50 o círculo 100 dependiendo del tamaño que se desee aplicar.

Ejemplo 2

A continuación, en la Figura 1.3 se presentará la imagen de una flor realizada en la herramienta FMSLogo, además, se explicará detalladamente el paso a paso sobre la construcción de la misma en el programa anteriormente mencionado.

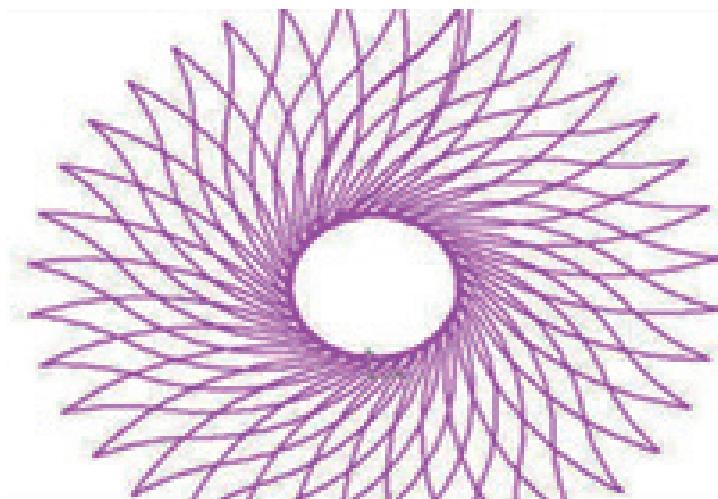


Figura 1.3. Programación de una flor en FMSLogo. Fuente: Elaboración propia.

En la Figura 1.3, se procedió a realizar una flor en la cual se detalla las veces que se desea ejecutar la acción utilizando el comando REPITE n [instrucciones], una vez que se han detallado los comandos de repetición se procede a copiar la programación y se dirige a EdTodo, una vez ahí

a lado de “para” se escribe como se llamará los comandos copiados. Entre la configuración “para” y “fin” se pega los comandos que hemos seleccionado con anterioridad, se selecciona “guardar y salir” y se procede a utilizar la siguiente programación REPITE [instrucciones], y en lugar de escribir cada comando se pone el comando flor, se ejecuta el programa y realizará todo el procedimiento.

Reflexiones

La programación es una vía para el desarrollo de diversas habilidades del pensamiento; tales como el creativo, lógico, la motivación, el PC; a su vez el mismo va a permitir el razonamiento en la solución de problemas ocupando como medio las computadoras, desarrollando así el análisis, la metacognición y la innovación; con el fin de que los estudiantes puedan dominar las tecnologías que los rodea.

La enseñanza del PC puede tener un impacto positivo en el desempeño académico de los estudiantes, ya que ha demostrado ser una nueva vía a explorar y desarrollar dentro de las etapas educativas especialmente en las áreas de matemáticas y ciencias. La implementación de enfoques pedagógicos basados en proyectos y resolución de problemas es una estrategia efectiva para enseñar la misma, debido a los beneficios que aportan y la cantidad de competencias que pueden trabajarse a través de Science, Technology, Engineering, Arts & Mathematics (STEAM).

Así mismo, se puede decir que Papert fue él que introdujo la idea de la programación en el ámbito de la tecnología en la educación; este se trata de un lenguaje computacional sencillo y fácil de usar, el cual permitirá que los educadores puedan realizar sus propios programas y utilizarlos según su metodología educacional, y de esta manera los niños y niñas puedan lograr desarrollar una manera de pensar y aprender de su propio aprendizaje.

FMSLogo es una aplicación que sirve para programar, la cual ayudará a que los estudiantes puedan adquirir habilidades del pensamiento y lógica, ya que en esta aplicación podrán construir elementos gráficos, los cuales van a apoyar en el aprendizaje en las diferentes áreas, como la matemática, geometría, etc., por esta razón hoy en día este tipo de programación sigue siendo un método ideal para los niños que quieran iniciarse en este campo.

Bibliografía

- Adell, S., Llopis, M., Esteve, F. y Valdeolivas, M. (2019). El debate sobre el pensamiento computacional en educación. *Revista Iberoamericana de Educación a Distancia*. 22(1), 171-186
- Álvarez, J. (2021). La robótica educativa y el pensamiento computacional en la educación infantil: Propuestas de intervención bajo un enfoque STEM. In *Entornos virtuales para la educación en tiempos de pandemia: Perspectivas metodológicas* (pp. 1781-1802). Dykinson. <https://dialnet.unirioja.es/servlet/articulo?codigo=8265959>
- Brugés, A. y Camperos, Y. (2022). Desarrollo del pensamiento computacional a través del aprendizaje de la programación en estudiantes de ingeniería. *Revista de investigación y praxis en ciencias sociales*. 1(1), 5.
- Calvo, N. B. (2022). El desarrollo del pensamiento computacional en la educación inicial mediante estrategias de enseñanza utilizando robótica educativa. <https://repositorio.udesa.edu.ar/jspui/bitstream/10908/19671/1/%5BP%5D%5BW%5D%20M.%20Edu%20Calvo,%20Natal%C3%ADn%20Bel%C3%A9n%C3%A9n.pdf>
- De la Rosa, A., Flores, I., Allier, I. y Valadez, S. (2020). El papel de la corriente construcionista en la práctica docente y el aprendizaje. *Humanidades, tecnología y ciencia del instituto politécnico nacional*, 22(1), 1-5.
- FMSLogo (s. f.). *FMSLogo y control por ordenador*. http://www.ieslosalbares.es/tecnologia/FMSLogo_y_control_por_ordenador/fmsLogo.html

Gómez, M. (2020). *Aspectos de adquisición de lenguaje en la enseñanza de programación*. Universidad Nacional de Córdoba-Facultad de Matemática, Astronomía, Física y Computación. Universidad Nacional de Córdoba. https://www.researchgate.net/profile/Marcos-Gomez-5/publication/345256073_Aspectos_de_adquisicion_de_lenguaje_en_la_ensenanza_de_programacion/links/5fa1c2d5a6fdccfd7b982334/Aspectos-de-adquisicion-de-lenguaje-en-la-ensenanza-de-programacion.pdf

González, V., Borull, A. y Valls, C. (2023). Fomentar las habilidades de programación y pensamiento computacional: una acción formativa para futuros docentes de educación infantil. *Campus Virtuales*, 12(2), 57-68. <https://dialnet.unirioja.es/descarga/articulo/9102401.pdf>

Montes, H., Hijón R., Pérez, D. y Montes, R. (2020). Mejora del pensamiento computacional en estudiantes de secundaria con tareas unplugged. *Education in the Knowledge Society* 21(24). <https://repositorio.grial.eu/bitstream/grial/2127/1/23002-81879-1-PB.pdf>

Otavo, N. (2022). *Implementar nociones de lenguaje de programación en los estudiantes del grado segundo de primaria del Liceo Nuestra Señora de las Nieves, por medio de la herramienta Scratch*. [Fundación Universitaria Los Libertadores]. Repositorio Institucional. https://repository.libertadores.edu.co/bitstream/handle/11371/5268/Otavo_Nury_2022.pdf?sequence=1&isAllowed=y

Polanco, N., Ferrer, S. y Fernández, M. (2021). Aproximación a una definición de pensamiento computacional. RIED. *Revista Iberoamericana de Educación a Distancia*, 24(1), 55-76. <http://dx.doi.org/10.5944/ried.24.1.27419>

Rodríguez, P. (2021). Estrategias para el desarrollo de habilidades de pensamiento computacional: experiencia de aprendizaje en el Seminario Pensamiento Computacional.

Roig, R. y Moreno, V. (2020). El pensamiento computacional en Educación. Análisis bibliométrico y temático. *Revista De Educación a Distancia* (RED), 20(63). <https://revistas.um.es/red/article/view/402621/279891>

Saldarriaga, G. E., Lasso, M. A. M., Sánchez, E. R. V., Badilla, E., Quinn, M., Alzati, E. C. y Solórzano, C. M. V. (2020). *Constructores de conocimiento: Papert y su visión I*. Corporación Universitaria para el Desarrollo de Internet. https://redlate.net/wp-content/uploads/2021/01/Papert_RED-LaTE-2.pdf

Vera, M. (2021). El desarrollo de la Competencia digital en el alumnado de Educación Infantil. Edutec. *Revista Electrónica de Tecnología Educativa*, (76), 126-143. <http://www.uajournals.com/ojs/index.php/campusvirtuales/article/view/1187/622>

Capítulo 2

El aprendizaje basado en simulación para aprender la programación de computadoras

Coloma Andrade María de los Angeles, Quezada Vera Lida Andreina, Sisalima Angamarca Darly Germania.

Universidad Nacional de Loja.

maria.coloma@unl.edu.ec, laquezadav@unl.edu.ec, darly.sisalima@unl.ed.ec

Introducción

Las implicaciones del presente capítulo está orientado a un impacto social educativo puesto que se analiza a la simulación como método para aprender sobre programación de computadoras donde implica estimular habilidades lógico cognitivas con el fin de mejorar los resultados de aprendizaje, un impacto positivo en los objetivos educativos enfocados a la programación de computadoras y por ende al desarrollo tecnológico, debido que la creciente era digital genera la demanda de habilidades, esto implica la capacidad de analizar, comprender y utilizar la tecnología, resultados que son observables en los procesos formativos.

Los antecedentes del presente estudio enfatizan la necesidad de buscar métodos que garanticen mejores resultados en el aprendizaje de la programación, considerando el bajo desempeño de esta asignatura que presentan los estudiantes respaldados por autores como Berrocal, quien enfatiza la necesidad de potenciar el uso de los simuladores en ambientes de aprendizaje. Por tal razón, el propósito central del estudio se enfoca al aprendizaje basado en la simulación y conceptos básicos de programación para identificar el impacto y relación que guardan entre sí estos temas, así como también se analiza un caso práctico del simulador denominado PSeInt.

El aprendizaje basado en la simulación

El aprendizaje basado en simulación se fundamenta en la idea de que la experiencia práctica es fundamental para el aprendizaje efectivo, según Alhadlaq (2023) “la simulación se ha convertido en una herramienta sofisticada que puede reflejar escenarios del mundo real con considerable precisión”, es decir, la simulación proporciona un entorno seguro para que los estudiantes cometan errores y experimenten sin temor a consecuencias negativas.

En consecuencia, la simulación es un método de enseñanza y aprendizaje estimulante para la obtención del conocimiento, orientado a satisfacer las necesidades de aprendizaje. Este método puede combinar partes mecánicas y virtuales que permiten la formación profesional o el uso como instrumento de entretenimiento; un ejemplo de esto son los laboratorios virtuales usados para ejecutar códigos de programación, los cuales son accesibles de forma gratuita y solo requieren un navegador web con acceso a internet (Aguilar y Heredia, 2013). Esta combinación de accesibilidad y funcionalidad hace de los laboratorios virtuales en una valiosa herramienta que proporciona a los estudiantes un entorno práctico y seguro para experimentar y aplicar sus conocimientos teóricos.

Asimismo, las simulaciones permiten a los estudiantes interactuar con el código en tiempo real, Jamil y Isiaq (2019) manifiestan que experimentar con diferentes configuraciones y visualizar los resultados facilita la comprensión de conceptos abstractos, de igual manera indican que “hoy en día, los estudiantes pueden localizar hechos, captar conocimientos científicos, ideas y principios”, estimulando habilidades intelectuales a través de actividades de aprendizaje simuladas en la computadora.

De esta manera, el aprendizaje basado en las simulaciones se potencializa bajo un entorno de aprendizaje constructivista enriquecido con la colaboración entre pares donde exige al alumno implicarse de forma activa en su aprendizaje de acuerdo a Berrocal (2009). Además, las simulaciones deben ser coherentes a los objetivos educativos, conocimientos previos de los estudiantes y contexto social para lograr una construcción oportuna de nuevos conocimientos. En función a Salas y Ardanza (1995), al emplear un simulador se debe considerar guías orientadoras que integren objetivos claros a lograr; demostración práctica que emplee medios de enseñanza y teoría; ejercitación de forma independiente y evaluación de resultados.

En cuanto al procedimiento metodológico de las simulaciones en el proceso de enseñanza y aprendizaje, López y Saiegg (2005) describen las siguientes etapas:

- Presentación de la simulación: representación esquemática del circuito, dispositivo o fenómeno a simular.
- Emisión de hipótesis por parte de los estudiantes: deberá orientarse a los estudiantes indiquen los posibles comportamientos, cambios en los parámetros y posibles resultados en el modelo a simular.
- Determinación de las acciones óptimas: se determinan las acciones que los estudiantes deberán realizar ante una situación.
- Constatación de la efectividad del proceso de simulación: puede realizarse por medio de tareas que permitan aplicar a nuevas situaciones los conocimientos construidos en el proceso de simulación.

Ventajas y desventajas de las simulaciones en el proceso educativo

Las simulaciones proporcionan a los discentes la oportunidad de enfrentarse a decisiones y desafíos, lo que resalta su valor en el proceso educativo, asimismo la experiencia en un entorno simulado proporciona

aptitudes prácticas limitadas al espacio del mismo (Hallinger & Wang, 2020). Aunque las simulaciones ofrecen ventajas como la experiencia práctica y la interactividad, también es crucial reconocer y abordar sus posibles limitaciones para garantizar una implementación efectiva.

A propósito de lo anterior, el uso de la simulación para el proceso de enseñanza aprendizaje trae consigo ventajas como desventajas como concuerdan los autores Aguilar y Heredia (2013), López y Saiegg (2005), Peña (2019) y Ruedas (1988), tal como se representa en la Tabla 2.1. Donde es importante reconocer al estudiante como el ente constructor activo del aprendizaje bajo espacios que estimulan la autoevaluación, creatividad y capacidades cognitivas; cabe resaltar que el éxito de implementar un simulador a los procesos formativos requiere de un adecuado análisis a las necesidades educativas, caso contrario, favorecerá a una mala inversión de tiempo y recursos.

Tabla 2.1. Ventajas y desventajas de las simulaciones en el proceso educativo.

	Las simulaciones son un mecanismo dinámico para la construcción del aprendizaje.
Ventajas	Permite la autonomía de los estudiantes fortaleciendo los procesos de metacognición.
	Favorece el desarrollo de actividades interdisciplinarias.
	Reduce el gasto de recursos y optimiza el tiempo.
	Puede producir errores en los resultados por el diseño mal desarrollado.
Desventajas	No responde a las necesidades educativas requeridas al no seleccionar adecuadamente el simulador.
	No es una técnica de optimización para saber qué es lo mejor al simular una situación.

Fuente: Adaptado a partir de Aguilar y Heredia (2013), López y Saiegg (2005), Peña (2019) y Ruedas (1988).

Programación de computadoras

La programación de computadoras requiere una dedicación considerable de tiempo para su aprendizaje, según Ibarra et al. (2021) manifiestan que es desafiante para principiantes, en especial para discentes que aún no están familiarizados con el paradigma de los lenguajes procedimentales. Por ende, el propósito de enseñar programación no solo es adquirir habilidades en el uso de herramientas, sino también obtener experiencia en la resolución de problemas durante el proceso de programación.

Por tal razón, la programación de computadoras no solo corresponde a la escritura de código, comprende un conjunto de tareas que se debe realizar para resolver un problema en específico, por tanto, se debe transformar en un algoritmo y luego en un programa como manifiesta Juganaru (2014). Por consiguiente, un algoritmo es una lista de operaciones definida y ordenada que dan respuesta a un problema específico mediante pasos desde un estado inicial y llegan a un estado final, para describir un algoritmo se realiza en tres niveles como se muestra en la Figura 2.1, a continuación:



Figura 2.1. Niveles de descripción de un algoritmo. Fuente: Adaptado a partir de Juganaru (2014).

El algoritmo puede expresarse mediante un lenguaje natural, pseudocódigo, diagramas de flujo y programas como menciona Jukanaru (2014). Mientras que, para desarrollarlo con base lógica, comprende cuatro etapas como se muestra en la Figura 2.2, que corresponden a: definición, análisis, diseño e implementación. Cada una de estas etapas se desarrolla de manera secuencial y ordenada para el funcionamiento adecuado del algoritmo, el mismo que parte de la identificación del problema, análisis de sus características y soluciones, diseño de la solución y ejecución de pruebas.

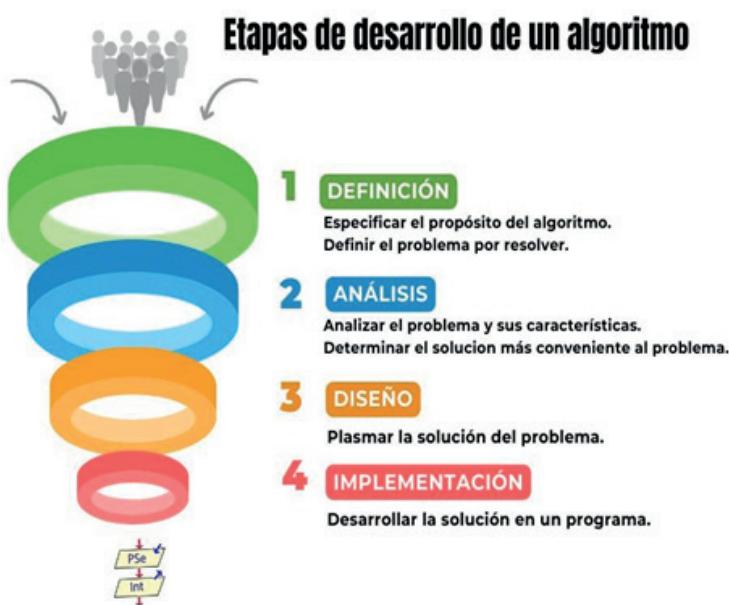


Figura 2.2. Etapas de desarrollo de un algoritmo. Fuente: Adaptado a partir de Jukanaru (2014).

El conjunto de instrucciones que al ejecutarse la computadora realiza una o varias tareas, se denomina programa informático y de forma general software. Donde Jukanaru (2014) alude que se debe escribir

con instrucciones en un lenguaje de programación definido por su propia sintaxis, reglas de escritura y semántica de los tipos de datos, instrucciones, entre otros elementos. Un programa de usuario inicia en la edición que corresponde al editor de texto donde se escribe el programa según el lenguaje elegido; luego la compilación donde el código fuente se transforma en instrucciones para la máquina y finalmente el enlazado que construye un ejecutable a partir de código objeto y librerías de funciones.

Programación de computadoras en base a simulaciones

La simulación no ha encontrado la suficiente acogida como metodología articuladora a excepción de su uso como recurso o técnica puntual de aula; estudios realizados por Angelini et al. (2024) indican que las simulaciones pueden compensar la falta de interacción en el aula y proporcionar a los alumnos un aprendizaje participativo práctico. De este modo, aprender a programar proporciona un entorno interactivo donde los conceptos abstractos pueden ser aplicados en situaciones concretas, fomentando la experimentación y el aprendizaje activo.

En consecuencia, aprender sobre la programación requiere conocimientos de tipo procedimental, así Aguilar y Heredia (2013) señalan que los estudiantes para desarrollar programas eficientes deben cubrir cada uno de los niveles (conocer, comprender, aplicar, analizar, sintetizar, evaluar) según los procesos cognitivos complejos presentes en la taxonomía de Bloom; donde el alumno simula el proceso (algoritmo) y lo desarrolla en un programa. Por ende, en la Tabla 2.2 se establece las actividades orientadas promover procesos cognitivos.

Tabla 2.2. Actividades de programación relacionadas con la taxonomía de Bloom.

Nivel	Actividad que se espera de un alumno que esté en ese nivel
Conocer	Saber cuáles son los métodos existentes y su sintaxis.
Comprender	Leer un conjunto de instrucciones y decir qué hacen paso a paso.
	Leer un programa y decir qué hace.
Aplicar	Usar elementos de una biblioteca de funciones.
	Poder aplicar un método para generar un efecto deseado.
Analizar	Leer un programa con el objetivo de hacerle cambios. Puede ser para corregir errores o para realizar una modificación a su funcionamiento.
	Traducir un programa a un diagrama de flujo.
Sintetizar	Escribir un programa dados los requerimientos.
Evaluar	Comparar dos programas dados que resuelven el mismo problema.

Fuente: Adaptado a partir de Aguilar y Heredia (2013).

En contraste con otro autor sobre el uso práctico de los simuladores en la educación, Berrocal (2009) menciona que el uso de las simulaciones potencializa los recursos tecnológicos y los procesos de aprendizajes en la elaboración de algoritmos de programación debido a las resultados que se evidenciaron en estudiantes cuando se les proporcionó animaciones sobre algoritmos quienes crearon sus propios ejemplos y explicaron los pasos del algoritmo para producir un resultado afianzando la comprensión de los conceptos.

Caso práctico: PSeInt

PSeInt es un simulador de licencia libre, gratuito y multiplataforma destinado a estudiantes en sus primeros pasos en el aprendizaje de la programación mediante un pseudolenguaje en español simple e intuitivo en función a un editor de diagramas de flujo para apoyar los conceptos fundamentales de algoritmos, construcción de programas o algoritmos computacionales; con un entorno de trabajo, recursos didácticos,

conjunto de ayudas y asistencias para encontrar errores y comprender la lógica de los algoritmos.

Se determina pertinente este simulador ya que da cumplimiento a las actividades de programación que se relacionan con la taxonomía de Bloom comprendiendo sus 6 niveles para desarrollar un algoritmo correspondiente a: conocer la sintaxis que maneja el simulador, comprender el conjunto de instrucciones, aplicar los elementos que nos brinda para diseñar una actividad de adivinar un número, analizar si el juego presenta posibles errores traduciéndolos a un diagrama de flujo, sintetizar los requerimientos para escribir la actividad y evaluar los resultados de ejecución como se muestra en la Figura 2.3.

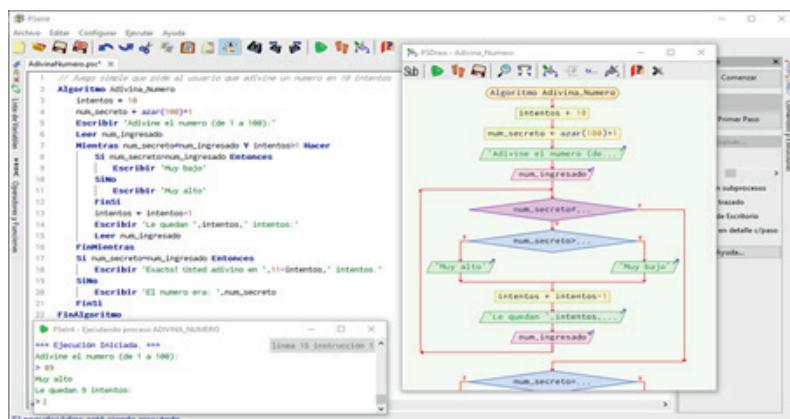


Figura 2.3. Simulador PSeInt. Fuente: Adaptado a partir de Novara (2003).

En una revisión sistemática de literatura sobre la enseñanza-aprendizaje de la programación en PSeInt, basada en criterios y directrices Preferred Reporting Items for Systematic reviews and Meta-Analyses (PRISMA), reveló un aporte significativo al desarrollo de habilidades de pensamiento algorítmico en el contexto educativo, en consecuencia, existe un fuerte impacto en las competencias computacionales y matemáticas en la resolución de problemas (Navarrete et al., 2023). Por

tanto, es de menester el uso PSeInt ya que contiene diversos elementos necesarios para el aprendizaje tanto en la descripción como el desarrollo de un algoritmo en las diversas formas que puede ser este expresado.

En este sentido, en un estudio comparativo entre PSeInt y RAPTOR realizado por Figueroa et al. (2023), a esta primera, la determinan como una herramienta que favorece el aprendizaje para la introducción a la programación, puesto que cuenta con la representación de flujoogramas, como se puede observar en la Figura 2.4, sobre un algoritmo para adivinar un número; también utiliza en el pseudocódigo palabras claves para representar acciones u operaciones mostrando donde existen errores, promoviendo habilidades de pensamiento computacional como la automatización y depuración. De esta manera, estimula la capacidad de solucionar y encontrar errores lógicos en este proceso de aprendizaje.

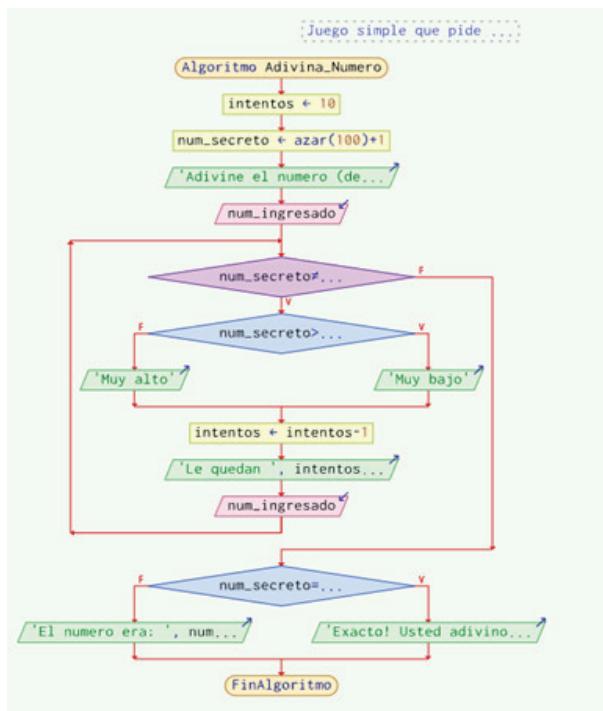
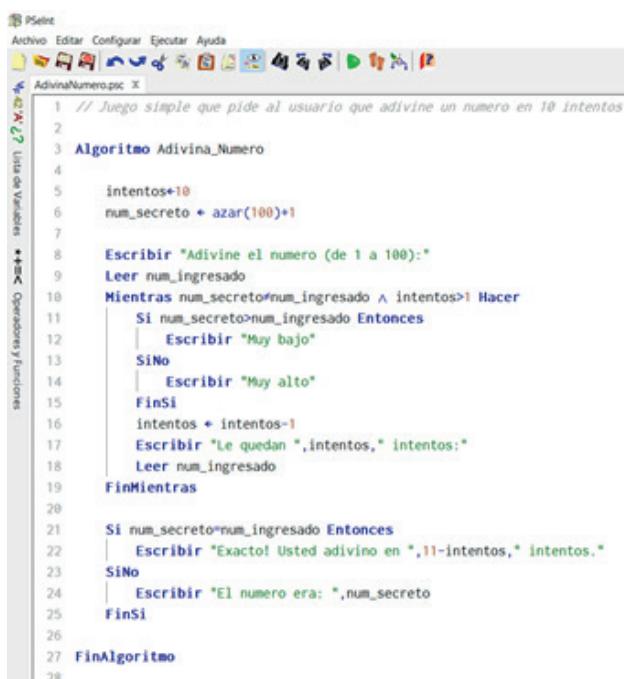


Figura 2.4. Simulador PSeInt. Fuente: Adaptado a partir de Novara (2003).

Por otro lado, en el aprendizaje de la programación existen varios pasos que nos permiten implementar el simulador PSeInt como herramienta de alto valor, primero se realiza un análisis del problema para obtener los datos de Entrada-Proceso-Salida; segundo, se identifica los datos para determinar las variables respectivas; y tercero, se procede a desarrollar el algoritmo de manera ordenada, finita y congruente a la solución del problema (Ascencio y Gil, 2023). Conviene resaltar que este proceso de implementación tiene concordancia con la taxonomía de Bloom donde se aborda diferentes niveles ascendentes de aprendizaje en cuanto a la programación de computadoras.



```

1 // Juego simple que pide al usuario que adivine un numero en 10 intentos
2
3 Algoritmo Adivina_Numero
4
5     intentos+10
6     num_secreto + azar(100)+1
7
8     Escribir "Adivine el numero (de 1 a 100):"
9     Leer num_ingresado
10    Mientras num_secreto!=num_ingresado Ambiental intentos>1 Hacer
11        Si num_secreto<num_ingresado Entonces
12            Escribir "Muy bajo"
13        SiNo
14            Escribir "Muy alto"
15        FinSi
16        intentos + intentos-1
17        Escribir "Le quedan ",intentos," intentos:"
18        Leer num_ingresado
19    FinMientras
20
21    Si num_secreto==num_ingresado Entonces
22        Escribir "Exacto! Usted adivino en ",11-intentos," intentos."
23    SiNo
24        Escribir "El numero era: ",num_secreto
25    FinSi
26
27 FinAlgoritmo
28

```

Figura 2.5. Simulador PSeInt. Fuente: Adaptado a partir de Novara (2003).

En función de lo planteado sobre la lógica de programación en el contexto del aprendizaje, Salazar (2022) utiliza dos simuladores: DFD y PSeInt, para analizar y resolver ejercicios propuestos a lo largo de

los capítulos de su libro. PSeInt, siendo este último simulador, es una herramienta que permite representar pseudocódigos, es decir, la representación textual de un algoritmo, facilitando así el aprendizaje de programación mediante un lenguaje natural, como se muestra en la Figura 2.5 con un ejemplo de algoritmo. El uso de este simulador proporciona una variedad de elementos que favorecen la comprensión y estimulan los procesos cognitivos.

Reflexiones

El aprendizaje basado en la simulación se convierte en una herramienta indispensable para el proceso educativo puesto que brinda espacios para cometer errores que no repercuten a consecuencias negativas reales que afecten la integridad de las personas, en consecuencia, estimula la comprensión de conceptos abstractos y potencializa habilidades intelectuales; sin embargo, es necesario determinar la coherencia de la simulación con los objetivos educativos, contexto social y los conocimientos previos de los estudiantes con la finalidad de constatar la efectividad de los nuevos conocimientos adquiridos.

El proceso de enseñanza aprendizaje en la programación de computadoras implica una verdadera conciencia en el cambio de paradigma actual, donde la construcción del conocimiento implica un aprendizaje activo del estudiante en función de la guía docente, una característica principal de los simuladores, donde los conocimientos son escalables por los niveles de desarrollo de un algoritmo, potenciando habilidades cognitivas relacionadas al pensamiento computacional y matemático en la resolución de problemas.

El caso práctico del presente capítulo revela la pertinencia del simulador PSeInt como herramienta indispensable para involucrarse en los primeros pasos en la programación, ya que cuenta con la representación de fluajogramas y utiliza pseudocódigos con un conjunto

de asistencias para encontrar y comprender errores en la lógica del algoritmo, abordando los niveles ascendentes de aprendizaje en función a la taxonomía de Bloom. Por otra parte, la investigación se limitó a una revisión bibliográfica más no práctica que involucre la participación directa con estudiantes, una brecha para futuras líneas de investigación.

Bibliografía

- Aguilar, I. y Heredia, J. (2013). La simulación en el aprendizaje basado en competencias de la programación estructurada. En Santillán F. (Coord). El proceso de aprendizaje e innovación en el siglo XXI; Una experiencia Iberoamericana basada desde la perspectiva del alumno, profesor y TIC (pp. 215-218). Centro de Estudios e Investigaciones para el Desarrollo Docente A.C. <http://ri.uaemex.mx/handle/20.500.11799/41155>
- Alhadlaq, A (2023). Computer-Based Simulated Learning Activities: Exploring Saudi Students' Attitude and Experience of Using Simulations to Facilitate Unsupervised Learning of Science Concepts. *Applied Sciences*, 13(7). <https://www.mdpi.com/2076-3417/13/7/4583>
- Angelini, L., Diamanti, R. y Jiménez, M. (2024). La simulación telemática y su impacto en la formación inicial de profesorado. *Revista Iberoamericana De Educación*, 94(1), 55–82. <https://doi.org/10.35362/rie9415942>
- Ascencio, E. y Gil, M. (2023). Enseñanza de la programación: ¿Nueva didáctica o una didáctica diferente? *Revista Cubana de Educación Superior*, 42(3). http://scielo.sld.cu/scielo.php?script=sci_abstract&pid=S0257-43142023000300018&lng=es&nrm=iso
- Berrocal, V. (2009). Consideraciones para el uso de simulaciones en entornos virtuales para el aprendizaje de las estrategias de programación de computadoras. *Fundación Dialnet*, 11(16). <https://dialnet.unirioja.es/servlet/articulo?codigo=5181345>

Figueroa, G., Paladines, J., Paladines, J., Aguilar, L., Moreira, M., Campozano, Y. y Merchán E. (2023). *Estudio comparativo entre pseint y raptor basado en la estructura while*. Editorial ALEMA. <https://editorialalema.org/libros/index.php/alema/article/view/7/8>

Hallinger, P. & Wang, R. (2020). The Evolution of Simulation-Based Learning Across the Disciplines, 1965–2018: A Science Map of the Literature. *Simulation & Gaming*, 51 (1), 9-32. <https://doi.org/10.1177/1046878119888246>

Ibarra, R. E., Castillo, J. O., Trujillo, P. C., García, C., Montesino, R. y Pando, B. (2021). Enseñanza-aprendizaje de programación de computadoras: avances en la última década. *Revista Científica*, 42(3), 290-303. <https://doi.org/10.14483/23448350.18339>

Jamil, M. G. y Isiaq, S. O. (2019). Teaching technology with technology: approaches to bridging learning and teaching gaps in simulation-based programming education. *Int J Educ Technol High Educ* 16(25). <https://doi.org/10.1186/s41239-019-0159-9>

Juganaru, M. (2014). *Introducción a la programación*. Grupo Editorial Patria. https://www.academia.edu/31289773/Introducci%C3%B3n_a_la_programaci%C3%B3n

López, C. y Saiegg, C. (2005). Uso de la simulación como estrategia de mejora en el proceso de enseñanza-aprendizaje en las universidades. Una aplicación para la carrera de informática. *Universidad Nacional de la Patagonia San Juan Bosco*.

Navarrete, L., Freire, G., Fernández, G., Gilces, E. y Mego, C. (2023). La enseñanza-aprendizaje de programación en computadora con PSeint: Una revisión sistemática. *Revista Científica*

Multidisciplinar G-ner@ndo, 4(2), 923–936. <https://doi.org/10.60100/rcmg.v4i2.179>

Novara, P. (2003). PSeInt (20210609). Multiplataforma. Argentina.

Peña, O. (2019). La simulación como recurso didáctico para el aprendizaje de los fundamentos de programación. *Revista Atlante: Cuadernos de Educación y Desarrollo*. <https://www.eumed.net/rev/atlante/2019/07/simulacion-recurso-didactico.html>

Rueda, S. (1988). El rol de la simulación en la teoría constructivista. SEDICIRepositoryInstitucionalde la UniversidadNacionalde la Plata. http://sedici.unlp.edu.ar/bitstream/handle/10915/24046/Documento_completo.pdf?sequence=3&isAllowed=y

Salas, R. y Ardanza, P. (1995). La simulación como método de enseñanza y aprendizaje. *Revista Scientific Electronic Library Online (SciELO)*, 9(1). http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0864-21411995000100002

Salazar, C. (2022). Lógica de programación. *Instituto Superior Tecnológico CEMLAD*. https://dspace.itsjapon.edu.ec/jspui/bitstream/123456789/3764/1/logica_de_programacion.pdf

Capítulo 3

Herramientas de codigoalfabetización y Code.org para la enseñanza de la programación

Labanda Jaramillo Milton Leonardo, Gonzalez Armijos Edgar Wilfrido, Villamagua Vicente Sandra del Carmen.

Universidad Nacional de Loja.

miltonlab@unl.edu.ec, edgar.w.gonzalez@unl.edu.ec, sandra.villamagua@unl.edu.ece

Introducción

Actualmente la mayoría de las personas posee un smartphone, desempeñando un papel importante en el diario vivir resultando una herramienta muy útil para que los niños comiencen su aventura en el universo del coding, apoyando el proceso de enseñanza aprendizaje. El coding empleado en la educación pretende enseñar lenguajes de programación desde una edad temprana para que empiecen a familiarizarse y desenvolverse con los conceptos más básicos, como vía para desarrollar futuras habilidades digitales.

Se debe fomentar la codigoalfabetización en el campo educativo porque desarrolla en los estudiantes el pensamiento lógico-sistématico preparando para un mayor nivel profesional, por lo tanto, la organización code.org ha promovido la iniciativa Hour of Code un movimiento global que fomenta la programación en los sistemas educativos para formar ciudadanos capaces de pensar, de formular y solucionar problemas sirviéndose de la tecnología que nos circunda.

Por tanto, en el presente artículo se presentarán los diferentes recursos que existen para la enseñanza-aprendizaje de la programación y se hará uso de la plataforma code.org, una organización que tiene como objetivo

ampliar el acceso a las ciencias de la computación en las escuelas. La plataforma permite al usuario la oportunidad de aprender ciencias de la computación de la misma manera que aprende biología, química o álgebra.

La codigoalfabetización en el siglo XXI

Años atrás la programación era asunto de profesionales en la materia: ingenieros en sistemas, en redes, informáticos, físicos y matemáticos, pero actualmente podemos apreciar un contexto en donde los más pequeños comprenden estos lenguajes. La programación es actualmente una tendencia en crecimiento y forma parte de las actividades más innovadoras en contextos educativos donde la tecnología tiene el protagonismo (García, 2021), fomentando la formación y desarrollo de habilidades a los estudiantes, que posibiliten la resolución de problemas del ámbito escolar.

Por ello, se empieza con la definición de la codigoalfabetización del inglés code-literacy que es el conjunto de procesos de enseñanza-aprendizaje de la lectoescritura con los lenguajes informáticos de programación (Román, 2016). Es decir, se asume que un individuo está código-alfabetizado cuando posee la capacidad de leer y escribir utilizando el lenguaje de los ordenadores y otras máquinas; en otras palabras, pensar computacionalmente.

El movimiento code literacy surge en los Estados Unidos alrededor del año 2010 y tiene como axioma fundamental el considerar y promover, la programación informática como el alfabetismo clave en el siglo XXI (Román, 2016). Promoviendo como un aspecto importante para el siglo XXI, su inclusión al proceso de enseñanza-aprendizaje vinculado a la lectoescritura de los lenguajes informáticos de programación

Debido a que actualmente, el concepto de alfabetización ha hecho una transición, precisamente gracias al desarrollo y creciente uso de los recursos tecnológicos en los distintos sectores de la sociedad. Esto ha provocado en el campo educativo según Caballero (2020), se realice una actualización a procesos clásicos como la lectoescritura, desarrollando una alfabetización digital desde los primeros años de escolaridad.

Por eso, para incentivar la programación se inicia con la tendencia de la codigoalfabetización en educación denominado coding, que fomenta el aprendizaje de lenguajes de programación en niños y jóvenes desde edades tempranas (Avello et al, 2020), donde desarrolla el pensamiento lógico-sistemático, prepara al alumno para obtener un mayor nivel profesional y de competencia laboral.

Como resultado, la codigoalfabetización se está convirtiendo en un elemento central en el aprendizaje de las áreas de conocimiento, es decir, Ciencia, Tecnología, Ingeniería y Matemática, comúnmente utilizada en sus siglas en inglés como STEM; permitiendo interactuar a mayor nivel de complejidad con la lectoescritura de código, liberando las limitaciones de las opciones predefinidas y configuraciones por defecto de las aplicaciones informáticas que se utilizan cada día (Bers, 2018), por lo cual, se ha empezado a implementar la programación o coding dentro de los centros formativos para los más pequeños.

Recursos para la enseñanza-aprendizaje del coding

Dentro de este orden de ideas se presenta la recopilación de recursos para la enseñanza-aprendizaje de la programación y fomentar la codigoalfabetización en la educación, caracterizado con tres criterios: descripción, tipo de lenguaje de programación que enseña y la edad así como también los niveles de educación de acuerdo al currículo, destacando entre ellos: Alice, App Inventor, App Lab, Bee-Bot App,

Bitbloq, Blockly, Cargo-Bot, Codecombat, Code.org, Codecademy, CodeHS, Daisy the Dinosaur, Hopscotch, Khan Academy, NetLogo, Play my Code, S4A Scratch for Arduino, Scratch, ScratchJr, The Foos, que se describen en la Tabla 3.1, a continuación.

Tabla 3.1. Recursos para fomentar la codigofabetización.

Recursos	Descripción	Tipo de lenguaje de programación	Edad y nivel educativo
Alice	Es un entorno de programación innovador basado en bloques que facilita la creación de animaciones, la creación de narrativas interactivas o la programación de juegos sencillos en 3D. Está diseñada para enseñar habilidades de pensamiento lógico y computacional, principios fundamentales de programación y ser una herramienta para el acercamiento a la programación orientada a objetos (Universidad Carnegie Mellon, 2021). Es un software libre orientado a objetos, permitiendo la creación de animaciones mediante la función de arrastrar y soltar objetos.	Visual por bloques	Educación General Básica (EGB) (9 a 14 años)
App Inventor	Es un entorno de programación visual e intuitivo que permite a todos, incluso a los niños, crear aplicaciones completamente funcionales para teléfonos inteligentes; basada en bloques que facilita la creación de aplicaciones complejas de alto impacto en corto tiempo que los	Visual por bloques	EGB (9 a 14 años)

Recursos	Descripción	Tipo de lenguaje de programación	Edad y nivel educativo
	entornos tradicionales (Instituto de Tecnología de Massachusetts, s/f), facilitando elaborar aplicaciones para el sistema operativo Android, por medio de la programación por bloques que permite enlazar una serie de piezas que contienen ciertos códigos para crear la aplicación.		
App Lab	<p>Continúa con la idea de programar a través de bloques de colores, pero con una novedad, que posibilita crear apps completas para escritorio y móvil. Permitiendo crear aplicaciones, juegos y todo tipo de herramientas en un formato web,, accesible y ejecutable a través de un navegador.</p> <p>La ventaja de este recurso es que utiliza JavaScript, un lenguaje de programación de scripts (secuencia de comandos) orientado a objetos (Eguíluz et al, 2018).</p>	Visual por bloques	EGB (9 a 14 años)
Bee-Bot App	<p>Es una aplicación de codificación que facilita la codificación y la programación mejorando las habilidades de los usuarios en el lenguaje direccional, secuencias de programación de adelante, atrás. Disponible para dispositivos iOS (iPhone y iPad); en fin, es una aplicación de programación para los niños porque hace que la programación sea divertida y fácil (Ferrada et al, 2019).</p>	Visual por flechas	Educación Inicial o comienzos de EGB (3 a 8 años)

Recursos	Descripción	Tipo de lenguaje de programación	Edad y nivel educativo
Bitbloq	<p>Es una herramienta creada para usarse conjuntamente con una placa reprogramable de Arduino. El objetivo es programar por bloques una serie de instrucciones para que sean introducidas en la placa y, conectando sensores y actuadores, se reproduzca físicamente lo programado (Alzate, 2023).</p>	Visual por bloques	EGB (9 a 14 años)
Blockly	<p>Es un proyecto de Google que muestra largos fragmentos de código basado en texto y en bloques visuales, es una biblioteca de JavaScript, permite producir código fuente en diferentes lenguajes (IONOS, 2020).</p>	Visual por bloques	EGB (9 a 14 años)
Cargo-Bot	<p>Es un juego programado completamente en iPad usando Codea™. Mejorando las habilidades de programación de una manera divertida por su atractivo diseño y fácil uso. La plataforma está diseñada de forma que el usuario pueda seleccionar diferentes tipos de flechas para controlar un brazo robótico, provocando que el robot mueva las cajas mediante las instrucciones del internauta (Eguíluz et al, 2018).</p>	Visual por flechas	Educación Inicial o comienzos de EGB (3 a 8 años)
Codecombat	<p>Es un juego de rol online en el cual se pueden aprender los fundamentos de lenguajes de programación basados en texto</p>	Textual	Bachillerato (15 a 17 años)

Recursos	Descripción	Tipo de lenguaje de programación	Edad y nivel educativo
Codecombat	tan populares como Python y JavaScript. Es online gratuito, y funciona como herramienta para la didáctica de la programación.	Textual	Bachillerato (15 a 17 años)
Codecademy	A través del cual permite establecer bloques de código para avanzar con el juego. Es un tipo de lenguaje textual (Melgosa y Vargas, 2020).	Textual	Bachillerato (15 a 17 años)
CodeHS	Es una plataforma interactiva y colaborativa para enseñar y aprender cómo codificar que puede ayudar a adquirir habilidades en lenguajes de programaciones populares como: JavaScript, Python, Ruby y plataformas como: JQuery, Projects y APIs (Sharp, 2019).	Textual	Bachillerato (15 a 17 años)
Daisy the Dinosaur	Es una plataforma de enseñanza integral para ayudar a las escuelas a enseñar informática. Ofrecen currículo basado en la web, herramientas y recursos para docentes, y desarrollo profesional. Además, los alumnos aprenden a programar a un nivel avanzado en JavaScript, Python y Java (CodeHS, 2018).	Textual	Bachillerato (15 a 17 años)
	El objetivo del juego es permitir al usuario elegir lo que quiera que haga el objeto y arrastrando la orden o instrucción a una caja sea capaz de hacer una determinada acción. Las instrucciones son muy sencillas	Visual por bloques	EGB (9 a 14 años)

Recursos	Descripción	Tipo de lenguaje de programación	Edad y nivel educativo
Daisy the Dinosaur	que son: mover, saltar, girar, crecer, encoger y las instrucciones típicas para repetir utilizando comandos como while para realizar la repetición o bucle (Lennon et al, 2022).	Visual por bloques	EGB (9 a 14 años)
Hopscotch	Es una aplicación para programar con bloques desde dispositivos Apple, y que permite crear proyectos y compartirlos con la comunidad, así como descubrir los ya creados y publicados por todo el mundo, destacando que su interfaz es sencilla e intuitiva (Espeso, 2018).	Visual por bloques	EGB (9 a 14 años)
Khan Academy	Es una plataforma web que ofrece ejercicios de práctica, videos instructivos y un panel de aprendizaje personalizado que permite a los estudiantes aprender a su propia ciencia, la programación de computadoras, al ritmo, dentro y fuera del salón de clases. Aborda las matemáticas, la historia, la historia del arte, la economía, etc., (Lasso y Conde, 2021).	Textual	Bachillerato (15 a 17 años)
Play my Code	Es una plataforma HTML5 para crear y distribuir juegos desde el navegador. Actualmente, en Beta, el sitio se muestra promisorio como una ventanilla única para las personas que buscan ingresar a la codificación de juegos HTML5.	Textual	Bachillerato (15 a 17 años)

Recursos	Descripción	Tipo de lenguaje de programación	Edad y nivel educativo
S4A Scratch for Arduino	S4A es una modificación de Scratch que permite programar la plataforma de hardware libre Arduino de una forma sencilla. Proporciona bloques nuevos para tratar con sensores y actuadores conectados a una placa Arduino (Caraballo, 2022).	Visual por bloques	EGB (9 a 14 años)
Scratch	Es un entorno de programación visual que permite a los usuarios crear proyectos multimedia interactivos. La programación se realiza enlazando bloques de comandos, de diferentes colores, para controlar objetos gráficos. Además, los usuarios pueden registrarse y compartir sus proyectos, comentar otros proyectos y formar grupos con intereses comunes (Díaz y Lozano, 2019).	Visual por bloques	EGB (9 a 14 años)
The Foos	Es un juego que presenta conceptos de programación y ciencias de la computación, a través de un lenguaje de codificación visual que es divertido y accesible para personas de todas las edades. Muestra la programación como un videojuego donde el usuario resuelve problemas de los personajes (Eguíluz et al., 2018).	Visual por flechas	Educación Inicial o comienzos de EGB (3 a 8 años)
Pilas Engine	Pilas Engine es un motor de videojuegos de código abierto diseñado para facilitar el aprendizaje de la programación mediante la creación de juegos. Con una interfaz amigable y visual,	Visual por bloques / Textual	EGB (9 a 14 años)

Recursos	Descripción	Tipo de lenguaje de programación	Edad y nivel educativo
Pilas Engine	Pilas Engine permite a los usuarios desarrollar juegos de forma rápida y sencilla, sin necesidad de tener conocimientos avanzados de programación (Belcastro, 2020).	Visual por bloques / Textual	EGB (9 a 14 años)
Pilas Bloques	Pilas Bloques es una versión de Pilas Engine diseñada específicamente para enseñar programación a niños y principiantes mediante el uso de bloques de código visualmente intuitivos. Con Pilas Bloques, los usuarios pueden arrastrar y soltar bloques para crear juegos y animaciones, lo que facilita la comprensión de los conceptos de programación básicos como bucles, condicionales y funciones (Kuz y Ariste, 2022).	Visual por bloques	EGB (9 a 14 años)
LightBot	LightBot es un juego educativo que enseña conceptos básicos de programación a través de rompecabezas y desafíos. En el juego, los jugadores controlan a un robot que debe encender baldosas en un tablero utilizando comandos de programación básicos como avanzar, girar y encender. A medida que avanzan los niveles, planificar cuidadosamente sus movimientos y utilizar la lógica para completar los desafíos (Kuz y Ariste, 2022).	Visual por bloques	EGB (9 a 14 años)

Recursos	Descripción	Tipo de lenguaje de programación	Edad y nivel educativo
Turtle Art	<p>Es un entorno de programación visual que utiliza un lenguaje basado en bloques para crear arte y animaciones. Está diseñado para enseñar conceptos de programación a través de la manipulación de una “tortuga” virtual que dibuja en la pantalla.</p> <p>Los usuarios pueden arrastrar y soltar bloques de programación para controlar el movimiento de la “tortuga” y crear patrones y formas complejas. Es una herramienta educativa para enseñar matemáticas, geometría y conceptos de programación (Kuz y Ariste, 2022).</p>	Visual por flechas	Educación Inicial o comienzos de EGB (3 a 8 años)

Fuente: Adaptado a partir de Roman (2016), Cruz (2019) y Kuz y Ariste (2022).

Los recursos anteriores seleccionados, al ser de coste gratuito, ofrecen una ventaja significativa al usuario al permitirle acceder a ellos sin incurrir en ningún gasto. Esto resulta especialmente beneficioso para aquellos que desean desarrollar habilidades en pensamiento crítico y computacional, creatividad y resolución de problemas, ya que pueden utilizar estos recursos de manera libre y sin restricciones económicas.

Es importante destacar que, si bien estos recursos son gratuitos, mantienen restricciones en su copyright. Esto significa que, si bien se pueden utilizar sin costo, no se pueden modificar, vender o utilizar de manera libre como ocurre con el software libre. Esta limitación puede influir en la forma en que se utilizan estos recursos, ya que se debe

respetar la propiedad intelectual y las condiciones de uso establecidas por los creadores de los recursos.

Code.org

Para la enseñanza-aprendizaje del coding se hace uso de la plataforma code.org, una organización sin fines de lucro a la vez dedicada a ampliar el acceso a las ciencias de la computación en las escuelas (Gamboa y Varela, 2020). En la página principal de la plataforma code.org <https://code.org/> se encuentra tres apartados importantes que se detalla a continuación:

- Cursos elementales: Incluye cursos de iniciación destinados a niños y adolescentes de educación inicial, educación general básica y bachillerato y cursos de introducción previstos para desarrollar.
- Sección para los docentes: Permite a los interesados conocer las ciencias de la computación, especialmente a los docentes enseñar la informática a sus alumnos por medio de code.org.
- Proyectos: Permite revisar proyectos iniciales en Game Lab, App Lab y Weblab, estos incluyen descripciones de proyectos, consejos y proyectos de demostración. Además, permite que los estudiantes puedan remezclar o puedan crear sus propios proyectos.

Es importante mencionar que la plataforma Code.org facilita programar aplicaciones web y para móvil con bloques visuales tipo Scratch, pero tiene la ventaja de visualizar el código en JavaScript; sin embargo, prohíbe la descarga del código. La Figura 3.1, presenta una aplicación que permite calcular las cuatro operaciones básicas: suma, resta, multiplicación y división; desarrollado en code.org en la sección de App Lab, siendo ejecutable tanto en el navegador como en los dispositivos móviles.



Figura 3.1. Pantalla principal de la aplicación Fuente: Tomado de Code.org (2024)

Mientras que, en la Figura 3.2 muestra el código fuente en un lenguaje visual por bloques, lo que facilita la comprensión para aquellos que están empezando a aprender programación. Este tipo de representación gráfica permite ver de manera clara cómo se estructura el programa y cómo fluyen las instrucciones.

```
1
2 onEvent( ▶"buttondivide" , ▶"click" , function( ● ) {
3     setText( ▶"labeldivideans" , getText( ▶"text_inputdividefirst" )
4     + getText( ▶"text_inputdividesecond" ) );
5 }
6 onEvent( ▶"buttondivide" , ▶"click" , function( ● ) {
7     setScreen( ▶"screendivide" );
8 }
9 onEvent( ▶"buttoncalculatedivide" , ▶"click" , function( ● ) {
10     setText( ▶"labeldivide" , getNumber( ▶"text_inputdivide1" ) /
11     getNumber( ▶"text_inputdivide2" ) );
12 }
13 onEvent( ▶"buttondivback" , ▶"click" , function( ● ) {
```

Figura 3.2. Código fuente lenguaje visual por bloques Fuente: Tomado de Code.org (2024)

Por otro lado, la Figura 3.3 presenta el mismo código, pero en lenguaje textual, específicamente en JavaScript. Este lenguaje es más común en el desarrollo de aplicaciones web y permite una mayor flexibilidad y control sobre el programa. Sin embargo, puede resultar más complejo para quienes no están familiarizados con la sintaxis de JavaScript.

```
onEvent("buttondivide", "click", function() {
    setText("labeldivideans", getText("text_inputdividefirst") %
    getText("text_inputdividesecond"));
});
onEvent("buttondivide", "click", function() {
    setScreen("screendivide");
});
onEvent("buttoncalculatedivide", "click", function() {
    setText("labeldivide", getNumber("text_inputdivide1") /
    getNumber("text_inputdivide2"));
});
onEvent("buttondivback", "click", function() {
});
```

Figura 3.3. Código fuente convertido en lenguaje textual JavaScript. Fuente:
Tomado de Code.org (2024).

El enlace <https://bit.ly/3Cayo2D> proporcionado lleva a la aplicación desarrollada, lo que brinda la oportunidad de ver el código en acción y comprender mejor cómo funciona la operación de dividir implementada en la aplicación; proporcionando a los estudiantes y programadores una variedad de formas de aprender y comprender cómo se puede implementar una operación básica como la división en un programa informático.

En fin, la programación o coding ha ganado un lugar relevante en la educación debido a su capacidad para desarrollar habilidades y destrezas en los estudiantes. Se ha vuelto común encontrar asignaturas específicas de programación o su inclusión de manera transversal en diversos niveles

educativos (Cruz, 2019). Esto se debe a que la programación no solo enseña a programar computadoras, sino que también fomenta el pensamiento crítico, la resolución de problemas, la creatividad y la colaboración.

Al integrar la programación en el currículo escolar, se impulsa a los docentes a adoptar metodologías activas como el aprendizaje basado en retos, la gamificación y el aprendizaje cooperativo. Estas metodologías no solo hacen que el aprendizaje sea más interactivo y significativo, sino que también ayudan a los estudiantes a aplicar lo que aprenden en situaciones prácticas y reales (Kuz y Ariste, 2022). De esta manera, los proyectos de programación no solo se convierten en una forma de adquirir conocimientos sobre tecnología, sino también en una oportunidad para desarrollar habilidades de trabajo en equipo, liderazgo y comunicación.

Reflexiones

Para finalizar, el coding en educación pretende enseñar lenguajes de programación desde una edad temprana para que empiecen a familiarizarse y desenvolverse con los conceptos más básicos. Considerando que la codigofalfabetización desarrolla el pensamiento lógico-sistemático, prepara al discente para obtener un mayor nivel profesional y de competencia laboral, por lo tanto, algunas instituciones han implementado la programación en sus planificaciones.

En conclusión, los recursos anteriormente detallados para la enseñanza aprendizaje del coding, resultan de gran ayuda, porque están sistematizados en diferentes categorías de enseñanza, siendo un gran potencial para empezar la programación en niños y adolescentes.

Se ha verificado que la plataforma code.org es una iniciativa para enseñar a programar a niños con bloques permitiendo guiar al alumno

a diferentes niveles de dificultad, fomentando la enseñanza de la programación mediante cursos interactivos para el aprendizaje de cualquier tipo de lenguaje de programación. Se recomienda utilizar la plataforma en educación porque permite enseñar a los niños la programación mediante herramientas para escribir código y ayudas como los tutoriales, videos y programas gratis.

En último lugar, se presentan algunos trabajos futuros que pueden desarrollarse como resultado de esta investigación o que no han podido ser tratados con la suficiente profundidad, entre ellos destacan: determinar la aplicabilidad e implementar los recursos de enseñanza-aprendizaje de programación en las instituciones educativas ecuatorianas o implementar code.org para la enseñanza-aprendizaje de programación, desarrollar el pensamiento lógico-sistemático y fortalecer el pensamiento computacional en las instituciones educativas.

Bibliografía

- Alzate, Y. (2023). El fortalecimiento de la robótica educativa y el pensamiento computacional a través de VEX.code y Bitbloq. *Revista Ciencias Y Humanidades*, 17(17), 73–92. <https://doi.org/10.61497/rcyh.v17i17.209>
- Avello, R., Lavonen, J. y Zapata, M. (2020). Codificación y robótica educativa y su relación con el pensamiento computacional y creativo. Una revisión compresiva. *Revista de Educación a Distancia* (RED), 20(63). <https://doi.org/10.6018/red.413021>
- Bers, U. (2018). Coding, playgrounds and literacy in early childhood education: The development of KIBO robotics and ScratchJr. 2018 *IEEE Global Engineering Education Conference (EDUCON)*, 2094–2102. <https://doi.org/10.1109/EDUCON.2018.8363498>
- Belcastro, A., Dibez, P., Quiroga, M. y Bertone, R. (2020). Evento de transferencia DeCoDev, desarrollo colaborativo de un videojuego. In XXII Workshop de Investigadores en Ciencias de la Computación (WICC 2020, El Calafate, Santa Cruz). <http://sedici.unlp.edu.ar/handle/10915/103920>
- Caraballo, H. (2022). Los entornos virtuales de aprendizaje como espacios en la sociedad del conocimiento. *Tecnologías de la información y la comunicación en el contexto educativo.* (pp. 77-99). Corporación Universitaria Minuto de Dios – UNIMINUTO. <https://doi.org/10.26620/uniminuto/978-958-763-565-2.cap.1>
- CodeHS. (2024). *What is CodeHS?* | CodeHS. [online]. <https://codehs.com/info/>

Caballero, Y. (2020). *Desarrollo del pensamiento computacional en Educación Infantil mediante escenarios de aprendizaje con retos de programación y robótica educativa* [Tesis doctoral, Universidad de Salamanca Programa de Doctorado Formación en la Sociedad del Conocimiento)]. Repositorio Institucional de la Universidad de Salamanca.

Cruz, E. (2019). *Centro de recursos para la enseñanza y el aprendizaje en una educación basada en competencias digitales: gaming y videojuegos para las alfabetizaciones múltiples y la formación del profesional de la información* [Tesis doctoral, Universidad Carlos III de Madrid]. <http://hdl.handle.net/10016/28225>

Díaz, G. y Lozano, R. (2019). *Uso de las aplicaciones code. org y scratch para el aprendizaje de programación en los estudiantes del 5º y 6º grado de EBR del CE N° 82099 de la provincia de San Pablo, 2018* [Tesis de pregrado, Universidad Privada Antonio Guillermo Urrelo]. Repositorio Institucional de la Universidad Privada Antonio Guillermo Urrelo.

Eguílez, A., Garaizar, P. y Guenaga, M. (2018). *An Evaluation of Open Digital Gaming Platforms for Developing Computational Thinking Skills.* InTech. <http://dx.doi.org/10.5772/intechopen.71339>

Espeso, P. (2018). *Hopscotch, la app para programar con bloques desde tu tablet.* Educación tres punto cero. <https://n9.cl/mjwm4>

Ferrada, C., Díaz, D., Salgado N. y Parraguez, R. (2019). Propuesta de actividades STEM con Beebot en matemática. *Edma 0-6: Educación Matemática en la Infancia*, 8(1), 33-43.<http://funes.uniandes.edu.co/13715/1/72-1-228-1-10-20190719.pdf>

- Gamboa, R., y Varela, M. (2020). CODE. ORG: *Software para estimular el pensamiento lógico y abstracto* [Tesis de pregrado, Universidad Católica de Pereira]. Repositorio Institucional de la Universidad Católica de Pereira.
- García J. (2021). Implicancia de la inteligencia artificial en las aulas virtuales para la educación superior. *Orbis Tertius - UPAL*, 5(10), 31–52. <https://doi.org/10.59748/ot.v5i10.98>
- IONOS. (2020). *¿Qué es Blockly?* IONOS: España. <https://n9.cl/z4nob>
- Instituto de Tecnología de Massachusetts. (s/f). *MIT App Inventor*. <https://appinventor.mit.edu/about-us>
- Kuz, A. y Ariste, M. (2022). Análisis y revisión de softwares educativos para el aprendizaje de la programación en entornos lúdicos. *Tecné, Episteme y Didaxis: TED*, (52), 117-136. <https://doi.org/10.17227/ted.num52-13159>
- Lasso, C y Conde, K. (2021). Khan Academy como herramienta en el aprendizaje de las matemáticas y la programación. *Revista Interamericana De Investigación Educación Y Pedagogía RIEP*, 14(1), 225–250. <https://doi.org/10.15332/25005421.5777>
- Lennon, M., Pila, S., Flynn, R., y Wartella, E. (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. <https://doi.org/10.1016/j.compedu.2022.104608>
- Melgosa. I., y Vargas, V. (2020). CodeCombat. [Mensaje en un blog]. <https://cutt.ly/ZmNNPQ1>

Román, M. (2016). *Códigoalfabetización y pensamiento computacional en Educación Primaria y Secundaria: validación de un instrumento y evaluación de programas* [Tesis doctoral, Universidad Nacional de Educación a Distancia (España)]. Repositorio Institucional de la Universidad Nacional de Educación a Distancia (España).

Sharp, J. (2019). Using Codecademy Interactive Lessons as an Instructional Supplement in a Python Programming Course. *Information Systems Education Journal*, 17(3), 20-28. <http://isedj.org/2019-17>

Universidad Carnegie Mellon CMU. (2021). *Alice*. <https://www.alice.org/>

Capítulo 4

Enseñanza de la programación orientada a objetos con la herramienta Alice

*Jhon Jairo Carrión Castillo, Stiven Enrique Lima Neira, Ortega Medina
Maria Alexandra, María de los Angeles Coloma Andrade.*

Universidad Nacional de Loja.

jhon.j.carrión@unl.edu.ec, stiven.lima@unl.edu.ec, maria.a.ortega.m@unl.edu.ec, maria.coloma@unl.edu.ec

Introducción

En el contexto educativo actual, la enseñanza de la programación informática se enfrenta a desafíos significativos, principalmente en los cursos introductorios, donde los estudiantes a menudo encuentran dificultades en comprender y aplicar los conceptos básicos, en la Programación Orientada a Objetos (POO). Integrar la programación en el currículo educativo es reconocido por muchos educadores como una estrategia clave para promover un aprendizaje significativo y mejorar las habilidades de los estudiantes.

En este sentido, en el presente capítulo se enfatiza la importancia de la programación como una herramienta didáctica clave, resaltando a Alice como medio pedagógico idóneo para enseñar los principios fundamentales de la POO de manera tangible y comprensible, gracias a su interfaz intuitiva y su capacidad facilitar simular situaciones en un entorno 3D.

Programación

La incorporación de la tecnología en el ámbito educativo es una transformación constante que ofrece técnicas y recursos de última generación con el propósito de mejorar el proceso de enseñanza y

aprendizaje. Con la utilización de entornos interactivos, materiales digitales y equipos de última tecnología, los educadores pueden crear experiencias de aprendizaje personalizadas y cautivadoras, como afirma Patrón (2022). Esto les permite convertir las aulas en espacios vivos y ajustados a los requerimientos específicos de cada estudiante.

La programación es el arte de estructurar acciones y órdenes dentro de un programa de manera que se ejecuten de forma esperada. Se trata de un proceso de toma de decisiones donde, por ejemplo, un docente planifica su intervención pedagógica de manera intencionada y metódica. como describe Gaul et al. (2013), la programación como la visión anticipada y organizada de una actividad, considerando todos los elementos que participan en su ejecución, y se traduce en la creación de un plan de acción ante cualquier tarea humana a realizar.

En el ámbito educativo, la programación actúa como una herramienta clave al estructurar el proceso enseñanza-aprendizaje, adaptándose a las necesidades culturales y contextuales, según Gómez (2001), la programación se considera importante porque es una asignatura esencial que facilita la creación de programas con comportamientos específicos. Por lo tanto, la escritura de código demanda conocimientos multidisciplinarios y un sólido dominio del lenguaje de programación para asegurar la correcta ejecución del código.

Por esta razón, los lenguajes de programación son el medio donde los humanos instruyen a las computadoras, especificando acciones a ejecutar. Martín et al. (2021) definen la programación como la creación de programas que, a través de un lenguaje formal, organizan algoritmos y procesos lógicos que un sistema informático ejecutará, controlando así su funcionamiento y la interacción con el usuario.

Siguiendo esta línea, los lenguajes de programación se componen de símbolos, reglas sintácticas y semánticas, expresadas en forma de instrucciones y relaciones lógicas destinadas a desarrollar el código fuente de una aplicación o pieza de software determinada. Solano (2011) señala que varios lenguajes actualmente son considerados de alto nivel, tales como BASIC, COBOL, Pascal, Ada, C, C++, Java y Python, diseñados para ofrecer funcionalidades avanzadas y simplificar el desarrollo de programas complejos.

Paradigmas de Programación

Los paradigmas de programación son fundamentales porque proporcionan un marco de trabajo que impacta directamente en la eficiencia, claridad y calidad del software que se desarrolla. En consonancia con Cerón y Domínguez (2010), estos paradigmas influyen en la estructura de los programas, definiendo la realización de cálculos y la organización de tareas. Entre ellos se encuentran los enfoques procedimentales y declarativos, cada uno con principios distintos que orientan a la creación de soluciones de software.

Paradigmas procedimentales

Los paradigmas procedimentales, conocidos por su enfoque en la ejecución de soluciones, detallan secuencialmente los pasos necesarios para alcanzar un resultado. Solano (2011) explica que un paradigma procedimental, actúa como un agente activo que opera sobre datos pasivos mediante instrucciones, que son acciones específicas que realice el programa.

En lo que se refiere a este paradigma, se compone de tres elementos claves: las estructuras en la creación de datos, las llamadas a procedimientos y bloques de código asociados que definen el procedimiento; la habilidad

de combinar y extender estos elementos propicia a los programadores crear procedimientos más complejos y sofisticados. Lenguajes como FORTRAN, COBOL, Pascal, C y Ada ejemplifican este enfoque, proveyendo las herramientas necesarias para una programación estructurada y eficiente.

Paradigmas declarativos

Los paradigmas declarativos se centran en el razonamiento lógico y la definición de problemas mediante proposiciones y ecuaciones. Estos paradigmas instruyen al sistema sobre qué resultado se desea obtener, sin necesidad de especificar los pasos exactos. Como expresa Domínguez (2005), en este enfoque las variables se utilizan para almacenar valores, pero no en modificar estados, lo que elimina la posibilidad de asignaciones que alteren el estado del sistema. Esto refleja una aproximación más abstracta y matemática a la programación, donde lo importante es describir qué debe hacerse, más que cómo hacerlo.

Programación Pedagógica

La resolución de problemas mediante lenguajes de programación constituye la esencia de la formación en informática. Por ende, Díaz et al. (2018) ratifica que es imprescindible que el estudiante adquiera un conjunto de habilidades específicas y domine los conocimientos y procedimientos alineados con las técnicas de programación actuales y las herramientas de desarrollo pertinentes. Más allá de la teoría, el modelo educativo exige que el futuro docente internalice prácticas y estrategias que optimicen su desempeño profesional en el entorno escolar.

En cuanto a su función principal, radica en la resolución de problemas mediante el uso de lenguajes de programación. Para lograrlo, Díaz et al. (2018) destaca lo importante que desarollen habilidades específicas,

sino que también adquieran un conocimiento profundo de las técnicas y herramientas de programación actuales los estudiantes. Más allá de la teoría, el modelo educativo profesional requiere que los futuros profesores adopten prácticas pedagógicas efectivas que enriquezcan su desempeño en el aula.

Sin embargo, Ruiz (2010) sugiere que la capacitación de los docentes debe ser una prioridad al enfrentar desafíos emergentes, con la finalidad de maximizar el potencial de las Tecnologías de la Información y la Comunicación (TIC). Ampliando esta idea, Tadeu (2020) corrobora que es fundamental que los educadores estén completamente familiarizados con las tecnologías, con el propósito de transformar las prácticas pedagógicas y enriquecer la experiencia de aprendizaje de acuerdo a sus necesidades.

Entorno de programación Alice

Alice constituye una plataforma de programación autónoma y didáctica, desarrollada en Java y enfocada en la orientación a objetos. De acuerdo con Ramírez et al. (2011), su entorno de desarrollo integrado (IDE) es de fácil manejo y promueve la creación de animaciones mediante una interfaz de arrastrar y soltar. Además, posee una amplia galería de modelos 3D que estimula a los usuarios a animar sus conceptos de forma interactiva.

Esta plataforma está disponible en distintas versiones, como la 2.0 y 2.2, adaptadas a diferentes niveles educativos; la versión más actual, “Alice 3” ha superado su fase beta y sigue mejorando constantemente con actualizaciones y correcciones de errores (Ramírez et al., 2011). Esta última versión se caracteriza por su editor de diseño avanzado, controles de cámara enmendados, una función de búsqueda optimizada, así como la capacidad de seleccionar y editar subpartes de los objetos con detalle.

Dentro de este marco, Vecino y Seleey (2009) enfatiza que Alice proporciona una retroalimentación visual inmediata, siendo fundamental para que los estudiantes comprendan cómo el código que escriben afecta el comportamiento de los objetos en el programa. Esta interacción directa con los objetos en un espacio virtual mejora significativamente la comprensión de los conceptos fundamentales de la codificación, haciendo de Alice una herramienta pedagógica excepcional hacia los cursos introductorios de programación.

¿Cómo se usa Alice?

Alice es ideal a quienes se acercan por primera vez al mundo de la programación, gracias a su entorno gráfico que simplifica la escritura de código a través de una interfaz intuitiva, tal como lo resaltan Ramírez et al. (2011). Esta accesibilidad la convierte en una herramienta valiosa y accesible para los principiantes, estimulando su aprendizaje y fomentando una comprensión clara de los conceptos esenciales de la programación.

El proceso de creación de programas en Alice es un proceso sencillo y accesible, gracias a su entorno visual interactivo. Briceño (2009), indica que, al iniciar, el usuario elige un escenario, como arena, hierba, espacio o nieve, entre otros, para comenzar el desarrollo de su programa. Posteriormente, ofrece una interfaz gráfica completa donde se pueden agregar todo tipo de objetos tridimensionales de su biblioteca, dependiendo de la temática seleccionada.

Cabe resaltar que Alice brinda un entorno de programación gráfico donde cada objeto cuenta con una serie de acciones predefinidas, como girar, cambiar de color y moverse. Estas acciones otorgan a los usuarios interactuar con su entorno virtual y animar objetos con facilidad. Según Vecino y Seleey (2009), este entorno 3D permite a los

usuarios programar y animar objetos, proporcionando una introducción accesible y agradable a la codificación. La visualización inmediata de ver los resultados del código mejora la comprensión de la codificación por parte de los usuarios. Además, la interfaz de Alice está organizada en cinco secciones principales como se detalla en la Tabla 4.1, lo que contribuye a su accesibilidad y facilidad de uso.

Tabla 4.1. Interfaz gráfica.

Ventana de todo el mundo	Se visualiza el planeta que se va creando con los objetos que se hayan incluido.
Ventana de objetos	Se muestra en forma jerárquica o de árbol, los objetos integrados. La raíz corresponde al mundo y en ella se ligan todos los objetos añadidos. En cualquier instante debería estar seleccionado un objeto o mundo.
Ventana de detalles	Se visualizan en 3 categorías las propiedades del objeto seleccionado: características, procedimientos y funcionalidades. Cabe decir que dependiendo de cada objeto dichas características, procedimientos y funcionalidades son concretas.
Ventana de programa	Corresponde a la visualización de los procedimientos accesibles en un mundo virtual. Por medio de la técnica de arrastrar y soltar aquí se puede detallar el “programa”, o sea, las secuencias de ocupaciones que deberán hacer una vez que se ejecute el programa
Ventana de eventos	En esta se añaden nuevos eventos, que tienen la posibilidad de conceptualizar por medio de la ventana de programas.

Fuente: Adaptado a partir de (CMU, 2014).

De este modo, la interfaz de Alice admite a los usuarios crear instrucciones mediante la técnica de “drag and drop”, en la que arrastran elementos gráficos. Vecino y Seleey (2009) comparan estas instrucciones con

líneas de código de lenguajes de programación como Java o C++, y resaltan que al observar cómo se ejecutan las animaciones en tiempo real, los estudiantes pueden comprender la relación entre el código y el comportamiento de los objetos.

Programación orientada a objetos con Alice

Alice es una plataforma que emplea Java, un lenguaje que admite múltiples paradigmas de programación, incluida la POO, centrada en la resolución de problemas mediante la creación de objetos, que son entidades que contienen datos y comportamientos. Es fundamental entender que cada objeto se caracteriza por tener atributos y métodos, como se ilustra en la Tabla 4.2, qué son las propiedades y acciones que pueden realizar.

Tabla 4.2. Propiedades de un objeto.

Atributos (Características)	Métodos (Comportamientos)
Nombre.	Ladrar.
Edad.	Caminar.
Color.	

Fuente: Adaptado a partir de Ramírez et al. (2011).

Basándonos en lo mencionado anteriormente, se presenta el siguiente ejemplo para una comprensión más clara: un perro puede ser considerado un objeto, ya que puede tener características específicas definidas por sus atributos y métodos. De esta manera, se exponen las propiedades individuales de un objeto con el fin de facilitar la gestión y el diseño de recursos mediante la herramienta Alice.

Herramienta Alice para crear entornos dinámicos de aprendizaje

Alice fomenta el aprendizaje mediante la creatividad al proporcionar un entorno donde se pueden desarrollar habilidades de pensamiento lógico y computacional. Morales (2022), deduce que esta herramienta está diseñada para enseñar los fundamentos de la programación, abarcando desde su sintaxis hasta los principios de la POO. También, ofrece recursos adicionales que guían a los usuarios de todas las edades en el uso de la plataforma, adaptándose a sus intereses específicos en informática.

La interacción en Alice crea un entorno innovador que facilita la creación de modelos en 3D y animaciones para narrar historias o desarrollar videojuegos interactivos, los cuales pueden compartirse en línea como se indica en la Figura 4.1. Su interfaz hace de Alice una herramienta intuitiva, similar a Scratch, pero con la ventaja de crear entornos tridimensionales, lo que resulta visualmente atractivo y capta la atención de los estudiantes desde el principio.



Figura 4.1. Interfaz de Alice. Fuente: Tomando de la CMU (2014)

Por consiguiente, en el marco del proyecto interactivo 3D según la CMU (2014), se encuentran tres elementos fundamentales; primero, los “métodos”, son un conjunto de instrucciones que propicia ajustar el tamaño o cambiar la posición de los objetos en la escena; segundo, “los argumentos”, se añaden a los métodos para definir detalles como movimiento, la dirección o la velocidad de los objetos; y tercero, en el apartado de “eventos” se encuentra las acciones del usuario, como presionar una tecla, que desencadenan respuestas de los objetos, tales como saltar o mostrar un mensaje. Estos elementos trabajan en conjunto con la finalidad de dar vida a tus ideas en el entorno virtual de Alice.

Para demostrar estos conceptos, se siguen estos pasos: 1. Generar un mundo virtual. 2. Seleccionar el archivo “Grass”. 3. Hacer clic en “Aregar Objetos”. 4. Explorar la carpeta “Nature” y seleccionar “Happy” de la galería, luego elige “frog” en “Animals”. En la Tabla 4.3 se detalla el procedimiento que contienen diferentes argumentos, facilitando una interacción dinámica en el mundo virtual de Alice.

Tabla 4.3. Procedimientos en el software Mundo Alice.

Procedimientos regulares	
Cambiar la magnitud de un objeto	A partir del árbol de objetos, seleccionamos frog (rana) y haciendo clic derecho vamos a ver el procedimiento frog resize y seleccionamos 2(twice a big) como tamaño. Para ver la animación clic en el botón play.
Gire el objeto con el procedimiento (turn) (izquierda, derecha, etc).	El objeto se mueve en la condición relativa (orientación del objeto. Sentido de dirección).
Utilice el procedimiento Turn left y ¼ revolutions.	Para ver la animación así mismo clic sobre el botón Play.

Seleccionando el método roll gira

Seleccionar el objeto	Sobre el método frog turn to face happy, tree la rana que aparece y mira hacia el árbol.
Aplicar el método stand up	
Desplazar una sub parte	La Rana (frog) sacando la lengua. Elegir la sub parte de la lengua y utilizar el procedimiento frog.head.jaw.tongue.move y seleccionar forward (hacia adelante) ½ meter y presionar enter.
Aproximación	Acercar la cámara hacia el objeto.

Fuente: Adaptado a partir de Buriticá (2014).

Resulta claro, que la falta de un editor de código en Alice es una ventaja notable, especialmente hacia los principiantes en la POO. Esto simplifica las dificultades relacionadas con la precisión sintáctica, permitiendo que los estudiantes se concentren en la lógica de la programación en lugar de la sintaxis. Alice facilita la introducción a la programación de una manera que fomenta la creatividad del estudiante y reduce la presión para cumplir con estrictas normas sintácticas.

Reflexiones

La programación es entendida como la habilidad de organizar secuencias de comandos en un software facilitando obtener los resultados deseados. En el contexto educativo, su incorporación es esencial para sistematizar el proceso de enseñanza-aprendizaje, adaptándolo a las necesidades y habilidades individuales por estudiante con el propósito de potenciar su creatividad y capacidad analítica al máximo ante los retos de la era digital actual.

Alice se destaca entonces como una herramienta que agiliza la creación de juegos y simulaciones, permitiendo a los estudiantes sin experiencia previa en programación entender rápidamente la lógica de la POO y familiarizarse con conceptos clave como clases, atributos y métodos, lo cual es crucial para su desarrollo académico y profesional.

Bibliografía

- Briceño, T. (2009). El paradigma científico y su fundamento en la obra de Thomas Kuhn. *Tiempo y Espacio*, 19(52), 285-296. https://ve.scielo.org/scielo.php?pid=S131594962009000200006&script=sci_arttext
- Buriticá, O. (2014). Relaciones de aprendizaje significativo entre dos paradigmas de programación a partir de dos lenguajes de programación. *Tecnura*, 18(41), 91-102. https://scielo.org.co/scielo.php?pid=S0123921X2014000300008&script=sci_arttext
- Cerón, J. y Domínguez, J. (2010). *Diseño e implementación de un objeto virtual de aprendizaje para la introducción a la programación de computadores dirigido a estudiantes de tercer semestre del programa de Licenciatura en Informática* (Doctoral dissertation, Universidad de Nariño). <https://sired.udenar.edu.co/8/>
- Díaz,K.I.,Fierro,E.yMuñoz,M.(2018).Laenseñanzadelaprogramación: una experiencia en la formación de profesores de informática. *Educación*, 27(53), 73-91.http://www.scielo.org.pe/scielo.php?script=sci_arttext&pid=S1019-94032018000200005
- Domínguez, I. (2005) de Programación, C. D. P. Fundamentos teóricos de los Paradigmas de Programación.
- Gaul, M., Massé, M. y Sarmiento, N. (2013). Análisis de Alice para la enseñanza básica de la programación. In *VIII Congreso de Tecnología en Educación y Educación en Tecnología*. <https://sedici.unlp.edu.ar/handle/10915/27578>

Gómez, E. (2001). Fundamentos de Programación. Escuela Politécnica del Ejército. <https://sites.google.com/a/espe.edu.ec/fundamentos-de-programacion/>

Martín, C., Urquía, A. y Rubio, M. (2021). *Lenguajes de programación*. Editorial UNED. <https://n9.cl/ax4ydd>

Morales, M. V. (2022). *Evidencias empíricas en el proceso de enseñanza de la programación. Una aproximación usando ALICE en el contexto universitario* (Doctoral dissertation, Universidad de Sevilla). <https://dialnet.unirioja.es/servlet/tesis?codigo=309459>

Patrón, C. E. (2022). El aula invertida, estrategia para la mejora del proceso enseñanza-aprendizaje. *Formación Estratégica*, 4(01), 32-47. <https://formacionestrategica.com/index.php/foes/article/view/50/27>

Ramírez, M., Castillo, M., Garza, J. J., García, L. y Vargas, J. A. (2011). “alice”: un entorno diferente para aprender programacion orientada a objetos. *CienciaUAT*, 6(2), 64-68. <https://www.redalyc.org/pdf/4419/441942926010.pdf>

Ruiz, A. B. M. (2010). Formación Docente en TICS. ¿están los docentes preparados para la (R) evolución TIC?. *International Journal of Developmental and Educational Psychology*, 4(1), 35-44. <https://www.redalyc.org/articulo.oa?id=349832327003>

Solano, J. (2011). Lenguajes de Programación. Temario. Lima: Universidad Nacional de Ingeniería, Facultad de Ciencias, 50-54. <https://n9.cl/owdks>

Tadeu, P. (2020). La competencia científico-tecnológica en la formación del futuro docente: algunos aspectos de la autopercepción en

respeto a la integración de las TIC en el aula. *Educatio Siglo XXI*, 38(3 Nov-Feb), 37-54. <https://revistas.um.es/educatio/article/view/413821>

Universidad Carnegie Mellon [CMU]. (2014). Alice. <https://www.alice.org/>

Vecino, O. y Seleey, (2009). d. a. enseñando programación orientada a objetos usando un ambiente 3D. https://www.rlcu.org.ar/recursos/ponencias_IX_encuentro/Monroy_Vecino.pdf

Capítulo 5

Fundamentos de programación en Scratch para el desarrollo de habilidades cognitivas en niños de Educación primaria

Medina Vaca Jhonny Alexander, Ortega Medina Maria Alexandra, Yaguana Cumbicos Carlos Manuel.

Universidad Nacional de Loja.

jhonny.medina@unl.edu.ec, maria.a.ortega.m@unl.edu.ec, carlos.yaguana@unl.edu.ec

Introducción

En la era digital actual, comprender los fundamentos de la programación se ha convertido en una habilidad indispensable; este conocimiento permite a los estudiantes desenvolverse en un entorno tecnológico avanzado y establecer una base sólida para el crecimiento personal y profesional. Al enfocarse en la metodología y estructura, los fundamentos de programación sirven como una guía esencial hacia los principiantes en el arte de la codificación. Es crucial que los alumnos de primaria reciban una introducción que esté adaptada a su nivel de comprensión, evitando así la complejidad que implicaría el aprendizaje completo de un lenguaje de programación.

En el marco de este capítulo, se ha seleccionado Scratch como la herramienta de programación ideal para introducir a los niños al mundo de la codificación. Su diseño basado en bloques y su interfaz amigable hacen de Scratch una plataforma accesible que estimula un aprendizaje lúdico y afectivo. Es fundamental señalar que la finalidad de la presente investigación es poder mostrar los beneficios de familiarizar la programación desde niveles académicos primarios; mediante la

herramienta Scratch, se busca no solo enseñar programación de manera atractiva, sino también fomentar el desarrollo de habilidades cognitivas esenciales que enriquecerán su proceso educativo.

La tecnología en la educación

La tecnología en la educación representa una revolución continua que proporciona métodos y herramientas vanguardistas para enriquecer la enseñanza y el aprendizaje. Mediante la implementación de plataformas interactivas, recursos digitales y dispositivos avanzados, los docentes tienen la capacidad de diseñar experiencias educativas más atractivas y adaptadas a las necesidades individuales, como ratifica Patrón (2022). En este sentido, brindan al docente la posibilidad de transformar el aula en un entorno dinámico de acuerdo a las necesidades individuales.

No obstante, la eficiencia de la tecnología depende de la competencia digital de los docentes. Laffey (2004), citado por Casado (2023), señala que muchos no están suficientemente preparados para integrarse efectivamente en el aula. Por ello, Tadeu (2020) sostiene que los educadores desarrollen habilidades digitales básicas y reconozcan sus limitaciones para garantizar un uso adecuado. No se trata solo de manejar herramientas digitales, sino de comprender su valor en el proceso educativo. Si los docentes utilizan la tecnología sin un conocimiento profundo, corren el riesgo de no aprovechar todo su potencial y restringiendo así las posibilidades de aprendizaje.

Fundamentos Generales de la Programación

La programación es un proceso esencial que facilita a las computadoras y otros dispositivos electrónicos realizar tareas específicas; esta disciplina también implica la conceptualización y el diseño de soluciones innovadoras. Al introducir los Fundamentos Generales de la

programación, se exploran los conceptos clave que forman la esencia de esta habilidad crítica. Según Gómez (2001), la programación es una “asignatura básica que permite crear programas que exhiban un comportamiento deseado”. Por ende, el proceso de escribir código requiere conocimientos en diferentes áreas, así como el dominio del lenguaje con la finalidad de garantizar el funcionamiento adecuado del código a ejecutar.

Por tanto, la programación se define como el proceso de ejecutar algoritmos mediante códigos que brindan al computador instrucciones precisas para realizar diversas acciones. Es importante destacar que esto se logra a través de software conocido como lenguajes de programación, además, es esencial considerar que un algoritmo consiste en una secuencia ordenada de pasos para resolver un problema. Para reforzar esta idea, Niklaus (s.f.), citado en Gómez (2001), explica que un programa se compone de algoritmos y estructuras de datos, lo que permite a la computadora procesar las instrucciones codificadas por el usuario y llevar a cabo la tarea requerida; es decir, transformar ideas en acciones digitales que amplían nuestras capacidades y resuelven dificultades complejas.

Habilidades cognitivas

Definir el concepto de cognición es fundamental, dado que su estudio ha impulsado numerosas investigaciones. En este trabajo, se considera una disciplina importante porque se enmarcará en el desarrollo de las habilidades cognitivas en niños de educación primaria. La palabra “cognición” deriva del latín “cognitivo” que significa conocimiento alcanzado mediante el ejercicio de las capacidades mentales; se explica como función o como estructura que denota el proceso donde las personas adquieren conocimiento (Rivas, 2008). Según lo expuesto

por el autor, la programación serviría como un vehículo para que los estudiantes amplíen su conocimiento y fortalezcan sus habilidades mentales, aprovechando los procesos que esta actividad involucra.

Por otra parte, al hablar de habilidades cognitivas, también llamadas habilidades intelectuales o del pensamiento, se hace referencia a las habilidades que son necesarias al ser humano para realizar diversas tareas. Reed (2007) citado en Cornejo et al. (2016), sostiene que estas habilidades son facilitadoras del conocimiento, puesto que son responsables de adquirirlo y luego recuperarlo para ser utilizado posteriormente. Por esta razón, es necesario potenciarlas desde edades tempranas de esta manera los niños no tendrán mucha dificultad en solucionar problemas cotidianos que se les presenten.

Adentrándose en este tema, Piaget (1980) enfatiza que la teoría de las habilidades cognitivas se centra en nuestra percepción, adaptación y manipulación del entorno. Aunque es ampliamente reconocida por sus etapas de desarrollo, su esencia se encuentra en entender cómo los individuos adquieren, construyen y aplican el conocimiento con el tiempo. Según Piaget, el desarrollo cognitivo es el resultado de la interacción del niño con su entorno, el cual se transforma significativamente a medida que crecen. Esto se evidencia en las cuatro etapas que propone: sensoriomotora, preoperacional, operaciones concretas y operaciones formales.

Siendo así, es importante tomar en cuenta el rango de edad y nivel académico para determinar en qué etapa del desarrollo cognitivo se encuentra el niño; para esta investigación se optaría por tomar como referencia dos etapas del desarrollo cognitivo Piaget. La primera, la etapa de operaciones concretas se desenvuelven de los 7 a 11 años, donde el niño aprende operaciones lógicas y asocia su pensamiento al mundo real, caracterizándose como niño práctico. La segunda, la etapa

de operaciones formales, se inicia entre los 11 a 12 años y se distingue por la capacidad del niño para emplear lógica avanzada, razonamiento científico y proporcional.

Programación en bloques mediante la herramienta Scratch para el desarrollo de habilidades cognitivas

Iniciar en el mundo de la programación puede presentar desafíos, especialmente al abordar lenguajes de programación avanzados como Java o C++. Sin embargo, Páez (2019) alude que, al utilizar herramientas como Scratch, se facilita la comprensión de conceptos matemáticos y computacionales eficientemente incorporados en el programa, tales como la interactividad, las condiciones lógicas, el manejo de coordenadas y variables, entre otros.

De este modo, Scratch es una plataforma gratuita e intuitiva con un enfoque de programación visual que elimina la necesidad de escribir código, lo que minimiza los errores de tecleo. Vázquez y Díez (2021) indica que esta herramienta es versátil para usuarios de todas las edades, G la posibilidad de crear una amplia gama de proyectos y actividades personalizadas, desde juegos hasta animaciones y simulaciones. Además, Scratch promueve un aprendizaje interdisciplinario, animando a los estudiantes a integrar conocimientos de diferentes materias mientras desarrollan habilidades fundamentales como el razonamiento crítico, la resolución de problemas y la colaboración, todo dentro de un espacio creativo y de apoyo mutuo.

Agregando a lo anterior, la herramienta de programación en bloques Scratch se considera como un nuevo ambiente de codificación que utiliza metáfora de piezas encajables para animar los objetos en pantalla. Esta aproximación convierte la programación en una actividad más accesible y estimulante, promoviendo la creatividad y enfocándose

en el desarrollo de habilidades de pensamiento lógico y computacional. Como expresa Pascual (2015), Scratch es un lenguaje de programación desarrollado por un grupo de investigadores del Instituto de Tecnología de Massachusetts (MIT), y se caracteriza por su entorno visual y multimedia que permite crear programas de forma intuitiva mediante una interfaz gráfica, convirtiéndola en una herramienta interactiva y dinámica.

Por otra parte, según Sormenezko (2010) menciona que “Scratch ayuda a comprender fácilmente conceptos matemáticos e informáticos que están muy bien integrados en el programa, como son los procesos interactivos (bucles), los criterios condicionales (si, entonces, si-no), las coordenadas en un plano y las variables”. Por ello, se considera adecuado para iniciar a estudiantes de primaria en la programación, ofreciendo una experiencia atractiva y libre del temor a los códigos, gracias a su sistema de bloques.

También, Pérez et al. (2020) argumentan la importancia de reconocer que la herramienta Scratch ayuda al desarrollo de habilidades cognitivas mediante la experimentación de forma creativa, fortaleciendo el pensamiento lógico y la comprensión del funcionamiento computacional, sino que también contribuye significativamente al crecimiento intelectual con las competencias necesarias para enfrentar desafíos de manera efectiva.

Actualmente, Scratch se ha convertido en una herramienta global para la enseñanza de la programación, utilizada por personas de todas las edades en más de 30 países y en distintos contextos educativos. Un estudio de Sormenezko (2010), basado en las estadísticas del MIT, menciona que se registran casi 10 millones de proyectos realizados en múltiples idiomas, creados por unos 7 millones de usuarios de entre 4 y 75 años. Respecto a ello, este estudio nos da un impulso de confianza en Scratch como una plataforma que promueve positivamente la programación.

Experiencias de usuarios al utilizar la herramienta de programación Scratch

Los experimentos de usuarios al utilizar la herramienta Scratch ofrecen una visión detallada sobre cómo esta plataforma está siendo empleada para potenciar habilidades de programación y fomentar la creatividad. Según un estudio realizado por Terneus et al. (2019) de la Universidad de las Fuerzas Armadas “ESPE”, involucró a 380 niños de entre 8 a 11 años y demostró que la programación en Scratch potencia el desarrollo cognitivo infantil. Se implementaron talleres formativos centrados en Scratch, lo que condujo a una participación entusiasta, otorgando a los participantes desplegar su creatividad y aplicar principios de programación en sus proyectos.

De acuerdo a los resultados del estudio de Terneus et al. (2019), se argumenta que son una prueba contundente del impacto social positivo que tiene la introducción temprana de la tecnología, beneficiando el desarrollo cognitivo de los niños en múltiples aspectos, como el pensamiento lógico-matemático y la adquisición de habilidades necesarias para navegar en la sociedad de la información.

De acuerdo al artículo de investigación realizado por Angamarca y Andrade (2022) de la Universidad Católica de Cuenca, que involucró a niños de 8 a 15 años del Cantón Cañar, se demostró que Scratch no solo mejora la habilidad de comprensión, sino que también suscita el pensamiento sistemático. Esto concluye que la herramienta educativa mejora el razonamiento de los niños de forma creativa. Este hallazgo subraya la importancia de enseñar programación y manejo de tecnología desde la primaria. En efecto, Scratch no solo promueve el desarrollo del pensamiento lógico en los niños, sino que también les enseña a organizar sus pensamientos de manera estructurada para crear secuencias de comandos de programación.

Presentación de la herramienta Scratch

Para facilitar la comprensión de los fundamentos de programación en Scratch, se ha creado una presentación interactiva que detalla los 8 bloques de programación esenciales. La presentación comienza con un índice intuitivo que expone los elementos esenciales para programar en Scratch, ilustrados en la Figura 5.1. Esta estructura ayuda al usuario interactuar significativamente con la herramienta, integrando la información directamente dentro de la misma.



Figura 5.1. Contenidos básicos de Scratch. Fuente: Adaptado a partir de MIT (2012).

Es importante destacar que la interfaz de Scratch está diseñada para simplificar la navegación y el aprendizaje, como se puede apreciar en la Figura 5.2 presenta una ventana unificada con paneles claramente definidos que mantienen los elementos esenciales al alcance del usuario. En el lado izquierdo, la paleta de comandos se organiza de manera intuitiva, con botones que facilitan la selección de bloques de programación. Esta disposición contribuye a una experiencia de usuario fluida y centrada en la creación y el aprendizaje.



Figura 5.2. Interfaz de Scratch. Fuente: Adaptado de Fuente: Adaptado a partir de MIT (2012).

En cuanto al Bloque de Movimiento, se encuentran las instrucciones que propicia mover el objeto en el plano x-y, girar en sentido horario o antihorario, cambiar la dirección del objeto hacia la derecha, izquierda, arriba o abajo, posicionar el objeto en un lugar específico, y hacer que rebote si toca algún borde, entre otras funcionalidades, tal como se ilustra en la Figura 5.3. Estas acciones son fundamentales para la creación de proyectos dinámicos y la comprensión de la geometría espacial en un contexto de programación.



Figura 5.3. Bloques de Movimiento de Scratch. Fuente: Adaptado a partir de MIT (2012).

Del mismo modo, en el bloque de apariencia, visible en la Figura 5.4 se brinda la posibilidad de cambiar de disfraz al objeto, hacer que diga algún comentario, aplicar efectos digitales a la imagen de disfraz, cambiar de tamaño, mostrar o esconder el objeto, y manejar su posición en capas. Por otro lado, el Bloque de sonido facilita la incorporación de sonidos a partir de archivos o notas musicales, potenciando así la interactividad y el encanto sensorial de los proyectos en Scratch.

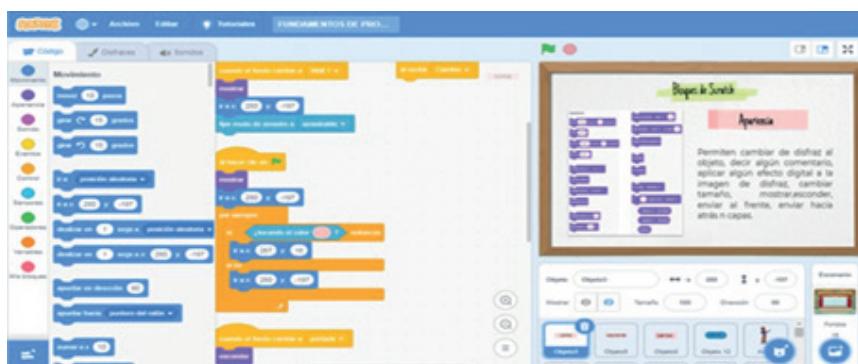


Figura 5.4. Bloque de Apariencia de Scratch. Fuente: Adaptado a partir de MIT (2012).

Con relación al Bloque de Eventos en Scratch, es esencial para la interactividad, ya que permite a los objetos responder a acciones específicas, como se muestra en la Figura 5.5. Esta funcionalidad es clave para la creación de programas dinámicos y juegos interactivos, ya que los objetos pueden reaccionar a eventos como clics del ratón, presiones de teclas o mensajes recibidos de otros objetos. Además, el Bloque de Eventos facilita la ejecución de instrucciones al comienzo de un programa o animación con solo presionar la Bandera Verde, lo que simplifica el proceso de inicio.



Figura 5.5. Bloque de Eventos de Scratch. Fuente: Adaptado a partir de MIT (2012).

Acerca del Bloque de Sensores facilita la interacción con el entorno del proyecto. Como se observa en la Figura 5.6, estos bloques proporcionan que los objetos detecten colores específicos, reaccionen a teclas presionadas y sigan la posición del ratón, tanto en coordenadas x-y como en distancia al puntero. Además, este conjunto de instrucciones es capaz de capturar entradas de dispositivos externos, ampliando así las posibilidades de interactividad.

Por ejemplo, se puede formular una pregunta y registrar la respuesta del usuario en una variable, enriqueciendo la experiencia de usuario más rica y personalizada.



Figura 5.6. Bloque de Sensores de Scratch. Fuente: Adaptado a partir de MIT (2012).

Estas características hacen del Bloque de Sensores un componente esencial para proyectos que requieren una mayor sensibilidad y respuesta al entorno.

En contraste, el Bloque de Operadores en Scratch, identificado por su color celeste como se muestra en la Figura 5.7, es fundamental para la realización de operaciones lógicas y matemáticas esenciales. En este bloque, los usuarios pueden ejecutar operaciones lógicas como and, or y not, así como operaciones matemáticas básicas que incluyen suma, resta, multiplicación, división, raíz cuadrada, operaciones logarítmicas y trigonométricas básicas, mayor, menor e igual. Estas operaciones son cruciales en el desarrollo de proyectos que requieren cálculos precisos y lógica condicional, donde los usuarios pueden explorar y aplicar conceptos matemáticos en un entorno interactivo y visual.



Figura 5.7. Bloque de Operadores de Scratch. Fuente: Adaptado a partir de MIT (2012).

Mientras tanto, en el Bloque de Variables, caracterizado por el color naranja oscuro como se detalla en la Figura 5.8 permite crear variables, que almacenan un valor. Estos Bloques Propios o Funciones son un grupo de instrucciones caracterizadas por un color rosa.

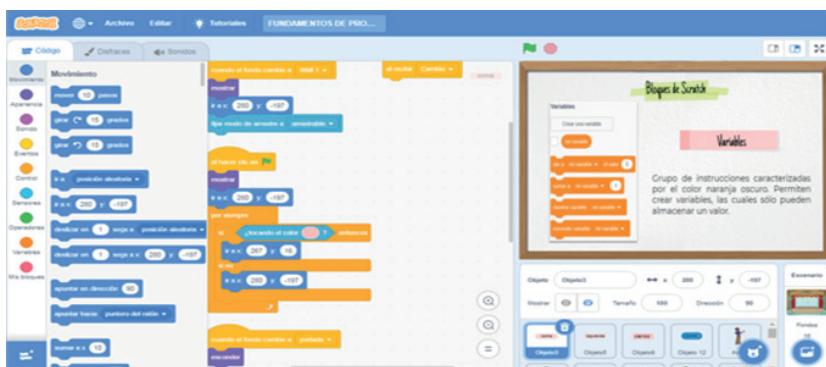


Figura 5.8. Bloque de Variables de Scratch. Fuente: Adaptado a partir de MIT (2012).

La integración de estos bloques en Scratch proporciona un entorno de programación visual que es ideal para introducir a los niños en la programación. Al utilizar estos bloques, pueden aprender conceptos fundamentales de programación, despertando su interés y curiosidad. Este enfoque práctico y visual es coherente con las teorías de desarrollo cognitivo de Piaget, que enfatizan la importancia de la experiencia concreta y activa en el aprendizaje, especialmente en las etapas de desarrollo temprano.

Reflexiones

La habilidad de los educadores en el manejo de herramientas digitales es fundamental en el contexto educativo actual. Esta competencia no solo eleva la calidad de la enseñanza y el aprendizaje, sino que también prepara a los estudiantes al mundo digitalizado en el que vivimos. Por ello, es importante incorporar una materia que esté destinada a enseñar los fundamentos de la programación a niños de educación primaria y, así fortalecer sus habilidades, permitiendo desarrollar un pensamiento organizado y fortalecer sus habilidades cognitivas.

En consecuencia, la introducción a la programación debe abordarse de manera dinámica, destacando el valor de herramientas como Scratch, que se distinguen por su facilidad de comprensión y adaptabilidad al nivel educativo. Scratch es una herramienta muy interesante para introducir a los niños a la programación por bloques, ya que desarrollan habilidades tanto cognitivas como lógicas, facilitando desenvolverse adecuadamente dentro del entorno educativo como en su diario vivir, y enmarcando la solución a los problemas que se puedan presentar en su desempeño estudiantil y personal.

Bibliografía

- Angamarca, O. y Andrade, D. (2022). Enseñanza de programación a niños de edad escolar utilizando Scratch para mejora del razonamiento lógico. *Pro Sciences: Revista de Producción, Ciencias e Investigación*, 6(42), 111-121. <https://journalprosciences.com/index.php/ps/article/view/475/524>
- Casado, C. (2023). La enseñanza de la programación en la infancia: estudio de casos de la situación en Cataluña. <https://openaccess.uoc.edu/handle/10609/147632>
- Cornejo, M. A. N., Agreda, O. O. y Iturralde, L. N. P. (2016). *Desarrollo de habilidades cognitivas usando software interactivo*. In *Memorias del segundo Congreso Internacional de Ciencias Pedagógicas: Por una educación innovadora, para un desarrollo humano sostenible* (pp. 1236-1245). Instituto Superior Tecnológico Bolivariano. <https://dialnet.unirioja.es/servlet/articulo?codigo=7199844>
- Gómez, E. (2001). *Fundamentos de Programación*. Escuela Politécnica del Ejército. <https://sites.google.com/a/espe.edu.ec/fundamentos-de-programacion/>
- Instituto de Tecnología de Massachusetts [MIT]. (2012). *Scratch* (2021).<https://scratch.mit.edu/>
- Pascual, J. (2015). Scratch, programación sencilla y gratis para niños y mayores. <https://n9.cl/azei7b>
- Páez, F. (2019). Desarrollo de competencias a través del uso de las herramientas Scratch y Arduino en niños y jóvenes pertenecientes a zonas urbanas marginales del distrito

- Metropolitano de Quito. Revista Vínculos, 4(2). <https://journal.espe.edu.ec/ojs/index.php/vinculos/article/view/1554>
- Patrón, C. E. (2022). El aula invertida, estrategia para la mejora del proceso enseñanza-aprendizaje. *Formación Estratégica*, 4(01), 32-47. <https://formacionestrategica.com/index.php/foes/article/view/50/27>
- Pérez, H. O., Roig, R. y Jaramillon, L. (2020). Uso de SCRATCH en el aprendizaje de Programación en Educación Superior.
- Piaget, J. E. A. N. (1980). Teoría del desarrollo cognitivo de Piaget. *Creative Commons Attribution-Share Alike*, 3(1), 1-13. <https://terapia-cognitiva.mx/wp-content/uploads/2015/11/Teoria-Del-Desarrollo-Cognitivo-de-Piaget.pdf>
- Rivas, M. (2008). *Procesos cognitivos y aprendizajes significativos*. Comunidad de Madrid - España. CONSEJERÍA DE EDUCACIÓN. Viceconsejería de Organización Educativa. <https://n9.cl/bhu2pt>
- Sormenezko, I. (2010). Guía Didáctica para Profesores Scratch. <https://n9.cl/nmrq>
- Tadeu, P. (2020). La competencia científico-tecnológica en la formación del futuro docente: algunos aspectos de la autopercepción en respeto a la integración de las TIC en el aula. *Educatio Siglo XXI*, 38(3 Nov-Feb), 37-54. <https://revistas.um.es/educatio/article/view/413821>
- Terneus, F., Tobar, J., Loza, D. y Naranjo, F. (2019). Desarrollo de competencias a través del uso de las herramientas Scratch y Arduino en niños y jóvenes pertenecientes a zonas urbanas marginales del Distrito Metropolitano de Quito. *Revista*

Vínculos ESPE, 4(2).<https://journal.espe.edu.ec/ojs/index.php/vinculos/article/view/1554/1172>

Vázquez, E. y Díez, P. (2021). Análisis de la percepción del estudiante sobre el empleo didáctico de Scratch en el aula. In IN-RED 2021: *VII Congreso de Innovación Educativa y Docencia en Red* (pp. 94-103). Editorial Universitat Politècnica de València. <https://riunet.upv.es/handle/10251/175038>

Capítulo 6

MBlock: bases técnicas y metodológicas para iniciar en el mundo de la programación a los estudiantes de Educación Básica

Labanda Jaramillo Milton, Ambuludi Lopez Alex Manuel, Salinas Patiño Jhandry Michael

Universidad Nacional de Loja

miltonlab@unl.edu.ec, alex.ambuludi@unl.edu.ec, jhandry.salinas@unl.edu.ec

Introducción

En la actualidad la complejidad de los programas que se desarrollan produce la necesidad de iniciar a los estudiantes de primaria o secundaria a aprender a utilizar técnicas de programación desde temprana edad comenzando con métodos de arrastrar objetos o de ordenar bloques para el desarrollo de un programa educativo.

La presente investigación tiene como objetivo introducir a los estudiantes en el mundo de la programación con un acercamiento respecto a las características y métodos necesarios para comenzar a programar, ya que dicho proyecto tiene como propósito identificar los factores que influyen en el estudiante al momento de programar, por medio de la resolución de problemas cuya solución se pueda escribir en términos de un algoritmo, es decir en dar una serie de instrucciones secuenciales para ejecutar acciones o programas. Posteriormente este algoritmo es traducido a un lenguaje de programación y ejecutado en una computadora.

Programación Orientada a Objetos

La Programación Orientada a Objetos (POO) es reconocida como el paradigma más adecuado para mejorar el desarrollo de los alumnos al momento de crear o mejorar un software (García, 2001); ya que favorece dos aspectos importantes como son la reutilización y la extensibilidad del código dentro de plataformas web que permiten descargar o interactuar con el estudiante en el momento que desarrolla las actividades.

Por ello, se cree conveniente presentar la herramienta de autor mBlock que tiene como propósito ser introducido en el ámbito educativo para aprender a programar que se utiliza la programación mediante bloques de código (Crespo, 2017). Es un Entorno de Desarrollo Integrado (IDE) multiplataforma, esto debido a la sencillez de instalación y uso con distintos tipos de sistemas operativos que también cuenta con una versión web. Además, de trabajar con código de bloques también puede ser usado el lenguaje Python dentro del desarrollo de un programa. Es gratuita y responde a la filosofía del software libre para la enseñanza de estudiantes y docentes.

¿Qué es mBlock?

mBlock es un entorno de programación gráfico educativo que permite aprender a programar utilizando la programación mediante bloques de código (Puga, s.f.), pensado para introducir la robótica de una forma atractiva, sencilla y divertida al proceso educativo en centros de formación.

Es una variante del software libre llamado Scratch 3.0 que permite trabajar con la programación visual o por bloques donde el usuario une bloques que le permiten construir un programa. Además, cuenta con una variedad de categorías de bloques disponibles que permiten modificar

el comportamiento de los objetos en los bloques programados, por ende, Páez (s.f.), menciona que mBlock está compuesto por categorías de instrucciones como se indica en la Tabla 6.1, que se presenta a continuación:

Tabla 6.1. Instrucciones de mBlock.

Objeto	Concepto
Movimiento	Conjunto de instrucciones relacionadas con el control del movimiento de cualquier personaje del escenario.
Apariencia	Instrucciones orientadas a modificar el aspecto de los personajes de nuestra aplicación.
Sonido	Conjunto de instrucciones relacionadas con la elaboración de aplicaciones musicales, emitiendo sonidos y notas musicales.
Control	Las instrucciones son imprescindibles para crear la lógica de nuestros programas. Incluyen condicionales, bucles y llamadas de procedimientos.
Sensores	Instrucciones de iteración con el ratón, el teclado, sonidos y los personajes.
Operadores	Operaciones matemáticas, lógicas y con cadenas de texto.
Variables	Instrucciones para el almacenamiento y gestión de datos.
Instrucciones de Programación	Las instrucciones de cada grupo corresponden a instrucciones de programación.
Editor	Es la parte principal donde estructuramos y programamos nuestro programa.
Programas	Se compone de todas las instrucciones que hace funcionar el código que programemos.
Disfraces	Cada objeto puede tener diferentes apariencias o disfraces para utilizar a lo largo de nuestro programa.
Sonido	También es posible añadir o grabar sonidos y guardarlos para futuros usos.
Objetos	Cada uno de los elementos del proyecto que podemos programar (personajes, animales, plantas, etc.).

Fuente: Adaptado de Páez (s.f.).

Educación STEAM

En la herramienta de autor mBlock se prioriza la aplicación de las TIC (Tecnologías de Información y Comunicación) dentro del proceso de enseñanza-aprendizaje, tomando en cuenta la educación STEAM que se refiere en sus siglas en español a la Ciencia, Tecnología, Ingeniería, Artes y Matemáticas. WebRobótica (2019) menciona que la educación STEAM anima a los estudiantes a desarrollar conocimientos en forma interdisciplinaria y de manera lúdica explorando y resolviendo problemas de la vida real, mejorando el pensamiento lógico, creatividad, habilidades de trabajo en equipo, etc.

Tecnologías que integra mBlock

Scratch

La programación por bloques integrada en mBlock es un lenguaje visual y modular basado en Scratch 3.0. De acuerdo a Martínez (2023), se organiza en bloques que poseen diferentes funciones y acciones agrupando bloques con opciones seleccionables mediante menús, basta con mirar detenidamente el conjunto de bloques para comprender las funciones que cumplen creando acciones complejas que generan efectos y movimientos a tus proyectos.

En términos de lo mencionado, existe un fuerte impacto del uso de Scratch en la enseñanza de la programación, donde su utilización determina el mejoramiento en los niveles de pensamiento computacional de los estudiantes (García, 2022). Por ende, esta herramienta permite diseñar proyectos orientados desarrollar destrezas con respecto a procesos cognitivos relacionados a la lógica y resolución de problemas.

Robótica Educativa

La robótica analizada en el ámbito educativo, se ha formulado el término robótica educativa, según Domínguez et al. (2018), es un sistema de enseñanza interdisciplinar que dentro de mBlock permite a los estudiantes a desarrollar tareas de programación sencillas utilizando los elementos necesarios para programa y construir un programa, desarrollando una de las habilidades cognitivas básicas del pensamiento lógico, el pensamiento computacional como un medio que mejore la calidad educativa y la motivación de los estudiantes.

Asimismo, la robótica educativa amplia la integración de diferentes áreas de conocimiento, por tanto, desde una revisión sistemática desarrollada por Raposo et al. (2022) arroja resultados con respecto al uso de la robótica en proyectos vinculados a las áreas STEAM para potenciar el pensamiento computacional en los estudiantes; siendo indispensable esta habilidad para estar a la vanguardia de las nuevas tecnologías.

Inteligencia Artificial (IA)

La aplicación de la inteligencia artificial en la educación, consiste fundamentalmente en emplear las técnicas de la IA para el desarrollo de sistemas de enseñanza inteligentes (Torres, 2016); que sean capaces de adaptarse a las características y ritmo de aprendizaje de cada estudiante permitiendo reconocer la voz, texto impreso y escrito, la emoción de los rostros en las personas por medio de los componentes de multimedia que poseen los dispositivos tecnológicos utilizados.

En el mismo orden de ideas, la IA impulsa nuevos espacios para transformar la educación, aplicando principalmente su uso en robots educativos para mejorar el aprendizaje de los estudiantes mejorando la calidad de la evaluación para que sea automática y objetiva las

tareas detectando factores significativos que hacen atractivas las clases (Martínez et al., 2023). De esta manera, se fortalece los procesos pedagógicos utilizando los beneficios y facilidades que brindan las tecnologías en permanente crecimiento.

Python

mBlock integra a Python como un lenguaje de programación, interpretado de tipado dinámico haciendo énfasis en una sintaxis que favorezca un código legible ya que soporta parcialmente la orientación a objetos, programación imperativa y programación funcional, permite escribir el código con características como la finalización, resultado de sintaxis y sangría automática lo cual contribuye una forma más sencilla para aprender a codificar desde programas básicos hasta más complejos.

Se caracteriza por ser interpretado debido a que se ejecuta sin necesidad de ser procesado por el compilador y se detectan los errores en el tiempo de ejecución (Amazon, 2023). De este modo, Santander Open Academy (2021) agrega que es multiparadigma, ya que soporta programación funcional, programación imperativa y programación orientada a objetos; tipado dinámico, puesto que las variables se comprueban en tiempo de ejecución; multiplataforma: disponible para plataformas de Windows, Linux o MAC; finalmente de tipo gratuito, ya que no dispone de licencia para programar.

Arduino

La interfaz es muy amigable e intuitiva. Suárez (2020), señala que Arduino usa bloques previamente definidos para dar disposiciones al robot y conociendo la programación se podrá trabajar con el entorno de Arduino, es una plataforma electrónica de código abierto basada en hardware y software fáciles de usar. Específicamente, Willging et al.

(2017) manifiestan que se trata de un Sistema Embebido (SE), es decir, combinaciones en conjunto con un sistema mecánico o electrónico diseñado para cumplir una función específica como puede ser la creación de código para la ejecución de órdenes al robot

En función a lo mencionado anteriormente, permite programar tus robots de forma inalámbrica mediante tecnología bluetooth o 2.4G, traducir los bloques de Scratch a código fuente de Arduino y probar en tiempo real el programa que se realizó en Scratch sin necesidad de grabarlo en la placa y una vez probados puedes grabarlo de forma permanente en tu robot educativo, tales como:

- Makeblock
- Robot Educativo mBot.
- Kit básico de Makeblock: Starter Kit.
- Robot más avanzado con muchos sensores mBot Ranger.
- Ultimate Kit.

Recurso educativo “El rango y combinaciones simples” desarrollado en mblock

El recurso educativo titulado “El rango y combinaciones simples”, está dirigido para estudiantes de Educación General Básica (EGB) de la materia de matemáticas con los temas establecidos en la Unidad 6 del libro base propuesto por el Ministerio de Educación como se observa en la Figura 6.1. Se lo desarrolló en el lenguaje de programación por bloques mBlock en su versión 3.4.12, es de tipo ejercitador ya que permite manejar la temática de una forma más cómoda y proporcionar retroalimentación, su navegación es jerárquica, pues el contenido se desarrolla de forma ramificada, es decir se puede visitar los contenidos de forma separada con el fin de mejorar proceso de enseñanza-aprendizaje.



Figura 6.1. Recurso Educativo El rango y Combinaciones Simples.

Al ingresar al Recurso Educativo encontraremos la respectiva portada y el menú donde se accederá a los botones de aprendamos, juguemos y la información. Al ingresar al apartado de “Aprendamos” se encuentra la información de la materia, el objetivo, destreza, eje de aprendizaje y el indicador de evaluación al que va dirigido el recurso, seguidamente aparecerá un nuevo subíndice la misma que cuenta con la respectiva portada y botones del contenido de la materia de matemáticas para 5to año de EGB con el tema rango y combinaciones simples, además se encontrará el botón de bibliografía y el botón de regresar a la pantalla principal.

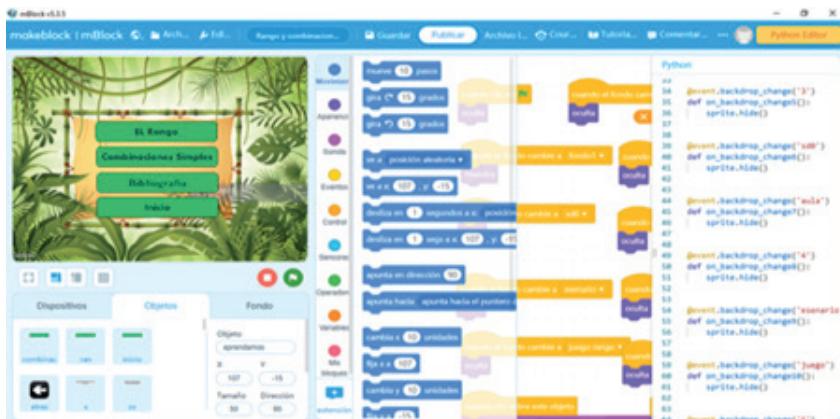


Figura 6.2. Recurso Educativo El rango y Combinaciones Simples.

FSeguidamente en la Figura 6.2 permite dirigirse al apartado de “Juguemos” ahí se encontrarán dos actividades y el botón de regresar, al momento de seleccionar alguna actividad aparecerá indicaciones que servirán de ayuda para desarrollarla, la primera actividad consiste en ir arrastrando las soluciones a los ejercicios expuestos para buscar su verificación.



Figura 6.3. Recurso Educativo El rango y Combinaciones Simples. Fuente: Autoría propia.

La siguiente actividad como se presenta en la Figura 6.3, es de memoria en donde tendrá que emparejar las figuras semejantes, cabe recalcar que cada actividad contiene un botón de ayuda que permitirá al estudiante desarrollar de mejor manera. Al finalizar permite acceder al apartado de “Información” donde se muestra la información del autor que consta de: el nombre del autor, nombre de la institución y ciclo en cual se elaboró el recurso educativo, además incluye un botón para regresar a la página principal.

Reflexiones

En conclusión, con la ayuda de la herramienta mBlock permite a los niños puedan obtener bases técnicas y metodológicas para introducirse en la programación de manera sencilla ya que permite trabajar con una interfaz gráfica, diferenciándose de Scratch que permite trabajar programando robots mediante una conexión con Arduino que puede ser comprendida con facilidad por los estudiantes, además y puedan desarrollar programas básicos o complejos de acuerdo al nivel de estudio donde se encuentren.

Adicionalmente, con la implementación de la educación STEAM en el proceso educativo utilizando la herramienta mBlock se puede trabajar dinámicamente de tal manera que los estudiantes puedan desarrollar el pensamiento lógico a través de la programación por bloques y Python con el fin de motivarlos e interesarlos por las diferentes metodologías y lenguajes que existen para programar, desarrollando actividades académicas donde se pueda reforzar las habilidades lúdicas, el pensamiento lógico y creatividad, que permitan mejorar la calidad del proceso de enseñanza aprendizaje desarrollado en escuelas y colegios.

MBlock se presenta como una herramienta fundamental para iniciar a los estudiantes de Educación Básica en el mundo de la programación, proporcionando bases técnicas y metodológicas que complementan la digitalización de contenidos académicos, generando curiosidad en los estudiantes al introducir nuevos temas en el aula. Las tecnologías involucradas permiten por tanto desarrollar nuevas destrezas de aprendizaje mediante la implementación de elementos atractivos y la creación de escenarios complejos, potenciando así el pensamiento computacional a través de actividades afines.

Bibliografía

- Amazon. (2023). ¿Qué es Python? Herramientas para desarrolladores. Amazon Web Services. <https://aws.amazon.com/es/what-is/python/>
- Crespo, E. (2017). Taller de Programación de Arduino por Bloques con mBlock. Aprendiendo Arduino. <https://n9.cl/h96fk>
- Domínguez, Y., Lázaro, M., Suarez, A., Martínez, P., y Beatón I. (2018). experiencias sobre robótica educativa y programación con mbot. <https://n9.cl/6lt55>
- García, J. (2001). ¿Es conveniente la orientación a objetos en un primer curso de programación? VII Jornadas de Enseñanza Universitaria de la Informática (JENUI'2001), Palma de Mallorca. <https://n9.cl/xjc7o>
- García, A. (2022). Enseñanza de la programación a través de Scratch para el desarrollo del pensamiento computacional en educación básica secundaria. Revista Academia y Virtualidad, 15(1), 161-182. <https://dialnet.unirioja.es/servlet/articulo?codigo=8433942>
- Martínez, M., Rigueira, X., Larrañaga, A., Martínez, J., Ocarranza, I. y Kreibel, D. (2023). Impacto de la inteligencia artificial en los métodos de evaluación en la educación primaria y secundaria: revisión sistemática de la literatura. Revista de Psicodidáctica, 28(2), 93-103. <https://www.sciencedirect.com/science/article/pii/S1136103423000114>
- Martínez, M. (2023). Qué es Scratch y para qué sirve. Desarrollo Web y Diseño Web. España. <https://profile.es/blog/que-es-scratch/>

Páez, C, (s.f.). mBlock N1 – Introducción. Robótica. Colombia. <https://n9.cl/dse7v>

Puga, G. (s.f.) Qué es mBLOCK. Tecno- Comunicación. Bajo licencia Creative Comms CC BY-NC-SA 4.0. <https://n9.cl/7xd5d>

Raposo, M., García O. y Martínez M. (2022). La robótica educativa desde las áreas STEAM en educación infantil Una revisión sistemática de la literatura (2005-2021). *Prisma Social: revista de investigación social*, (38), 94-113. <https://dialnet.unirioja.es/servlet/articulo?codigo=8532275>

Santander Open Academy. (2021). Python: qué es y por qué deberías aprender a utilizarlo. Santander Universidades. <https://www.santanderopenacademy.com/es/blog/python-que-es.html>

Suárez, R. (2020). Arduino. Tecnologías. Gobierno de Canarias. <https://www3.gobiernodecanarias.org/medusa/ecoblog/rsuagued/arduino/>

Torres, H, (2016). Inteligencia Artificial en la Educación. Universidad Nacional Abierta y a Distancia de México. <https://n9.cl/zt3j>

WebRobótica (2019). mBlock 5. Una poderosa plataforma para la educación STEAM. <https://www.web-robotica.com/robotica-educativa/mblock-5-una-poderosa-plataforma-para-la-educacion-steam>

Willging, P., Astudillo, G., Castro, L., Bast, S., Occelli, M. y Distel, J. (2017). Educación con tecnologías: la robótica educativa aplicada para el aprendizaje de la programación. In XIX Workshop de Investigadores en Ciencias de la Computación.

Capítulo 7

Programación por bloques: Caso de estudio en la herramienta Blockly

Labanda Jaramillo Milton Leonardo, Sarango Masache Alex Dario, Villamagua Vicente Sandra del Carmen.

Universidad Nacional de Loja.

alex.d.sarango@unl.edu.ec, miltonlab@unl.edu.ec, sandra.villamagua@unl.edu.ec

Introducción

El presente capítulo se centra en la programación por bloques, destacando su importancia en una sociedad más tecnológica. La educación moderna debe reflejar esta realidad, integrando la programación en el ámbito educativo desde una edad temprana mediante herramientas tecnológicas fáciles de usar, como las aplicaciones basadas en bloques; de esta manera, los estudiantes pueden aprender forma lúdica y accesible, facilitando así el desarrollo de habilidades cruciales en el siglo XXI.

En este caso, se centrará en el uso práctico de la herramienta de programación por bloques denominada Blockly, una aplicación que utiliza el lenguaje JavaScript, además, se puede implementar en las aulas, ya que permite a los estudiantes interactuar con actividades de programación estructuradas y adquirir conocimientos fundamentales de JavaScript. Su implementación es un paso hacia la integración efectiva de la tecnología en la educación.

Importancia de programar en el siglo XXI

En el mundo actual con el constante avance de la tecnología que cada día se está expandiendo en todos los ámbitos de la vida, es necesario conocer este mundo tecnológico, el aprender cómo funciona esta

sociedad plagada de tecnología nos hará personas más competentes y, para comenzar se requiere tener bases de cómo ejercer ese tipo de conocimientos; mientras en la comprensión y la manipulación es preciso entender los fundamentos de la programación.

En ese sentido, en el mundo del hoy y del futuro se van a requerir individuos capacitadas en programación. Por ello, resulta crucial fomentar su aprendizaje desde la niñez como alude González (2019) “En el siglo XXI, en la sociedad digital de información, es indispensable que los ciudadanos aprendan a “codificar” para ser productores digitales de información, no solo consumidores digitales de la misma” (p.3). En consecuencia, es importante que el sistema educativo entre en acción, facilitando su comprensión y posterior práctica.

En este aspecto, hay que saber el concepto de programar, que consiste dar instrucciones a una máquina para que realice un trabajo determinado. Kuz y Ariste (2022) mencionan que: “programar, esto es, diseñar algoritmos y definir el código que lo lleva a la práctica en lenguaje de computador” (p.121), es decir utilizar un lenguaje que el ordenador entienda para que realice una actividad determinada.

Por otra parte, según Agudelo y Salazar, (2020), la programación que consiste en la creación de algoritmos, que son secuencias estructuradas de instrucciones diseñadas para realizar tareas específicas o resolver problemas. Estos algoritmos se expresan luego en un lenguaje de programación, permitiendo el desarrollo de programas informáticos destinados a la resolución de problemas concretos. Además, entre los beneficios de implementarla en niños, es que está promueve el desarrollo del pensamiento computacional, una competencia clave en la era digital.

En cuanto al concepto del pensamiento computacional (PC), desde la posición de Wing citado por Quiroz et al. (2021), se describe que son

actividades que fomentan en los estudiantes habilidades que concedan comprender, interpretar y cambiar su entorno mediante el uso de conceptos informáticos. Por otro lado, Kuz y Ariste (2022) argumenta que la PC y la programación dotan al educando de las destrezas necesarias para adquirir conocimientos prácticos, esenciales para entender, analizar de manera crítica y tomar medidas en un entorno altamente influenciado por las tecnologías digitales.

Asimismo, este tipo de pensamiento a su vez se desarrolla con el uso de la programación, no solo desenvolverse en el manejo de la tecnología, sino también en otros aspectos de la vida (Astudillo et al., 2019). Por tal razón, la importancia, según los conceptos de los autores citados, de incluir la programación en la educación cataliza el desarrollo de competencias fundamentales como la resolución de problemas, imaginación, compresión, entre otros que permitan un mejor desempeño en el estudiante.

Programación en Educación

Los recursos tecnológicos ofrecen múltiples beneficios para la educación, tales como el fomento del desarrollo cognitivo en diversas áreas, la mejora del pensamiento lógico y la habilidad para clasificar objetos. En el ámbito de la programación, otorga a los usuarios identificar errores y explorar soluciones potenciales. Sánchez et al. (2020) destacan que, en la actualidad, los niños crecen inmersos en la tecnología, lo que les permite acostumbrarse a las Tecnologías de la Información y las Comunicaciones (TIC) desde temprana edad.

Considerando este tema, las tecnologías forman parte de la vida cotidiana y estos son medios de entretenimiento por lo cual, es crucial integrar las TIC en el entorno escolar, reconociendo su importancia como un recurso que complementa el trabajo del profesor. Agregando

a lo anterior, como señalan Cotik y Monte-verde (2016), citado por García et al. (2022), siguiendo esa perspectiva, corresponde al sistema educativo preparar a los estudiantes para su inserción en la sociedad; por ello, es cada vez más crucial incorporar tecnologías, con la finalidad de que puedan hacer funcionar sus procesos cognitivos.

En cambio, según la investigación de Sarmiento (2020), en la región de Latinoamérica ya existen proyectos para fomentar la programación, y por ende, el desarrollo del pensamiento computacional. Casos como Robot Inventor to Teach Algorithms - RITA, desarrollada por el grupo de investigación de la Universidad Nacional de La Plata en Argentina, y en el mismo país, el grupo de Innovación Educativa de la Universidad Nacional de La Pampa, hacen uso de las herramientas de programación con Lighthbot y Scratch, mientras en Chile utilizan Scratch y los robots de Lego. Todas las investigaciones impulsan el desarrollo del PC, aunque hay que recalcar que están destinados a estudiantes universitarios; sin embargo, se pueden adaptar para la enseñanza-aprendizaje en escuelas y colegios.

En el contexto de la programación por bloques, específicamente en el caso de Ecuador, Israel (2019) citado por Angamarca y Andrade (2022) señala que la programación ha mejorado la capacidad de los estudiantes para resolver problemas, organizar y corregir errores. Para introducir este enfoque en las aulas, es crucial adoptar una metodología centrada en motivar a los estudiantes y cultivar un interés por la programación desde los conceptos básicos, seguido de la aplicación práctica de los conocimientos adquiridos.

Además, aprender a través de la experimentación y el entorno permite a los estudiantes asimilar información, de manera especial a través del aprendizaje de los errores, fortaleciéndose las habilidades adquiridas

mediante juegos fomentan la programación, facilitando a los usuarios la mejora de sus capacidades para resolver problemas. Este enfoque beneficia no sólo el desarrollo de competencias fundamentales en los estudiantes, sino que además promueve un aprendizaje más profundo y significativo en el uso de herramientas como Blockly.

Asimismo, Gutiérrez (2022) menciona que hoy en día, la tecnología ha hecho que los juegos sean más accesibles en una variedad de dispositivos, como móviles, computadoras y consolas de videojuegos, contando con entornos visuales atractivos diseñados para captar la atención de los jugadores y motivarlos a avanzar y superar desafíos. Esta tendencia abre la puerta al uso de herramientas educativas que integran la programación de forma divertida, ofreciendo una experiencia de aprendizaje interactiva y entretenida mientras juegan.

Por ende, los beneficios de integrar la programación en las aulas son significativos, lo que hace imprescindible que los sistemas educativos modernos, con miras al futuro, fomenten su inclusión como un elemento esencial del currículo educativo. El futuro y el presente es saber programar; estos conocimientos y habilidades que se desarrollen por medio de la misma convertirán a las personas en más competentes y exitosas en esta sociedad tecnológica.

En este aspecto, el profesor es una pieza fundamental para ese proceso, pero debe ser la persona encargada de guiar, más no de ayudar, en el sentido de que no se convierta en el responsable de dar solución a los problemas; porque esa misión debe corresponderles a los mismos estudiantes, quienes construirán su propio conocimiento buscando soluciones a dificultades determinadas. Complementando a lo mencionado, Sánchez et al. (2020) concluye en su investigación que la implementación de aplicaciones de programación como Scratch Jr y el Scratch facilita la incorporación de los contenidos del plan de estudios en la formación mediante un enfoque de

aprendizaje colaborativo, lo que promueve el desarrollo del pensamiento lógico y creativo de los estudiantes.

En lo que respecta a este aspecto, la autora Sánchez, (2021) señala que los docentes no deben estancarse y tienen la responsabilidad de buscar la innovación. Esto implica el desafío de diseñar proyectos que incorporen recursos tecnológicos. Una estrategia efectiva para implementar tecnología educativa que promueva la programación de manera sencilla y accesible es el inicio con la programación por bloque.

Programación por bloques

La programación por bloques simplifica el proceso de codificación al utilizar bloques visuales en lugar de texto para representar comandos, permitiendo a los usuarios arrastrar y colocar estos bloques de manera que formen una secuencia lógica, similar a completar un rompecabezas. Como señalan Ahumada et al. (2019), “En los entornos de programación por bloques, las instrucciones están representadas por bloques. Un programa se construye arrastrando y encastrando bloques en el orden apropiado para un determinado fin” (p. 10). Al adoptar un enfoque lúdico, la programación por bloques convierte la tarea de programar en una experiencia divertida y educativa para los estudiantes.

De la misma manera, los bloques son conjuntos de código organizado que se pueden modificar y conectar con el propósito de crear secuencias lógicas que definen acciones (Camalle y Toca, 2023). Por lo tanto, debido a su simplicidad y al elemento lúdico que introduce, la programación por bloques representa una excelente oportunidad para su adopción en los sistemas educativos contemporáneos, con el objetivo es cultivar individuos competentes que aprovechen los múltiples beneficios cognitivos y prácticos que emergen del acto de programar.

Del mismo modo, la programación por bloques, aparte de fomentar la programación, también lo hacen en la motivación. Los autores López y Millán (2022) manifiestan que aplicaciones como Scratch aplicadas en el aula, están relacionadas a la diversión por medio del juego y esto motiva a los estudiantes a utilizarlas para aprender.

En esta línea, López y Millán (2022) destacan que las herramientas de programación por bloques, como Scratch, son valiosas en el aula para el desarrollo del pensamiento lógico y algorítmico, la solución de problemas, el autodiagnóstico, el aprendizaje de conceptos matemáticos, los fundamentos de la programación y el trabajo colaborativo. De manera similar, Agudelo y Salazar (2020) sostienen que fomentar el pensamiento lógico en los estudiantes no solo mejora su autoestima, sino que también afina sus habilidades analíticas, racionales y secuenciales, permitiendo organizar sus ideas, expresar con claridad e interpretar los eventos de su entorno de manera efectiva.

En consecuencia, estos procedimientos que ofrece la programación por bloques ayudarán a iniciarse de una manera sencilla y diferente de aprender, dado que es un método propio para ser usado, sino que también contribuye al desarrollo del pensamiento computacional. Con este fin, se han diseñado diversas herramientas orientadas a este propósito.

Herramientas para programar por bloques

Dentro del ámbito de la programación por bloques, existen diversas herramientas destacadas. Sarmiento (2020) menciona ejemplos como Blocky Games y Minecraft Hora del Código, mientras que Kuz y Ariste (2022) realizan un análisis más detallado que incluye herramientas como Pilas Bloques y Scratch, entre otras. Para obtener más detalles sobre las aplicaciones disponibles para programar en bloques, se puede consultar la Tabla 7.1.

Tabla 7.1. Herramientas basadas en programación por bloques.

Nombre	Descripción	Requisitos
Blocky Games	Es un lenguaje gráfico implementado en JavaScript, contiene una serie de juegos educativos por medio de bloques para enseñar a programar, al final de cada reto puedes obtener el código en texto.	Es compatible con todos los navegadores y no requiere una gran capacidad de componentes. Además, el archivo ejecutable es compatible con el sistema operativo Windows.
Minecraft Hora del Código	Minecraft es un videojuego que consiste construir mundos, excavar y construir, en ella se propone un desafío donde se hace uso de bloques para lograrlo.	Para obtener la versión Hora del Código se puede jugar en línea o descargar la versión que funciona en Windows 7, 8 o 10.
Scratch	La herramienta más famosa en programación por bloques. Este recurso nos permite crear nuestras propias animaciones, juegos, etc. Asimismo, tiene la opción de publicar los proyectos en internet.	Tiene una versión en línea y descargable, navegadores compatibles: Chrome, Edge, Firefox y Safari. Sistemas operativos: Windows 10 versión 1709 o superior MacOS 10.13 o superior.
Pilas Bloques	Es una aplicación desarrollada especialmente para el aula, en la cual se proponen desafíos con el fin de acercar a los estudiantes al mundo de la programación por bloques.	Se puede usar bajo sistemas Windows, Linux y Mac.

Fuente: Adaptado de Kuz y Ariste (2022) y Sarmiento (2020).

Las herramientas de programación por bloques, como Scratch, son perfectas para quienes se inician en la programación. En este caso, se dará a conocer la herramienta denominada Blockly de la empresa de Google, que ofrece una forma interactiva de programar a través de juegos. Esta herramienta no solo permite la manipulación visual de bloques, sino que también posibilita la exportación del código a lenguajes como JavaScript y Python, entre otros, según Calderón (2021). Esta aplicación es funcional tanto para educadores, estudiantes como para desarrolladores.

Programación por bloques utilizando Blockly Games

Blockly Games ofrece varias opciones para aprender a programar, como se muestra en la Tabla 7.2 donde el profesor analizará qué opción es la más conveniente o interesante para aplicar en sus educandos. La misma herramienta nos ofrece información de cómo aplicar estas opciones, la cual debemos dirigirnos a la opción de información de pedagogos. Asimismo, Portaleda et al. (2019), citado por Calderón (2021), resaltan la importancia de Blockly Games en el desarrollo de habilidades de programación en niños pequeños. Aunque se reconoce la intuitividad de su funcionamiento, se destaca la importancia de la orientación docente como elemento clave en el proceso educativo.

Tabla 7.2. Opciones de Blockly Games.

Opción	Definición
Rompecabezas	Es una introducción rápida a las formas de Blockly y cómo encajan las piezas. En esta opción el usuario debe ir armando un puzzle, la cual se selecciona los bloques necesarios a las características de los animales que se muestran, por ejemplo: gato= 4 patas, pelaje, etc.

Opción	Definición
Laberinto	Es una introducción a los bucles y condicional. Aquí el usuario debe ir encaminado a un personaje para que llegue a su objetivo, se debe utilizar bloques como por ejemplo avanzar, girar a la derecha, e izquierda, etc
Pájaro	Es una inmersión profunda en los condicionales. El flujo de control se explora con condiciones cada vez más complejas. Aquí el usuario debe conocer los grados para que el personaje pueda llegar a su objetivo, por ejemplo, el bloque rumbo = 90°.
La música	Es una introducción a las funciones. Utilice funciones para componer música. En esta opción se hace uso de bloques como por ejemplo reproducir nota, por tanto, es una opción interesante para conocer la música.
El tutor del estanque	Presenta la programación basada en texto. Los niveles alternan entre bloques y JavaScript real en un editor de texto.
Estanque.	Es un concurso abierto para programar el pato más inteligente. Utilice bloques o JavaScript.

Fuente: Adaptado a partir de Fraser et al. (2012).

Para este ejemplo práctico, se va a utilizar la opción de la “película”, que va a permitir crear pequeñas animaciones dinámicas. Primero, se establece el personaje que servirá para desarrollar la animación, comenzando por formar las figuras que conformarán: un círculo que representa la cabeza, un rectángulo que simboliza el cuerpo y dos líneas que van a presentar los brazos.

Mientras en la Figura 7.1 se ilustra los bloques para diseñar el personaje, donde se selecciona el bloque denominado “círculo” para la cabeza, ubicado en la opción de película, más el bloque “establecer el color a”, con la finalidad darle color a la figura, ubicado en la opción Color. Posteriormente, se arrastran ambos bloques a la parte derecha de la pantalla y se ensablan como si fuera un rompecabezas, luego se agregan los valores de (x = 50), (y = 70) y un radio de 10.

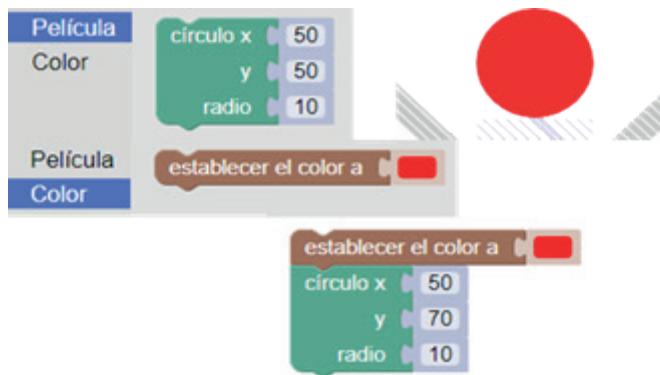


Figura 7.1. Creación de la cabeza para el personaje. Fuente: Adaptado a partir de Fraser et al. (2012).

Para el cuerpo del personaje, se siguen los mismos pasos que con la cabeza: primero se selecciona y arrastra hacia la derecha de la pantalla el bloque “*establecer el color a*”, seguido del bloque “*rectángulo*”, en la opción Película, con los valores de ($x = 50$), ($y = 40$), una anchura de 20 y altura de 40. Posteriormente se arman ambos bloques. En la Figura 7.2 se observa detalladamente.

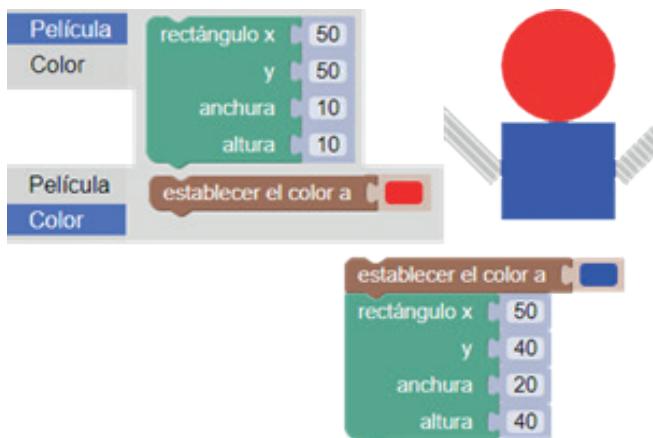


Figura 7.2. Creación del cuerpo del personaje. Fuente: Adaptado a partir de Fraser et al. (2019).

En lo que respecta a los brazos, se seleccionan dos bloques denominados “línea” para cada brazo, en la opción de película. Al igual que los casos anteriores, se hace uso del bloque “*establecer el color a*” colocándole color negro. En los bloques “línea” se coloca los valores en el brazo derecho de inicio ($x = 40$), inicio ($y = 50$), final ($x = 20$), final ($y = 70$) y anchura = 5. En cambio, en el brazo izquierdo se colocan los valores de inicio ($x = 60$), inicio ($y = 50$), final ($x = 80$), final ($y = 70$) y anchura = 5. En la Figura 7.3 se exponen detalladamente.

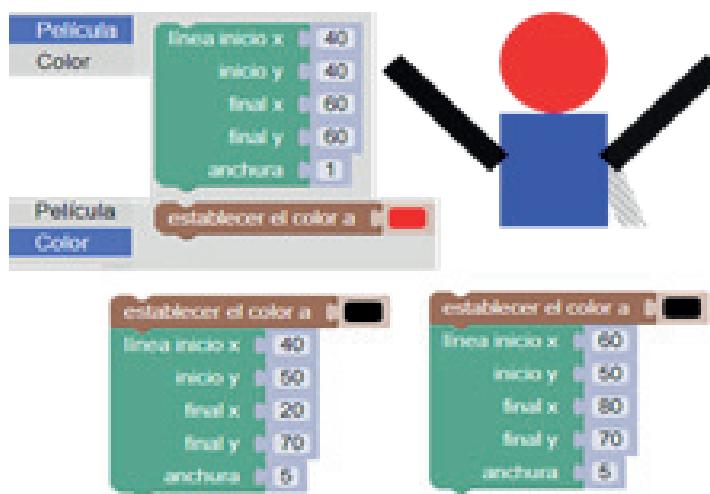


Figura 7.3. Creación de los brazos del personaje. Fuente: Adaptado a partir de Fraser et al. (2012).

Una vez creado el personaje, el siguiente paso es una animación. En la Figura 7.4 se presencian los bloques utilizados, donde se moverá su brazo izquierdo por la pantalla, y para ello se hace uso del “bloque de color rojo” [time (0-->100)] que incrustará en el “bloque del brazo izquierdo” en el valor (final y). Esto posibilita que, al reproducir la animación, el brazo izquierdo comience en posición final ($y = 0$) y termine en la posición final ($y = 100$), creando así el efecto de movimiento del brazo.

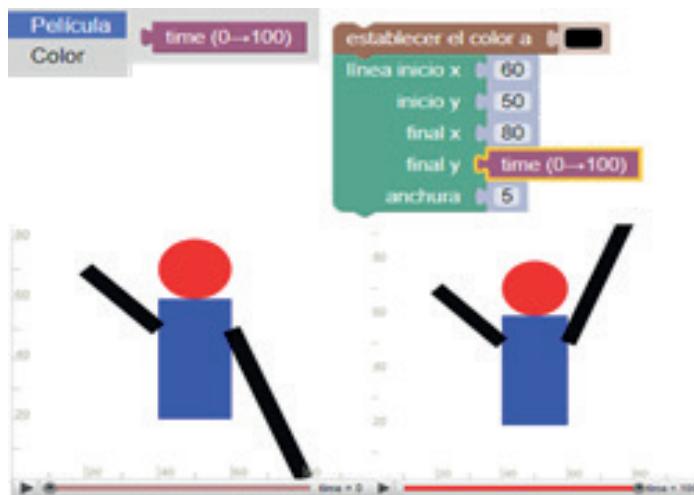


Figura 7.4. Animación de nuestro personaje y los bloques que se utilizaron.

Fuente: Adaptado a partir de Fraser et al. (2012).

Para aquellos interesados en conocer las líneas de código en JavaScript, estas se mostrarán con el mensaje de ¡Felicitaciones! una vez completada la misión elegida. En la Figura 7.5 es posible apreciar la compilación utilizada para la animación creada, representada por 8 líneas de código.

¡Felicitaciones!

Resolviste este nivel con 8 líneas de JavaScript:

```
penColour('#ff0000');
circle(50, 70, 10);

penColour('#3333ff');
rect(50, 40, 20, 40);

penColour('#000000');
line(40, 50, 20, 70, 5);

penColour('#000000');
line(60, 50, 80, time(), 5);
```

Figura 7.5. Código de JavaScript utilizado en la animación. Fuente: Adaptado a partir de Fraser et al. (2012).

De esta manera, se entiende que Blockly Games y otras aplicaciones ofrecen una introducción atractiva y accesible a la programación a través de juegos interactivos. Al manipular bloques que simbolizan estructuras de código, los estudiantes pueden ensamblar programas operativos, facilitando el desarrollo de habilidades clave en el pensamiento computacional, como la lógica secuencial, la resolución de problemas y el análisis crítico.

En consecuencia, se obtienen muchos beneficios, tal como detallan organismos como la Sociedad Internacional para la Tecnología en Educación (ISTE) y la Asociación de Docentes en Ciencias de la Computación (CSTA), citados por Ahumada et al. (2019), quienes indican las características que se desarrollarán con el pensamiento computacional, entre las que se incluyen:

- La capacidad de identificar problemas que pueden ser resueltos eficazmente con la ayuda de una computadora u otras herramientas digitales.
- Competencia para que organice y analice los datos correctamente.
- Habilidad para representar datos usando abstracciones como modelos y simulaciones.
- La destreza en la automatización de soluciones a través del pensamiento algorítmico.
- Aptitud para identificar, analizar e implementar posibles soluciones viables, buscando la combinación más eficiente para la resolución de problemas.
- Capacidad de generalizar el proceso de resolución de problemas y traducirlo en varias soluciones.

Por otra parte, Calderón (2021) en su trabajo de investigación recomienda usar Blockly Games con la intención de mejorar las habilidades de razonamiento en los estudiantes y destaca la importancia de contar

con un entorno propicio para el aprendizaje presencial, facilitando la participación y comprensión de acuerdo a las edades del estudiante; También recalca el valor de combinar el juego con el aprendizaje, lo cual resulta en una metodología óptima para la adquisición de nuevos conocimientos.

Reflexiones

En el siglo XXI, la habilidad de programar se ha vuelto fundamental debido al creciente papel que desempeña la tecnología en todos los aspectos de nuestra vida cotidiana. La programación en este siglo no solo es una habilidad útil, sino también esencial para prosperar en una sociedad cada vez más digitalizada y contribuir al progreso tecnológico y social.

Por otra parte, la programación en la educación es una herramienta poderosa para equipar a los estudiantes con las habilidades necesarias que facilita prosperar, fomentando la innovación y el progreso en nuestra sociedad. Su integración efectiva en los sistemas educativos puede tener un impacto positivo duradero en el futuro de la educación y el desarrollo humano.

Asimismo, la programación por bloques es una herramienta importante para enseñar codificación de una manera accesible, creativa y estructurada. Esta metodología invita a los estudiantes a ser protagonistas de su educación, alentándolos a desarrollar habilidades de resolución de problemas y a crear contenido tecnológico de manera innovadora. Agregando a lo anterior, el uso productivo de herramientas tecnológicas se ve potenciado por esta práctica. Por consiguiente, es fundamental que los educadores reciban la formación adecuada para guiar eficientemente este proceso de enseñanza-aprendizaje, asegurando así que los estudiantes puedan aprovechar al máximo las ventajas de la programación por bloques.

Finalmente, la herramienta Blockly Games permitirá aprender a programar a través de juegos interactivos basados en bloques, lo cual estimula la motivación y el interés en los estudiantes. Esta herramienta no solo diversifica los métodos de aprendizaje, sino que también introduce a los estudiantes al lenguaje de programación JavaScript, preparándolos para abordar conceptos más avanzados en el futuro, y por este medio, desarrollar el pensamiento computacional.

Bibliografía

- Agudelo, C. y Salazar, C. (2020). Gamificación Como Estrategia Para Posibilitar el Aprendizaje de la Programación en los Estudiantes de Educacion Media. Universidad de Santander. <https://repositorio.udes.edu.co/entities/publication/058d397b-ee2c-47af-bc9c-d2465d305c44>
- Ahumada, H., Rivas, D., Contreras, N. y Poliche, M. (2019). Pensamiento Computacional mediante Programación por Bloques. ReTyCa (Revista de Tecnología y Ciencias Aplicadas), 9-16. <https://retyca-tecnico.unca.edu.ar/wp-content/uploads/2022/12/4-1-2.pdf>
- Angamarca, O. y Andrade, D. (2022). Enseñanza de programación a niños de edad escolar utilizando Scratch para mejora del razonamiento lógico. Pro Sciences: Revista de Producción, Ciencias e Investigación, 6(42), 111-121. <https://journalprosciences.com/index.php/ps/article/view/475>
- Astudillo, G., Bast, S., Segovia, D. y Castro, L. (2019). Revisión de propuestas para la enseñanza de la programación. In XIV Congreso Nacional de Tecnología en Educación y Educación en Tecnología (TE&ET 2019), (Universidad Nacional de San Luis, 1 y 2 de julio de 2019). <https://sedici.unlp.edu.ar/handle/10915/90811>
- Calderón, R. (2021). Blockly games y su influencia en el desarrollo del pensamiento computacional, para los estudiantes de quinto año de la UE “Carlos maría de la Condamine (Master’s thesis, Universidad Nocial de Chimborazo). <http://dspace.unach.edu.ec/handle/51000/8310>

Camalle, T. y Toca, J. (2023). Aplicación web para el aprendizaje de programación en bloques para niños de 8-12 años, de la Provincia de Cotopaxi mediante la aplicación de prácticas ágiles (Bachelor's thesis, Ecuador: Latacunga: Universidad Técnica de Cotopaxi (UTC)). <http://repositorio.utc.edu.ec/handle/27000/10422>

Fraser, N., Mark Friedman., Neutron, Q. y Spertus E. (2012). Blockly (2019). <https://developers.google.com/blockly?hl=es-419>

García, A., Gutiérrez, P. y Ayuso del Puerto, D. (2022) «PROPUUESTA DIDÁCTICA DE INICIACIÓN A LA PROGRAMACIÓN EN EDUCACIÓN INFANTIL CONSIDERANDO EL DUA», Revista Infancia, Educación y Aprendizaje, 8(2), pp. 98–115. <https://panambi.uv.cl/index.php/IEYA/article/view/2897>

González, C. (2019). Estado del arte en la enseñanza del pensamiento computacional y la programación en la etapa infantil. https://pdfs.semanticscholar.org/3507/72152454eb755098bf405803964eb4c2cb21.pdf?_ga=2.205826902.414772606.1670358358-2058443123.1670358358

Gutierrez, Á. (2022). Los entornos virtuales de aprendizaje como espacios en la sociedad del conocimiento. Las tecnologías de la información y la comunicación en el contexto educativo. (pp. 77-99). Corporación Universitaria Minuto de Dios –UNIMINUTO. <https://doi.org/10.26620/uniminuto/978-958-763-565-2.cap.4>

Kuz, A. y Ariste, M. C. (2022). Análisis y revisión de softwares educativos para el aprendizaje de la programación en entornos lúdicos. Tecné, Episteme y Didaxis: TED, (52), 117-136. http://www.scielo.org.co/scielo.php?pid=S0121-38142022000200117&script=sci_arttext

López, A. y Millán, D. (2022). Evaluación de Competencias de Programación en Scratch Como Propuesta Digital de Aprendizaje. Universidad de Santander. <https://repositorio.udes.edu.co/entities/publication/48e50a6a-ab66-4241-b37d-a85baff723e3>

Quiroz, D., Carmona, J., Castrillón, A. y Villa, J. A. (2021). Integración del Pensamiento Computacional en la educación primaria y secundaria en Latinoamérica: una revisión sistemática de literatura. Revista de Educación a Distancia (RED), 21(68). <https://doi.org/10.6018/red.485321>

Sarmiento, M. (2020). Lenguajes y entornos de programación para fortalecer el desarrollo de competencias concernientes al pensamiento computacional. Hamut'ay, 7 (3), 86-97 <http://dx.doi.org/10.21503/hamu.v7i3.2200>

Sánchez, Y., Díaz, K. y Muñoz, M. (2020). La enseñanza del Scratch en la formación inicial de profesores de Informática. Serie Científica de la Universidad de las Ciencias Informáticas, 13(10), 93-102. <https://dialnet.unirioja.es/servlet/articulo?codigo=8590343>

Sánchez, M. (2021). Aprender programando y programar para aprender con Scratch Jr (Bachelor's thesis). <https://repositorio.21.edu.ar/handle/ues21/21328>

Capítulo 8

Tynker y el mundo de la programación basada en bloques en la educación primaria

Michay Caraguay Gloria Cecibel, Quezada Vera Lida Andreina, Saritama Cango José Pablo, Sisalima Angamarca Darly Germania

Universidad Nacional de Loja

cecibel.michay@unl.edu.ec, laquezadav@unl.edu.ec, jose.saritama@unl.edu.ec, darly.sisalima@unl.ed.ec

Introducción

La programación en la actualidad ha adquirido un papel importante en la vida educativa, ya que la sociedad se encuentra en constante transformación tecnológica, en consecuencia, crea la necesidad de aprender y desarrollar habilidades relacionadas a los algoritmos, pensamiento computacional y resolución de problemas. Desde una perspectiva general, la programación puede convertirse en un temado retador para comprender conceptos abstractos relacionados a la codificación, sin embargo, en los últimos años existen plataformas que permiten la introducción a este aprendizaje desde un sentido práctico e intuitivo.

En perspectiva, la programación abordada desde edades tempranas requiere de herramientas o recursos diseñados y adecuados específicamente a los interés y proceso cognitivo de esta población. Por consiguiente, se plantea en el presente capítulo un análisis sobre la programación en bloques enfatizando su utilización en la educación primaria, asimismo, se aborda a Tynker, una plataforma diseñada para la enseñanza-aprendizaje de la programación.

Educación primaria en el contexto ecuatoriano

La educación primaria en Ecuador se caracteriza por un enfoque en el desarrollo integral del niño, incluyendo el desarrollo de habilidades cognitivas, sociales, emocionales y físicas. En este contexto, las habilidades socioemocionales adquieren una gran importancia para el aprendizaje. Por ende, Vivanco (2022) manifiesta que “se debe dar prioridad en el aprendizaje contenidos relacionados con la temática desde los niveles iniciales, con el fin de que se logre estimular el compartir con otras personas, el intercambio de aprendizajes”. Por ello, la inclusión de programación basada en bloques se ve como una oportunidad para fortalecer el aprendizaje de los estudiantes y prepararlos para un futuro digital.

En el contexto de la educación primaria en Ecuador, la introducción de la programación representa un avance significativo hacia la preparación de los estudiantes para los desafíos del siglo XXI. Como señala Cárdenas (2019), el mundo está experimentando cambios a un ritmo cada vez más rápido, lo que implica una necesidad urgente de adaptación en el ámbito educativo. Es por ello que es necesario promover desde la educación el desarrollo de la creatividad y la capacidad de encontrar soluciones innovadoras ante los nuevos retos que se presentan.

La introducción de la programación en las aulas contribuye a cultivar habilidades cognitivas fundamentales como el pensamiento lógico, la resolución de problemas y la creatividad desde una edad temprana. No obstante, más allá de formar a futuros profesionales en tecnología, esta iniciativa busca fomentar una cultura de innovación y adaptabilidad en todas las esferas de la vida (Cárdenas, 2019). En este sentido, la programación en la educación primaria no solo es una herramienta pedagógica, sino también un medio para impulsar el desarrollo integral de los estudiantes y prepararlos para un mundo en constante evolución.

Programación basada en bloques

La programación es un conjunto instrucciones de computadora, donde se parte creando el algoritmo concebido como un conjunto finito de pasos a partir del cual se procede a codificarlo y llevarlo a un determinado lenguaje de programación, de modo que pueda ser ejecutado por una computadora. Así mismo Cáceres (2019), manifiesta que la programación consiste en traducir una serie de pasos, conocidos como algoritmo, a un lenguaje específico que las computadoras puedan entender, es decir, este proceso se resume en la traducción de un algoritmo a un lenguaje comprensible por las computadoras.

En el contexto actual, el mercado ofrece una diversidad de lenguajes de programación y tecnologías informáticas, como señala Cáceres (2019), quien define los lenguajes de programación como secuencias de comandos que, respetando la sintaxis, permiten resolver problemas de información. En los últimos años, se ha prestado una atención particular al desafío de la transición de la programación basada en bloques hacia la programación tradicional mediante lenguajes de texto. Este cambio se ha convertido en un área significativa de investigación y desarrollo en el campo de la Educación en Ciencias de la Computación, especialmente relevante en la integración de herramientas como Tynker en la educación primaria.

De esta manera, se han desarrollado diversos entornos de programación, cada uno con características y estrategias de enseñanza particulares, según varios estudios se plantea que tanto los entornos basados en bloques como en texto tienen beneficios y límites; además comunican que existe una brecha importante entre los dos ambientes, que dificulta los procesos de transición. Destacando a la programación por bloques, Prado et al. (2023), manifiestan que ésta segmenta las actividades que se asemejan o complementan, permitiendo a los usuarios arrastrar y

soltar estructuras, procesos o subrutinas (como condicionales, bucles) junto con variables y operadores (aritméticos, de cadena y lógicos) en lugar de escribir líneas de código manualmente.

Cómo argumento adicional, el crecimiento de los entornos de enseñanza y aprendizaje en línea, ha permitido la creación de múltiples herramientas para enseñar programación y enriquecer el proceso de aprendizaje, haciéndolo más interactivo, creativo e interesante; al respecto Venigalla y Chimalakonda (citados en Prado et al. 2023), coinciden en que para lograrlo, estas herramientas se valen de diversas tecnologías, tales como la realidad mixta, la inteligencia artificial, entre otras.

Sin embargo, es necesario abordar las competencias necesarias para desenvolverse con éxito en el mundo digital, haciendo hincapié en lo que respecta a la programación como herramienta para la resolución de problemas, donde Llorens (2015) opina que, para equipar a los discentes con las habilidades necesarias para enfrentar la actualidad, es crucial integrar al que podríamos denominar “lenguaje digital”, resaltando que la programación basada en texto se ha establecido como el estándar actual en la mayoría de los lenguajes y se espera que sea parte integral del aprendizaje en el contexto de la educación secundaria.

Tynker

Las herramientas que disponen de programación en bloques han facilitado la comprensión y construcción de código desarrollando así habilidades lógicas algorítmicas. En este sentido, Tynker (2013a) es una plataforma de aprendizaje con enfoque a la codificación destinadas para niños equipado por desarrolladores, educadores, hackers y padres para brindar los primeros pasos a la programación.

Cabe considerar que este software se destaca por su enfoque gradual en el aprendizaje de la programación. Por ende, Fernández (2021), concuerda que Tynker “intenta que los niños piensen directamente como programadores, los más pequeños aprenden lenguajes de programación sin dejar de divertirse”; permitiendo a los discentes progresar desde un lenguaje de bloques intuitivo hasta lenguajes de texto como JavaScript o Python.

En función de lo planteado, la programación implica un amplio abanico para desarrollar diversas habilidades, de acuerdo con Kuz y Ariste (2021), la programación no se limita a la creación de un código, más bien desarrolla habilidades que involucran el pensamiento computacional, la autonomía, la creatividad. Por ende, en la Tabla 8.1 se describe las habilidades que promueve la programación en los niños.

Tabla 8.1. Habilidades que desarrolla la programación

Habilidad	Descripción
Creatividad	Los niños ejercitan y construyen sus ideas mientras aprenden a codificar sus proyectos.
Confianza	La combinación entre el conocimiento técnico y la creatividad produce espacios de confianza para el aprendizaje.
Comunicación	Al utilizar un comunicación lógica y organizada en la computadora mejora esta habilidad interpersonal con las personas.
Pensamiento crítico	La programación comprende relaciones de causa y efecto que se utiliza para tomar decisiones estratégicas.
Empoderamiento	Los proyectos de codificar permiten a los niños ser perceptivos a su entorno circundante.
Enfoque y organización	La concentración y organización son efectos que produce en los niños cuando aprende a codificar.
Resiliencia	La codificación es una tarea frustrante por ello resulta retador mantener la paciencia y motivación para finalizar un proyecto.

Habilidades matemáticas	Los conceptos abstractos resultan divertidos al aprender a codificar los problemas matemáticos como lógica y organización.
Habilidades de escritura	Los proyectos creativos basados en historias permiten a los niños a planificar y ser concisos en su escritura.

Fuente: Adaptado a partir de Tynker (2013b)

En este sentido, se comprende que Tynker es una plataforma orientada al desarrollo de habilidades necesarias para el auge tecnológico, donde Gajardo y Webber (2021) enfatizan que una plataforma diseñada para enseñar programación debe integrar una serie de recursos que involucre resolver problemas de manera intuitiva incrementándose el nivel de complejidad gradualmente para responder al proceso de aprendizaje. En consecuencia, como se observa en la Figura 8.1, Tynker facilita a los niños diversas herramientas y recursos para la programación.



Figura 8.1. Herramientas que incluye Tynker. Fuente: Tynker (2013c).

De este modo, Tynker interviene en el aprendizaje utilizando varios recursos afines a una perspectiva del pensamiento computacional, Arrieta (2022) determina que esta plataforma es factible como propuesta de gamificación en el aula ya que considera que responde a las necesidades continuas del aprendizaje de los estudiantes en el área de la informática. Entonces, desde la perspectiva lúdica permite la creación de programas sencillos y juegos interactivos relacionados con los contenidos que se requiera desarrollar.

Tynker en la educación primaria

La educación primaria involucra a jóvenes estudiantes, quienes al estar inmersos en esta era digital requieren desarrollar habilidades relacionadas al ámbito tecnológico, por tanto, Elsawah, y Thabet (2022) en su publicación, aluden que Tynker es una plataforma pertinente para enseñar programación a estudiantes principiantes, enfocada desde la gamificación con la finalidad de aprender ideas de programación complejas con enfoque atractivo y adecuado según las necesidades, edades e intereses de los estudiantes. Asimismo, Tynker presenta una gran variedad de plantillas presentando una amplia opción de crear nuevos escenarios.

Se debe señalar que esta plataforma, usable en línea o instalada en el dispositivo, según Cerón (2022), es parte de la programación conectada; constituyéndose además en un software fundamental para la enseñanza y aprendizaje del lenguaje visual de programación en niños. Así por ejemplo, en la Figura 8.2 se observa el entorno que Tynker, en uno de sus juegos destinados a comprender la programación en función al tema del Sistema Solar.

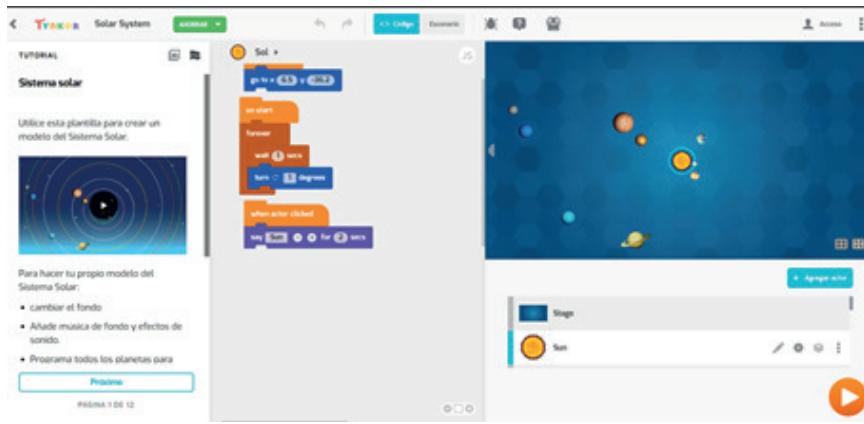


Figura 8.2. Entorno de Tynker. Fuente: Tynker (2013d).

Tal y como se describe en la Figura 8.2, Tynker incluye en la parte superior icónicos relacionados al nombre del proyecto, opciones de guardar, flechas de deshacer o rehacer los cambios, código, escenario, depurador, ayuda, mochila y acceso para iniciar sesión; además, en la parte izquierda de la pantalla brinda Tutorial donde explica el proceso para crear el proyecto; en esta misma zona, se encuentra el icono de puzzle, el cual despliega los diferentes bloques de programación; en el centro de la pantalla se ubica el espacio para la programación incluye también el icono JS, al hacer clic se despliega una pantalla que permite programar en este lenguaje de texto, JavaScript (JS); en la parte derecha de la pantalla corresponde al escenario que se está desarrollando, específicamente en la imagen se observa el diseño del Sistema Solar.

En evidencia, según los elementos mencionados con anterioridad, Tynker posibilita a estudiantes de primaria de entre 5 y 10 años a desarrollar habilidades de programación y pensamiento computacional guiados a través de cursos y juegos interactivos donde posibilita la creación de juegos, historias, aplicaciones, controlar drones y robots

entre otros (Katyetova, 2022). Si bien es cierto, esta plataforma anima a los estudiantes a trabajar de manera autónoma aprovechando todas las posibilidades de herramientas y recursos.

Asimismo, Tynker al facilitar a los estudiantes plasmar sus ideas a través de la programación De Miguel (2024), indican que la plataforma Tynker es una “plataforma de programación educativa dirigida al alumnado de entre 5 y 18 años que aborda esta disciplina desde el juego y los retos”, además cuenta con un manual de comandos ya plasmados en la herramienta que permite hacer el trabajo más interactivo. Adicionalmente, permite al discente reconocer patrones y enfrentarse a un problema mayor descomponiéndose en otros más pequeños de fácil resolución, que ayuda a los pequeños a razonar de forma lógica.

Reflexión

La programación en la actualidad ha tomado una fuerte influencia, ya que muchas aplicaciones se basan en programaciones, bucles, puzzles, entre otros. En el presente capítulo se aborda sobre Tynker una plataforma útil a la hora de crear una clase lúdica y didáctica en búsqueda del pensamiento computacional podemos, se puede encontrar una gran variedad de recursos que abordan diferentes áreas de conocimiento.

Entonces, se puede considerar como una herramienta práctica para utilizarla en la educación primaria, que comprende niños de temprana edad que dan sus primeros pasos a la programación, a razón que, es una plataforma con una interfaz intuitiva donde los límites están en la imaginación, cabe resaltar que la evolución tecnológica exige a la sociedad preparar ciudadanos con habilidades digitales bajo una apegada ética, siendo indispensable potenciar habilidades blandas, las mismas que están presentes como misión en Tynker.

Bibliografía

- Arrieta, G. (2022). Recurso educativo para la programación en el área de informática con la herramienta Tynker, desde el enfoque de gamificación [Tesis de maestría, Pontificia Universidad Católica del Ecuador]. Repositorio institucional. <https://repositorio.puce.edu.ec/server/api/core/bitstreams/5eb9209e-d15e-4d61-8f9e-e2d5a8f545fd/content>
- Cáceres, L. (2019). Introducción a la programación Introducción, fundamentos de la programación, herramientas de programación, metodología de programación, aplicaciones [Universidad Nacional de Educación Enrique Guzmán y Valle, Perú]. Repositorio institucional. <https://repositorioslatinoamericanos.uchile.cl/handle/2250/6540069>
- Cárdenas, L. (2019). La creatividad y la educación en el siglo XXI. Revista Interamericana de Investigación, Educación y Pedagogía, vol. 12, núm. 2, pp. 211-224, 2019. <https://www.redalyc.org/journal/5610/561068684008/html/>
- Cerón, J. (2022). Lenguaje de programación para niños y niñas: perspectivas conectadas y desconectadas en la educación básica. Revista Internacional de Pedagogía e Innovación Educativa 3(1), 45-66. <https://doi.org/10.51660/ripie.v3i1.108>
- De Miguel, R. (2024). Así es Tynker, la plataforma que enseña programación a través del juego. EDUCACIÓN 3.0. <https://www.educaciontrespuntocero.com/tecnologia/tynker-programacion/>
- Elsawah, W. y Thabet, R. (2022). The Effectiveness of Tynker Platform in Helping Early Ages Students to Acquire the Coding Skills

Necessary for 21st Century. En AI-Emran, AI-Sharafi y Shaalan(Ed.), International Conference on Information Systems and Intelligent Applications (381-397). Springer International Publishing.

Fernández, P. (2021). TYNKER: LA PLATAFORMA PARA QUE LOS NIÑOS APRENDAN A PROGRAMAR. IslaVisual. https://www.islavisual.com/articulos/desarrollo_web/tynker-plataforma-para-que-los-ninos-aprendan-a-programar#:~:text=A%20diferencia%20de%20otros%20tipos,programaci%C3%B3n%20sin%20dejar%20de%20divertirse.

Gajardo, A. y Webber, C. (2021). Plataformas de programación para niños: una revisión sistemática de la literatura. Revista Educacional Interdisciplinaria, 10(1), 166-179. <https://seer.faccat.br/index.php/redin/article/view/2254>

Katyetova, A. (2022). How Competitions Can Motivate Children to Learn Programming. Olympiads in Informatics, 16, 13-22. https://www.researchgate.net/profile/Aliya-Katyetova/publication/361780813_HowCompetitions_Can_Motivate_Children_to_Learn_Programming/links/62cd51253bbe636e0c591445/How-Competitions-Can-Motivate-Children-to-Learn-Programming.pdf?_sg%5B0%5D=started_experiment_milestone&origin=journalDetail&_rtd=e30%3D

Kuz, A. y Ariste, M. (2021). Un análisis desde la programación estructurada del lenguaje Scratch como entorno lúdico educativo. Revista Iberoamericana de Informática Educativa, (33), 14-21. <https://dialnet.unirioja.es/servlet/articulo?codigo=7985875>

Llorens, F. (2015). Dicen por ahí. Revista de Investigación En Docencia Universitaria de La Informática, 8(2), 11–14. <http://>

runningonempty.acm.org

- Prado, M., Paucar, R., Valarezo, J., Acosta, M. y Guaicha, K. (2023). Beneficios de la programación por bloques utilizando Sphero mini mediante aprendizaje móvil en la educación superior. E-Ciencias De La Información, 13(2). <https://doi.org/10.15517/eci.v13i2.54814>
- Tynker. (2013a). Sobre nosotros. <https://www.tynker.com/about/>
- Tynker. (2013b). Cómo la codificación desarrolla las habilidades del siglo XXI. <https://www.tynker.com/content/how-coding-develops-21st-century-skills>
- Tynker. (2013c). Aprobado por educadores. Amado por los niños. <https://www.tynker.com/parents/curriculum/>
- Tynker. (2013d). Solar System. <https://www.tynker.com/ide/v3?type=diy&p=54f4d2b284aafaae36000012>
- Vivanco, N. (2022). Las habilidades socioemocionales en el desarrollo cognitivo de los niños y niñas del Nivel Inicial 2 de la Unidad Educativa Fiscal Nueva Aurora [Tesis de maestría, Universidad Central del Ecuador, Quito, Ecuador]. Repositorio institucional. <https://www.dspace.uce.edu.ec/server/api/core/bitstreams/0a14e031-a140-4a4b-8bab-f3642b072faf/content>

Capítulo 9

Educación en inteligencia artificial y codificación en PictoBlox

Coloma Andrade María de los Angeles, Buri Orosco Elian Andres, Silva Suquilanda Gabriel Alejandro.

Universidad Nacional de Loja.

elian.buri@unl.edu.ec, gabriel.silva@unl.edu.ec, maria.coloma@unl.edu.ec

Introducción

La Inteligencia Artificial (IA) se ha convertido en un pilar fundamental dentro de la informática, desempeñando un rol crucial en la sociedad del conocimiento. Su integración en el sector educativo es esencial, dada su capacidad de adaptación y su valioso aporte a diversas áreas del saber. Facilita la interacción entre docentes y estudiantes en entornos virtuales enriquecidos por la implementación de nuevas tecnologías. En este contexto, el presente artículo se centrará en la herramienta de PictoBlox que emerge como una herramienta de programación gráfica innovadora, diseñada para inculcar en los estudiantes habilidades de manejo, programación y diseño de actividades interactivas basadas en la IA, abriendo así las puertas a un aprendizaje interactivo y contemporáneo.

En el ámbito educativo, la IA tiene como objetivo diversificar los conocimientos a través del desarrollo de programas que crean entornos de aprendizaje adaptativos y personalizados. Su propósito es desarrollar estrategias puntuales para la adquisición de conocimientos, beneficiando tanto a estudiantes como a los docentes al proporcionar métodos activos y herramientas avanzadas para la enseñanza efectiva y puntual. Entender

la IA como una herramienta pedagógica implica reconocerla como clave en la educación temprana de los niños; por ende, es indispensable partir definiendo las metodologías, estrategias y técnicas de enseñanza que se adapten a los entornos educativos actuales.

Pensamiento computacional

El pensamiento computacional implica resolver problemas, diseñar sistemas y comprender el comportamiento humano, basándose en los conceptos fundamentales de la informática, como los algoritmos, la generalización, abstracción, descomposición y evaluación (Wing, 2006). Por tanto, es esencial que los docentes promuevan actitudes y destrezas orientadas al pensamiento computacional, las mismas que permitirán innovar y crear ambientes de aprendizaje activos. Es fundamental que fomenten actividades con IA con el propósito de formar en los estudiantes un pensamiento lógico, capacidad de pensar algorítmicamente, habilidad para la descomposición, identificación de patrones y la capacidad de identificar mediante la abstracción.

Dentro de este marco, es conveniente mencionar a Bordignon e Iglesias (2020), quienes describen estas habilidades de la siguiente manera: primero, la capacidad de pensar algorítmicamente implica la habilidad de cada individuo para llevar a cabo el proceso de abstracción, modelado, deducción lógica y síntesis de la solución que conducirá a la escritura del algoritmo correcto. Promover esta capacidad en el estudiante permite reforzar habilidades de ordenamiento, que se complementan con la capacidad de descomposición, que busca dividir grandes problemas en porciones manejables.

Por último, la capacidad de identificar patrones es una parte principal de este proceso, ya que posibilita detectar similitudes entre los diferentes caminos a la resolución de problemas. Esto simplifica el proceso de

solución al facilitar la creación de generalizaciones que agilizan y optimizan el proceso; por otro lado, la abstracción es identificar los aspectos más relevantes de un problema y ocultar algunos detalles específicos que no son necesarios centrarnos.

Gamificación

En la educación contemporánea, la motivación emerge como un factor crucial en el proceso de enseñanza-aprendizaje de los estudiantes. En este contexto, la aplicación de técnicas, como la gamificación, cobra una relevancia significativa, dado que consiste en aplicar elementos lúdicos y estructuras de juego en la educación, promoviendo así el pensamiento crítico, la solución innovadora de problemas y la colaboración efectiva (López y Gutiérrez 2017). Esto permite innovar el proceso de aprendizaje en los niños al integrar juegos interactivos en el currículo, estimulando la creatividad y el gusto por el conocimiento.

En la enseñanza, esta estrategia se centra en proporcionar una visión lúdica y atractiva a los entornos educativos. Su objetivo es desarrollar una serie de habilidades identificadas por García et al. (2018), las cuales incluyen despertar la curiosidad por aprender, fomentar el descubrimiento y el conocimiento, fortalecer la autoestima y la motivación, estimular la creatividad y facilitar la adquisición de contenidos a través de la experiencia, competencias y habilidades.

Metodología STEAM

La evolución de los sistemas de educación a distancia abrió la posibilidad de la adopción de modelos educativos alternativos, brindando una oportunidad única para la exploración y el fortalecimiento de competencias diversas y esenciales para el siglo XXI. En consecuencia, García et al. (2017) describen el modelo STEAM se fundamenta en la

interdisciplinariedad de la Ciencia, Tecnología, Ingeniería y Matemática, su objetivo es motivar a niños y jóvenes a desarrollarse en asignaturas relacionadas a la ciencia y la investigación

De acuerdo con Ferrada et al. (2020) considera esta metodología como un proceso educativo enmarcado en la relación de las ciencias, tecnología y cultura, como punto de partida para guiar al estudiante al mundo de la investigación, diálogo y pensamiento computacional, dicha metodología nos permitirá reforzar la participación e innovación en los estudiantes.

Programación por bloques

La programación se ha convertido en una competencia fundamental en nuestra era digital, actuando como vínculo entre la creatividad humana y las capacidades de la tecnología. Según ProFuturo (2022), aprender a programar es necesario para el desarrollo de habilidades lógicas y de resolución de problemas, además su implementación en el contexto educativo permite el progreso de las nuevas tendencias tecnológicas se han avanzado en plataformas formativas que vinculan estas destrezas.

En este sentido se comprende que, la programación basada en bloques es un pilar fundamental en la enseñanza de la IA, proporcionando a los niños una introducción amigable al mundo de la programación, donde tienen la oportunidad de arrastrar y soltar fichas para crear soluciones a problemas, mediante el desarrollo de programas sencillos que los adentren a mundos con historias interactivas y simulaciones llenas de fantasía y creatividad (Maloney et al., 2008). De esta manera, la programación en bloques elimina los errores de sintaxis comunes en otros lenguajes, permitiendo a los estudiantes concentrarse en la lógica y el diseño del programa

Por consiguiente, esta forma de aprender a programar es cada vez más utilizada a quienes inician en el mundo de la codificación (Ahumada et al., 2018). Este tipo de programación capacita a los niños para crear animación, juego o representaciones de situaciones cotidianas, de manera sencilla, libre y organizada; además, se integran fácilmente con metodologías y estrategias activas que fomentan un pensamiento y razonamiento computacional.

PictoBlox

Para la aplicación de la inteligencia artificial con la programación basada en bloques se empleó PictoBlox un software gratuito basado en la tecnología de Scratch 3.0, que mantiene sus propias características y añade extensiones de IA con funciones de reconocimiento facial, voz y texto (Padilla, 2019). Esta herramienta permite a los niños desarrollar proyectos y actividades interactivas de forma dinámica y participativa.

Además, PictoBlox se adecua como un software educativo, ideal para que los niños den sus primeros pasos en el mundo de la programación; Agilo (2020) indica que su interfaz es fácil de usar, y la funcionalidad de arrastrar y soltar bloques eliminan la necesidad de memorizar la sintaxis y las reglas, algo que puede resultar intimidante en los lenguajes de programación tradicionales. Esto favorece que estos se enfoquen en el problema en cuestión, desarrollando habilidades como el razonamiento lógico, esenciales en un mundo impulsado por las tecnologías.

No hay limitaciones para su uso, porque PictoBlox es un software multiplataforma compatible con sistemas operativos como Windows, MacOS, iOS, Linux y Android. Dispone tanto de una versión para escritorio y la versión beta para la web. La interfaz gráfica de este software educativo se presenta en la Tabla 9.1, que se muestra a continuación.

Tabla 9.1. Herramientas de Pictoblox.

Herramientas	Descripción
Sprite	Un sprite es un objeto, o un personaje, que realiza diferentes acciones en los proyectos, entiende y obedece las instrucciones que se le dan, un sprite tiene disfraces (apariencias).
Stage (Escenario)	El escenario es el área donde el sprite realiza acciones de acuerdo con su programación, dentro de PictoBlox existe una biblioteca de imágenes de fondo para decorar esta sección.
Script	Un script es el lenguaje de programación de PictoBlox / Scratch, consiste en un conjunto de ‘bloques’ que se ordenan uno debajo del otro en un orden específico para realizar una tarea o una serie de actividades.
Bloques	Un bloque es como una pieza de rompecabezas que se utiliza para escribir programas simplemente arrastrándolos y soltándolos uno debajo del otro en el área de secuencias de comandos.
Paleta de bloques	Consiste en diferentes paletas como movimiento, apariencia, sonido, eventos, control, sensores, operadores y variables.
Extensiones	PictoBlox posee extensiones para la aplicación de la inteligencia artificial como reconocimiento facial, texto y voz, para la práctica de machine Learning.

Fuente: adaptado a partir de PictoBlox (2019).

Comprender la interfaz gráfica de esta aplicación es esencial, considerando que la configuración y el diseño de los elementos visuales son clave para propiciar un diálogo efectivo entre el creador, el contenido y el usuario, tal como sugiere Cordero (2018). Esta interacción es vital porque propicia que el usuario se involucre de manera intuitiva y significativa con la aplicación. En este sentido, la Figura 9.1 no sólo detalla las herramientas disponibles en PictoBlox, sino que también actúa como una guía visual que ayuda a los usuarios en la navegación y el uso eficiente del software.



Figura 9.1. Herramientas de PictoBlox. Fuente: Adaptado a partir de PictoBlox (2019).

Cabe mencionar que, para un funcionamiento óptimo, las herramientas de esta aplicación requieren extensiones específicas de PictoBlox, las cuales Sotomayor (2020) caracteriza como complementos esenciales para desarrollar proyectos con IA. Estas extensiones enriquecen la plataforma al proporcionar funcionalidades adicionales que expanden las capacidades del software más allá de su uso convencional. Para facilitar una comprensión detallada de cada herramienta y extensión de PictoBlox, la Tabla 9.2 ofrece una descripción de sus funciones y aplicaciones prácticas, facilitando así a los usuarios aprovechar al máximo las posibilidades que ofrece este entorno de programación visual.

Tabla 9.2. Extensiones IA de Pictoblox.

Extensión	Descripción
Detección de cara	Esta extensión permite a los estudiantes crear aplicaciones de detección de rostro, con cámaras sencillas.
Detección del cuerpo humano	Permite el desarrollo de actividades de interacción, utilizando como referencia el cuerpo humano.
Aprendizaje automático	Extensión que permite a que la máquina sea capaz de aprender de forma continua y al ritmo de su desarrollador.

Extensión	Descripción
Inteligencia artificial	Esta extensión une aplicabilidades como reconocimiento de voz, de audio y de rostro de forma completa.

Fuente: Adaptado a partir de PictoBlox (2019).

Para la aplicación del software educativo Pictoblox, en la Tabla 9.3 se presenta la codificación y la aplicación de la inteligencia artificial en el proyecto “Piedra, Papel o Tijera”. Este software facilita la enseñanza de habilidades, destrezas cognitivas, psíquicas y psicomotrices de niños entre 5 a 12 años.

Tabla 9.3. proyecto “Piedra, Papel o Tijera” en PictoBlox.

Tema	Imagen
Código del proyecto “Piedra, Papel o Tijera” en PictoBlox.	
Aplicación en PictoBlox proyecto “Piedra, Papel o Tijera”	

Fuente: Adaptado a partir de PictoBlox (2019).

Crear un juego de Piedra, Papel o Tijera en PictoBlox con IA representa una emocionante oportunidad de fusionar entretenimiento y enseñanza, tal como lo señala LearningML (s.f.). Este proyecto, ejemplificado por el autor, demuestra que la aplicación no solo brinda diversión, sino que también brinda una experiencia educativa interactiva. Al fusionar el juego con la tecnología, se promueve el pensamiento estratégico y se facilita la familiarización con herramientas tecnológicas avanzadas.

Reflexiones

El uso de la IA en la educación representa un avance significativo, permitiendo a los docentes incorporar estrategias activas como el pensamiento computacional, la gamificación y la metodología STEAM. Estas estrategias ayudan a adaptar las nuevas tecnologías al proceso de aprendizaje. PictoBlox, en particular, es un programa educativo que fomenta el interés en la programación y mejora las habilidades de pensamiento computacional, ofreciendo a los estudiantes la posibilidad de crear proyectos con principios de IA de manera sencilla e intuitiva.

PictoBlox, como plataforma de programación visual basada en bloques que transforma la educación y el aprendizaje. Su interfaz intuitiva y amigable favorece a los estudiantes más jóvenes o aquellos sin experiencia previa en programación participen de manera activa y creativa en la resolución de problemas y la creación de proyectos.

Al integrar PictoBlox en el aula posibilita que los educadores promuevan habilidades como el pensamiento computacional, la resolución de problemas, la creatividad y la colaboración entre los estudiantes. Además, brinda a los estudiantes la oportunidad de explorar conceptos de las disciplinas STEAM: Ciencia, Tecnología, Ingeniería, Arte y Matemáticas de una manera práctica y significativa.

Bibliografía

- Agilo, R. (2020a). Download PictoBlox Software. STEMpedia. <https://thestempedia.com/product/pictoblox/download-pictoblox/>
- Ahumada, H., Rivas, D., Contreras, N., Miranda, M. y Poliche, M. (2018). Pensamiento computacional mediante programación por bloques: intervención didáctica usando pilas bloques. In XIII Congreso de Tecnología en Educación y Educación en Tecnología (Posadas, 2018).
- Bordignon, F. y Iglesias, A. (2020). Introducción al pensamiento computacional. Universidad Pedagógica Nacional y Educar SE.
- Cordero, F. (2018). Diseño de interfaces gráficas para recursos didácticos digitales. DAYA. Diseño, Arte y Arquitectura, 5 (1), 11 – 29. https://revistas.uazuay.edu.ec/flip/daya/05/daya_05_01.pdf
- Ferrada, C., Carrillo, F., Díaz, D. y Silva, F (2020). La robótica desde las áreas STEM en Educación Primaria: una revisión sistemática. Educación en la sociedad del conocimiento: EKS.
- García, R., Bonilla, M. y Diego, J. M. (2018). Gamificación en la Escuela 2.0: una alianza educativa entre juego y aprendizaje. Gamificación en Iberoamérica, 71-95.
- García, Y., Reyes, D y Burgos, F. (2017). Actividades STEM LearningML. (s.f.). Actividad – Piedra, Papel o Tijeras. LearningML - AI made easy. <https://web.learningml.org/actividad-piedra-papel-o-tijeras/>
- López, J. y Gutiérrez, R. (2017). Programación visual por bloques en Educación Primaria: Aprendiendo y creando contenidos en Ciencias Sociales. Revista Complutense de Educación, 28(2), 49.

- Maloney, J., Kafai, Y., Resnick, M. y Rusk, N. (2008). Programming by choice: urban youth learning programming with scratch. En 39th SIGCSE Technical Symposium on Computer Science Education. (pp. 367-371).
- Padilla, R. (2019). La llegada de la inteligencia artificial a la educación. Revista de Investigación en Tecnologías de la Información: RTI, 7(14), 260-270.
- PictoBlox. (2019). PictoBlox(20230707) <https://pictoblox.ai/>
- ProFuturo. (2022). Enseñar a programar en las escuelas: ¿necesidad o lujo?. <https://profuturo.education/observatorio/tendencias/ensenar-a-programar-en-las-escuelas-necesidad-o-lujo/>
- Sotomayor, J. (2020). Extensión de Inteligencia Artificial en PictoBlox. Semillero de Innovación. <https://comunidad.semillerodeinnovacion.com/topic/extension-de-inteligencia-artificial-en-pictoblox-copy/>
- Wing, J. (2006). Computational thinking. Communications of the ACM, 49(3), pp. 33-35. <https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>

Capítulo 10

Desarrollo de la lógica de programación en edades tempranas con Bee-Boot

Granda Lima Luisa Victoria, Villamagua Vicente Sandra del Carmen.

Universidad Nacional de Loja.

luisa.granda@unl.edu.ec, sandra.villamagua@unl.edu.ec

Introducción

Las Tecnologías de la Información y Comunicación (TIC) han transformado los ambientes educativos, creando nuevos escenarios innovadores e integrando recursos con fines educativos para potenciar el razonamiento lógico durante su formación académica. Entre los recursos actuales, el robot denominado Bee-Bot es una opción que despierta mayor interés en la comunidad educativa, especialmente en los niños pequeños por su peculiar forma de abeja que posee. El robot otorga a los estudiantes desarrollar el pensamiento lógico y fomentar un criterio autónomo a través de la botonera ubicada en la parte superior donde se digitán los comandos para ser ejecutado; además, su aplicación móvil descargable ofrece práctica digital.

En este sentido, en la etapa de la niñez, la inclusión de distintos recursos de índole académico debe estimular la comprensión de contenidos, la superación de retos y la solución de problemas cotidianos, poniendo a prueba sus conocimientos, habilidades y razonamiento lógico. En este contexto, la implementación de Bee-Boot para desarrollar la lógica es una alternativa aplicable en niños. Este capítulo contextualiza al robot como una alternativa pedagógica en el aula, lo que permitirá el

desarrollo de la lógica de programación desde una edad temprana y anticipando experiencias educativas innovadoras mediante la inclusión en diversas áreas de estudio.

Desarrollo de la lógica en edades tempranas

La lógica busca expresar de forma clara y precisa términos, concepciones o expresiones. Además, determina las características esenciales de un tema o concepto haciendo uso del pensamiento crítico. De acuerdo con Díaz (2021), el pensamiento lógico hace referencia al uso de la razón, donde el ser humano es idóneo al analizar la información que capta por medio de los sentidos, posibilitando establecer conclusiones a través de las experiencias y deducciones, seguir una secuencia de reglas o pasos para llegar a conclusiones; Así, la lógica permite comprender la realidad del entorno que nos rodea y ser capaces de resolver los problemas que puedan presentarse utilizando el razonamiento y el pensamiento lógico.

En este sentido, Díaz (2021) enfatiza que el desarrollo de esta habilidad en los niños es un proceso gradual y espontáneo, que va desde lo concreto hasta lo abstracto, hasta llegar a ser capaces de hacer sus propios razonamientos hipotético-deductivos y extraer sus propias conclusiones a partir de sus experiencias vividas. Por lo tanto, los infantes son capaces de razonar de forma concisa y dar soluciones a problemas que se presenten.

Asimismo, de acuerdo con un estudio realizado por la Universidad Internacional de la Rioja (UNIR, 2020), señala que Piaget en su teoría dividió el desarrollo cognitivo en 4 fases; la primera fase sensomotora, comprende desde el nacimiento hasta los 2 años en la cual, el infante por primera vez entra en contacto con el entorno que lo rodea desarrollando específicamente el sentido del tacto y estableciendo sus primeras relaciones causa-efecto.

Seguidamente, en la segunda fase conocida como pre-operacional, que abarca entre los 2 a los 6 años, el razonamiento del niño comienza a ser más fluido, siendo capaces de desarrollar sus propias deducciones y conclusiones a partir de los problemas cotidianos del entorno. Mientras que la tercera etapa, denominada Operaciones concretas aborda desde los 6 hasta los 11 años, comienzan a pensar de manera más abstracta, llegar a conclusiones y establecer hipótesis.

Finalmente, en la cuarta etapa de las Operaciones formales, a partir de los 12 años, el adolescente irá adquiriendo competencias cada vez más complejas y diversas que le permitirán hacer sus propios razonamientos hipotético-deductivos, abstractos y fuera del marco espacio-temporal en el que se encuentra.

Desarrollo de la lógica en programación

La programación es una habilidad fundamental para los niños que están inmersos en la sociedad del conocimiento y la información, ya que fomenta el desarrollo del pensamiento lógico y la capacidad de identificar problemas y analizar posibles soluciones. En este contexto, Sampedro et al. (2021) afirma que la programación a edades tempranas facilita en los niños, desde los 5 y 6 años, optimicen habilidades y desarrollos destrezas en el ámbito de relaciones lógicas para dar soluciones a conflictos que se presenten en su entorno.

De este modo, el Plan Nacional de Calidad en Educación Parvularia (2019) comparte iniciativas y programas pedagógicos que potencian el razonamiento y la capacidad de resolver problemas, promoviendo así el desarrollo del pensamiento lógico y el análisis de soluciones, lo que impulsa a niños adquirir nuevas capacidades con la finalidad resolver las dificultades que se presenten en su entorno.

Desarrollo de la lógica en programación con Bee-Bot

La introducción de herramientas revolucionaria en el campo de la educación primaria, como Bee-Bot, se presenta como un robot programable que va más allá de la simple interacción. Según Abellán (2019), este robot está diseñado para el público infantil, incentivando la secuenciación lógica de comandos en forma de algoritmos, el desarrollo de la orientación espacial y la lateralidad, así como la generación de soluciones creativas y eficientes. Además, estimula el trabajo individual o colectivo, facilitando a los estudiantes reflexionar, anticipar, ensayar y comprobar mediante un aprendizaje basado en el pensamiento y la experimentación.

Teniendo en cuenta lo señalado por Tibot (2021), el Bee-Bot es un robot analógico y cuenta con botones en su parte superior que facilita programarlo para avanzar o retroceder (15 cm en cada paso) o girar a la derecha o izquierda (90°) sobre sí mismo, como se observa en la Figura 10.1. También, cuenta con el botón “GO” que facilita al robot ejecutar las órdenes introducidas, “CLEAR” que borra los comandos introducidos previamente; es importante recalcar que el robot tiene memoria y después de pulsar el botón “GO”, las órdenes introducidas se vuelven a ejecutar y el botón “PAUSE” detiene temporalmente el robot.



Figura 10.1. Robot programable para primeras edades escolares. Fuente:
Adaptado de Diago y Arnau (2017).

El robot Bee-Boot integra diferentes materiales físicos como se refleja en la Tabla 10.1, que otorga crear el espacio de trabajo donde los niños puedan utilizar el robot y familiarizarse con él; esto contribuye al desarrollo de la lógica, el razonamiento y la resolución de problemas.

Tabla 10.1. Materiales físicos de la Bee-Boot.

Materiales físicos	Descripción
Tablero	Las actividades con el robot Bee-Boot vienen acompañadas de tableros con cuadrículas pautadas (de 15 cm) sobre los que se desplazará el robot.
Cinta adhesiva de color	Esta cinta se utilizará para marcar los caminos que el robot ha de recorrer sobre el tablero.
Tarjetas de comandos y caja de secuenciación	Con el fin de que los estudiantes puedan representar un plan y compararlo con el movimiento del robot cuenta con tarjetas que corresponden a los comandos que puede ejecutar Bee-Bot.

Fuente: Elaboración del autor.

Considerando este tema, para enriquecer la comprensión de la orientación espacial en la programación del robot, se incorpora en las tarjetas de comando la silueta del Bee-Bot. Esto posibilita a los estudiantes visualizar y planificar los movimientos del robot en relación con su posición actual, facilitando así una mejor comprensión de las secuencias de programación. La Figura 10.2 ilustra estas técnicas, mostrando cómo la representación gráfica del robot en las tarjetas puede ser utilizada para diseñar rutas.

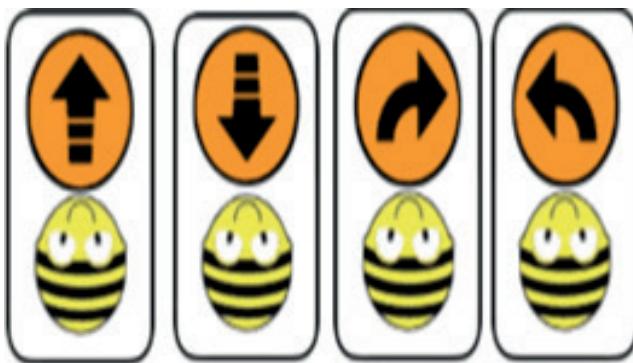


Figura 10.2. Tarjetas de comandos correspondientes a los movimientos que realiza Bee-Boot. Fuente: Adaptado de Diago y Arnau (2017).

En efecto, las tarjetas disponen el orden y secuencialidad de las instrucciones que el robot debe ejecutar dentro de la caja de secuenciación. Igualmente, contiene varios espacios numerados donde se insertará cada una de las tarjetas. Una flecha indica el orden de ejecución de los comandos como se refleja en la Figura 10.3. Finalmente, se acompaña con una tarjeta con el comando “GO” que indica la ejecución del programa y que los estudiantes deben colocar al final de la secuencia elaborada para indicar el fin del algoritmo.

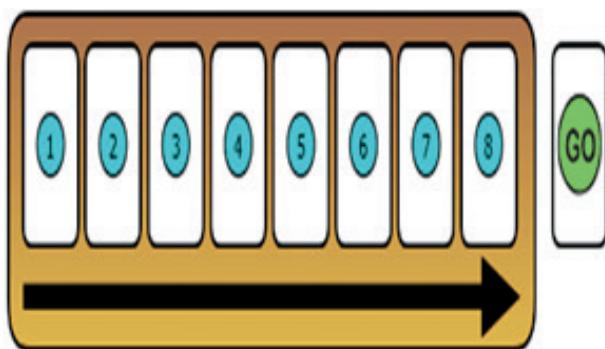


Figura 10.3. Caja de secuenciación sobre la que se colocan las tarjetas de comandos. Fuente: Adaptado de Diago y Arnau (2017).

Casos de Estudio y Experiencias Reales con Bee-Bot

La programación de dispositivos robóticos, como el Bee-Bot, ofrece beneficios significativos a personas con discapacidad visual, permitiéndoles desarrollar habilidades más allá de la percepción visual de formas y texturas. Según (Spinarova y Vachalova, 2021), esta práctica también estimula el razonamiento, cálculo, retención, procesamiento espacial y algorítmico. Además, estudios como el de Schina et al. (2021) han confirmado que el Bee-Bot no solo ayuda a estudiantes con discapacidad visual, sino que también puede ser muy útil para niños con trastorno por déficit de atención y dislexia.

Así vez, Chaldi y Mantzanidou (2021) destaca que algunos niños autistas son no verbales, requiriendo métodos alternativos de comunicación; por ende, sugieren que la programación de dispositivos robóticos, como el Bee-Bot, beneficia a personas con discapacidad visual al facilitar la comunicación aumentativa. Este enfoque también beneficia a niños autistas al mejorar sus habilidades matemáticas y adaptarse a diferentes estilos de aprendizaje. El Bee-Bot no solo ayuda a reconocer formas y texturas, sino que también desarrolla habilidades de razonamiento y procesamiento espacial.

En énfasis, el Bee-Bot no solo motiva a los estudiantes, aumenta su confianza y les brinda una sensación de logro al aprender sobre conceptos como el tiempo, los números y las direcciones, sino que también es beneficioso para niños con diversas necesidades y habilidades en entornos educativos. La inclusión de tecnologías como el Bee-Bot puede ser especialmente valiosa para niños en el espectro autista, ya que ayuda a mejorar sus habilidades comunicativas y participar en actividades de procesamiento matemático.

En general, el uso de dispositivos robóticos en la educación puede mejorar significativamente el aprendizaje y la participación de todos los niños, independientemente de sus capacidades y necesidades, fomentando una mayor interacción y compromiso entre los alumnos en comparación con enfoques tradicionales.

Aplicación digital Bee-Boot

El potencial del robot Bee-Boot permite trasladarse hasta el mundo digital, como se plasma en la Figura 10.4. Asimismo, ofrece una multitud de tableros de juego con diferentes niveles de complejidad. Así, en cada nivel aumenta la dificultad, de manera que potencie el desarrollo lógico y el razonamiento para alcanzar el objetivo en el nivel de juego aplicando diferentes secuencias que conlleven a alcanzar la meta propuesta.



Figura 10.4. Vista de los niveles y escenarios interactivos de Bee-Boot. Fuente: Adaptado de la aplicación digital Bee-Boot (2024).

En relación con la experiencia utilizando la aplicación Bee-Bot, su interfaz interactiva proporciona instrucciones claras para escribir algoritmos y observar la ejecución de las instrucciones. A medida que se progresan en los niveles, se emplean habilidades de pensamiento computacional para guiar a Bee-Bot hacia la consecución de sus objetivos, resultando la ejecución de los algoritmos fascinante y de mucha interacción en los diversos niveles y escenarios disponibles.

Reflexión

El capítulo busca destacar la relevancia de integrar el robot Bee-Bot en la educación, con el propósito principal de estimular y cultivar el pensamiento lógico en los niños a través de la programación de una manera divertida. Al introducir a los niños en escenarios dinámicos y desafiantes, se les capacita para resolver problemas que el docente presente, a su vez enriquece la práctica pedagógica del profesor al incorporar este recurso innovador. La inclusión de Bee-Bot en el aula no solo fomenta el aprendizaje de habilidades tecnológicas, sino también promueve el desarrollo de habilidades cognitivas y sociales importantes para el éxito en el mundo moderno.

Bibliografía

- Abellán, M. (2019). Introducción a la robótica educativa con Bee-Bot. <https://www.programoergosum.es/tutoriales/robotica-educativa-con-beebot/>
- Chaldi, D. y Mantzanidou, G. (2021). Mejora de la habilidad de respuesta del oyente utilizando Bee-Bot® en el trastorno del espectro autista: estudio de caso. Revista europea de investigación en educación especial, 7 (2).
- Diago, P. y Arnau, D. (2017). Pensamiento computacional y resolución de problemas en educación infantil: una secuencia de enseñanza con el robot bee-bot. viii congreso iberoamericano de educación matemática. <https://n9.cl/tzolnb>
- Díaz, S. (2021). El desarrollo de la lógica en los niños: cómo evoluciona y qué actividades y juegos podemos hacer para fomentarla. Bebés y más. <https://n9.cl/pmley>
- Plan Nacional de Calidad en Educación Parvularia. (2019). Bee Bot Recurso pedagógico. Ministerio de educación. http://clubbeebot.cl/wp-content/uploads/2020/09/Manual_Bee-bot.pdf
- Sampedro, M., Pabón, D., Maiguashca, J. y Guerrón, E. (2021). Programación infantil y desarrollo del ámbito de relaciones lógico-matemáticas en niños de Educación Primaria: Enseñanza con Bit by Bit. Revista Cognosis. Vol. (VII). 39-54. <https://revistas.utm.edu.ec/index.php/Cognosis/article/view/3577/4386>

Schina, D., Gonzalez, V. y Usart, M. (2020). Percepciones de los profesores sobre el juguete robótico bee-bot y su capacidad para integrarlo en su enseñanza. En Conferencia internacional sobre robótica en la educación (RiE) (121-132). Cham: Editorial Internacional Springer.

Spinarova, G. y Vachalova, V. (2021). La tecnología digital como apoyo a los procesos cognitivos de personas con discapacidad visual. En sociedad. Integración. educación. Actas de la Conferencia Científica Internacional (Vol. 3. 195-204).

Tibot. (2021). Que es Bee Bot. <https://www.tibot.es/blog/productos/que-es-bee-bot>

Universidad Internacional de la Rioja [UNIR]. (2020). Desarrollo del pensamiento en niños: importancia y metodologías. <https://ecuador.unir.net/actualidadunir/desarrollo-del-pensamiento-en-ninos-importancia-y-metodologias/>

Capítulo 11

Aprendizaje de la programación a través de la gamificación en la herramienta Grasshopper

Maria Alexandra Ortega Medina.

Universidad Nacional de Loja.

maria.a.ortega.m@unl.edu.ec

Introducción

En la actualidad, la tecnología se ha convertido en un elemento fundamental de nuestra vida diaria y su influencia se extiende a diversos ámbitos, incluida en la educación; en este contexto, los estudiantes buscan que los docentes adopten nuevas técnicas y herramientas dinámicas en el aula para aumentar su motivación y facilitar su aprendizaje. Por tal razón, la gamificación incorporada en cursos de programación emerge como una técnica prometedora para maximizar la participación y generar un impacto positivo en el proceso de aprendizaje. Esta innovación permite que la etapa de programación sea menos tediosa y más dinámica, adaptándose así a las necesidades y expectativas de los alumnos.

Por tanto, se logrará alcanzar y mejorar el proceso educativo implementando un software de aplicación didáctico, tomando en cuenta aspectos fundamentales como recompensa, status, logro y competición al momento de elegir una herramienta; con el propósito que el educando al aprender nuevos conceptos le sea atractivo. Finalmente, en el presente trabajo pretendo dar a conocer una herramienta que otorga un abanico de posibilidades para aprender a programar el lenguaje JavaScript y conceptos que puedan aplicarse a cualquier lenguaje como HTML y CSS.

Programación en la educación

En una época caracterizada por la constante innovación tecnológica, es fundamental que los estudiantes desarrollen habilidades acordes al siglo XXI. De acuerdo con Rasgado (2018), la programación emerge como una herramienta fundamental para cultivar diversas formas de pensamiento, incluyendo el creativo, el computacional y el crítico. Al dominar la programación, los estudiantes no solo están mejor preparados para afrontar los desafíos del futuro, sino que también adquieren una comprensión profunda y un control sobre las tecnologías que forman parte integral de su entorno.

Cabe considerar que, para dominar la programación, es crucial entender y aplicar una gama de conceptos básicos y avanzados que son fundamentales para el proceso de programación. Esto implica iniciar con lo básico y avanzar hacia lo complejo, desde el diseño de algoritmos y diagramas de flujo hasta el uso de lenguajes de programación y plataformas programables. En este contexto, Palma et al. (2015) recomienda que, al usar herramientas basadas en la web o aplicaciones, es preferible optar aquellas con una sintaxis intuitiva y en el idioma del aprendiz; además, debe incluir una interfaz amigable y depuración eficiente, con mensajes de error que ayuden a resolver problemas.

Ante este panorama, resulta evidente la necesidad de superar la brecha cultural tecnológica existente entre los docentes en ejercicio. Tadeu (2020) y Gutiérrez et al. (2022) coinciden en la importancia crítica de implementar programas de capacitación tecnológica para educadores, con el propósito de facilitar una integración efectiva de la tecnología en la enseñanza. Por tanto, Rasgado (2018) ratifica que la introducción de la programación en el ámbito educativo demanda una constante renovación y capacitación de los docentes. Esta formación es indispensable para proporcionar a los educadores las competencias y herramientas necesarias que les permitan profesionalizar y revitalizar su enfoque pedagógico.

Gamificación

En el ámbito educativo las Tecnologías de la Información y la Comunicación (TIC) desempeñan un papel crucial al desafiar las barreras tradicionales y fomentar un aprendizaje más constructivo, donde los participantes interactúan y colaboran para construir conocimiento. Como mencionan Revelo et al. (2017), la gamificación transforma el aprendizaje de Programación de Computadoras con un ambiente de aprendizaje más interactivo y motivado. Por ende, la gamificación emerge como una técnica innovadora ya que utiliza elementos del diseño de videojuegos para mejorar el proceso de enseñanza y aprendizaje.

Como considera González (2016), la gamificación es una técnica de aprendizaje que involucra al estudiante a motivar, estudiar, incentivar el ánimo de superación, ejercitar y despertar un interés especial, ya que disfruta jugando. De modo que, es importante reconocer que este tipo de aprendizaje genera capacidades en los alumnos a tal punto de acrecentar sus conocimientos, recordar conceptos, superar objetivos complejos y mantener la concentración hasta lograr un estado de inmersión total jugando.

Cabe considerar que las principales dinámicas aplicadas en la gamificación que impulsa al educando, con base a Educativa (2013) se encuentra en las recompensas, un valor que se logra al cumplir una actividad del juego que se obtiene premios como: puntos, medallas, acceso a nivel superior y otros; status, un factor fundamental que establece un nivel jerárquico social valorado; logros, conseguir un objetivo con cierto nivel de dificultad como superación o satisfacción personal; competición, medir el desempeño individual del jugador, comparando los resultados con los demás. Estos componentes son fundamentales para impulsar la participación y el compromiso del usuario en el proceso educativo.

Por ende, la gamificación en la educación transforma el aprendizaje en una experiencia más atractiva y manejable, simplificando las actividades difíciles, creando una retroalimentación positiva a través de premios, promoviendo el triunfo y la perseverancia. Esto permite que tanto estudiante como docente puedan hacer un seguimiento sobre su rendimiento y progreso.

Gamificación aplicada a la enseñanza de programación

En la era digital, los videojuegos se han convertido en una parte integral de la vida cotidiana de los niños y adolescentes, quienes interactúan con una variedad de dispositivos. En este punto, Astudillo et al. (2016) destacan la relevancia de esta interacción, señalando que la gamificación no solo captura el interés de los estudiantes, sino que también ofrece un marco dinámico para el aprendizaje. Los educadores juegan un rol esencial en este proceso, orientando a los alumnos hacia un aprendizaje significativo que abarca una amplia gama de temas. Esto es evidente en la enseñanza de la programación, donde la gamificación facilita que los estudiantes absorban conocimientos y exploren innovadores entornos de aprendizaje.

Cabe mencionar, que trabajar con plataformas interactivas significa aprender a programar varios lenguajes desde opciones básicas y sencillas de manera significativa. Pulido (2015), afirma que tiene más de una finalidad dentro del ámbito de la programación, como se detalla a continuación:

- El alumno no teme a la programación como algo complicado, sino como motivador y divertido para ellos.
- Reducir la tasa de abandono de alumnos que se inician en la programación.

- Promover el aprendizaje significativo.
- Llevar la programación a edades más tempranas para que aprendan a resolver problemas e incentivar su creatividad.

Sin embargo, las clases de programación aún carecen de pasión de querer enseñar porque algunos docentes no tienen dominio sobre sus contenidos o realizan una elección inadecuada de herramientas. Para superar esto, deben ser capaces de explorar nuevas técnicas dinámicas e interactivas con el objetivo de aprender a programar, a medida que el estudiante al crear, experimentar y probar diversas alternativas, vaya fomentando su creatividad.

Programando con Juegos

Los videojuegos son fundamentales en la cultura actual, influyendo en la juventud no solo como entretenimiento, sino también como medios de expresión y comunicación, contribuyendo al desarrollo social y cognitivo. La integración de los juegos en el contexto educativo en torno a la enseñanza de programación, tal como menciona Queiruga et al. (2014), sumerge a los estudiantes en un entorno familiar. Esto les facilita no solo la creación, prueba y juego de sus propias creaciones, sino también adquirir habilidades de programación usando un instrumento innovador: Grasshopper, es una herramienta tecnológica óptima para la transmisión eficaz de conocimientos.

La herramienta Grasshopper

Grasshopper es una herramienta que tiene como propósito enseñar programación a principiantes de forma dinámica a través de un pequeño saltamontes, como se aprecia en la Figura 11.1. Según el Instituto Nacional de Tecnologías Educativas y Formación del Profesorado (INTEF, 2018), esta aplicación es desarrollada por Área 120 de Google,

no solo rinde homenaje a Grace Hoppe, una figura icónica en el campo de la informática, sino también refleja la naturaleza accesible y amigable de la herramienta.

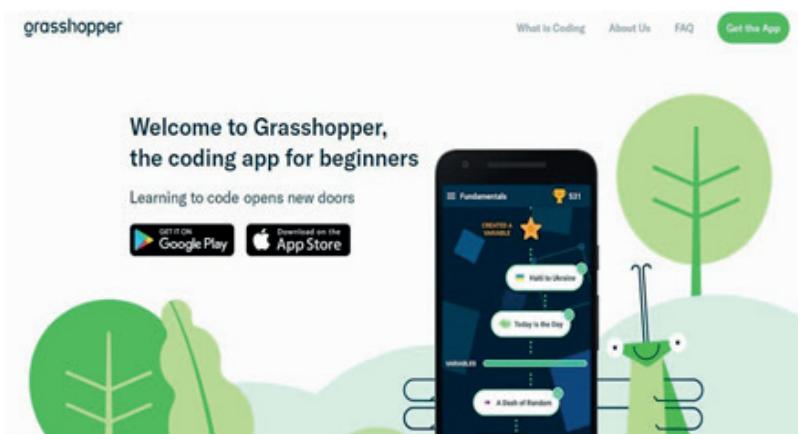


Figura 11. 1. Herramienta Grasshopper. Fuente: Adaptado a partir de Grasshopper.

En este sentido se comprende que Grasshopper se ha convertido en un pilar fundamental para la enseñanza, destacándose por su enfoque innovador y su adaptabilidad a las necesidades de aprendizaje modernas. Como señala Noguera (2018), este software permite aprender a programar mediante unidades de conocimiento básicos, resolución de problemas y construir una página web simple, centrándose en la codificación de cosas reales, evitando clases tediosas. Los usuarios avanzan de nivel en el juego al completar cuestionarios simples, resolver rompecabezas visuales y participar en pequeñas actividades, enfrentándose a desafíos de mayor dificultad a medida que progresan.

Características de la aplicación Grasshopper

Sumergirse en el mundo de la programación puede ser una experiencia tan desafiante pero no imposible. En este contexto, Grasshopper se convierte en una opción indispensable para aquellos que desean

aprender a programar con un enfoque visual tipo saltamontes que hacen que este proceso sea intuitivo y efectivo. A continuación, la Compañía de Responsabilidad Limitada (Google LLC, s.f.), presenta las siguientes características de la herramienta, como se describen en la Tabla 11.1, las cuales incluyen lo siguiente.

Tabla 11.1. Herramienta Grasshopper.

Características
Es una app gratuita, que se encuentra disponible en iOS, Android enfocado en proporcionar una experiencia solo en inglés o español.
Fue diseñado para enseñar a los principiantes programadores a que dominen los conceptos básicos de programación y aprendan a escribir JavaScript a través de niveles.
Se adapta según a tu nivel de conocimiento, por ende, al momento de acceder en la herramienta te interroga.
Cuenta con recordatorios diarios, notifica las actualizaciones e incluye frases de motivación evitando las clases pesadas, y un plan de estudios similar al juego para seguir aprendiendo programación.
Dispone de un área de galería, que permite explorar nuestras habilidades de codificación en práctica de lo aprendido, para crear nuevos fragmentos o editar eventos especiales incluyendo el lenguaje marcado HTML y el estilo CS.

Fuente: Adaptado a partir de Google LLC (s.f.).

Por tanto, se podría definir que la gamificación tiene el potencial de involucrar a los estudiantes a enseñar nuevos conceptos. Como señala Mora et al. (2022), los principales beneficios en el proceso de aprendizaje son visualizar que están progresando con el concepto aprendido. Esto motiva a los estudiantes a esforzarse por dominar los conceptos, es decir, que la herramienta contiene las características necesarias que llaman la atención al usuario, creando un ambiente de aprendizaje sano y adquisición de nuevos conocimientos para superarse a sí mismo, cada vez que adquiere un nuevo nivel en el juego.

Estructura de la herramienta Grasshopper

En un mundo digital en constante evolución, el acceso a herramientas tecnológicas es fundamental para fomentar la creatividad y la innovación. Estévez (2020), describe que Grasshopper se rige como un ejemplo destacado que simplifica la programación para principiantes mediante un enfoque lúdico y estructurado, sino también convierte la complejidad de la programación en un sistema visual comprensible y manejable. Al iniciar el juego, se divide la pantalla en 3 secciones, como se ilustra en la Tabla 11.2, que se hace presente en los diferentes niveles. Esta estructura facilita el aprendizaje del desarrollo de software de manera efectiva y amena.

Tabla 11.2. Descripción de las características de las secciones.

Sección	Características
Izquierda	Se encuentran distintas instrucciones de cada nivel que presenta el juego, explicaciones y aclaraciones para poder solventarlo.
Derecha	Demuestra lo que haga en el editor se irá reflejando, es decir, se refleja el resultado del código que vayas escribiendo. Esto se encuentra en los diferentes niveles.
Derecha	Demuestra lo que haga en el editor se irá reflejando, es decir, se refleja el resultado del código que vayas escribiendo. Esto se encuentra en los diferentes niveles.
Centro	Está la parte del editor tendrá que escribir el código y en la parte abajo aparecerán una serie de atributos o propiedades que tendrás que usar en el editor para realizar lo que pide la lección.

Fuente: Adaptado a partir de Estévez (2020).

Niveles

A través de la herramienta Grasshopper se aprecian diferentes niveles que se encuentran establecidos en la Tabla 11.3. En cada uno, la estructura está cuidosamente diseñada para construir sobre el conocimiento previo, asegurando que los usuarios ganen confianza y habilidades de manera gradual mientras se va completando los distintos desafíos y retos que se plantean. A continuación, describo la función de cada nivel:

Tabla 11.3. Niveles de la herramienta Grasshopper.

Nivel	Función
Fundamentos	Proporciona funciones, crear variables, cadenas, controla el flujo de códigos y entre otros.
Fundamentos II	Al adquirir conocimientos de Fundamentos, nos servirá para aprender como manipular strings y arreglos.
Métodos de arreglo	Presenta distintas formas de buscar arreglos, combinar elementos y transformar elementos mediante las funciones de flecha.
Animaciones	Permite aprender cómo crear, manipular y animar formas. Este curso usa la biblioteca D3, gráficos vectoriales escalables (SVG) y JavaScript.
Animaciones II	Indica como crear animaciones interactivas, manipulables y crear funciones más complejas.

Fuente: Adaptado a partir de Google LLC (s.f.).

Cabe mencionar que la aplicación ofrece instrucciones detalladas en cada nivel con el propósito de evitar confusión entre los participantes ante algunas acciones que pueden estar abiertas a diferentes interpretaciones; incluso, provee una guía necesaria en caso de no resolver las actividades

o presentar inconvenientes. En tal caso, se puede acceder al código de la solución inmediata y proceder con el mismo, pero se obtendrá un acertijo diferente como retroalimentación para cumplir y proceder al siguiente nivel del juego.

Funciones

En cada uno de los niveles previamente mencionados, se pueden observar diversas funciones de programación disponibles en la herramienta Grasshopper (Google LLC, s.f.). Esto se detalla más ampliamente en la Tabla 11.4, donde se especifican las funciones específicas disponibles en cada nivel para abordar las distintas actividades propuestas

Tabla 11. 4. Funciones de la herramienta Grasshopper.

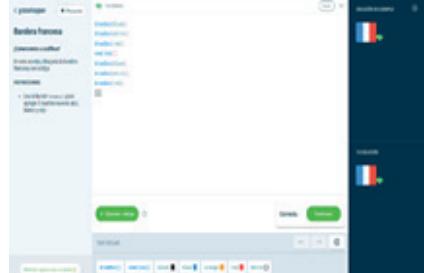
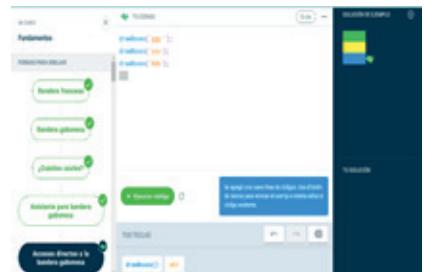
Nombre	Función
Print ()	Esta función toma el argumento, la parte dentro del paréntesis, y lo imprime, luego mueve el cursor a una nueva línea.
PickRandom ()	Toma matrices o números, o variables que contienen números o matrices, y selecciona un elemento aleatorio.
BrawBox ()	Toma colores y los convierte en cuadrados (rectángulos SVG). Puede tomar cadenas de colores o variables que contengan cadenas de colores
BrawBoxes()	Dibuja un cuadrado para cada letra proporcionada. Cada letra corresponde a un color. Por ejemplo, ‘b’ es para ‘azul’ y ‘o’ es para ‘naranja’ y un espacio permitirá mover los cuadros a una nueva línea.
NewLine ()	Mueve el cursor, el pequeño saltamontes, a la línea siguiente.
Str	Se utiliza para agregar una cadena de letras, palabras o frases acompañadas de comillas para mostrar el inicio y el final de un string. Por ejemplo: ‘Hola mundo’.

Fuente: Adaptado a partir de Google LLC (s.f.).

Experimentación de la herramienta Grasshopper

Como aplicación práctica de este estudio, se llevó a cabo una exploración detallada del recurso Grasshopper, resultado de lo cuál se describe en la Tabla 11.5 tres actividades específicas, junto con sus objetivos y las soluciones detalladas mediante el uso de funciones correspondientes. Estas soluciones están diseñadas para guiar al personaje saltamontes en la ejecución de las tareas requeridas, incluyéndose además imágenes que ilustran lo obtenido durante el desarrollo de cada actividad.

Tabla 11.5. Actividades del nivel de fundamento.

Actividad 1		
Objetivo	Descripción de la solución	Imagen
Crear una bandera francesa	La función drawBox (white) dibujará un cuadro y dentro de ella se dibujará el cuadro de color blanco; newLine () mueve el resultado a una nueva línea para que los cuadros dibujados. La bandera francesa en este rompecabezas debe haber seis drawBox() y un newLine() para que los cuadros estén uno encima del otro.	
Crear la bandera Gabonesa	Para dibujar la bandera gabonesa, necesita tres filas: verde, verde, verde, nueva línea, amarillo, amarillo, amarillo, nueva línea, luego, azul, azul, azul. Esto significa que el string que va dentro de drawBoxes () es ‘ggg yyy bbb’.	

Fuente: Adaptado a partir de Google LLC (s.f.).

Mediante la realización de estas actividades, se determinó que permite aprender significativamente conceptos de programación. Durante su uso se propone retos mientras se completan los niveles y se va tornando más complejo a medida que mejoran las habilidades; además, es posible repetir los niveles anteriores las veces que se desee para recordar conocimientos; cabe destacar que son sencillas para iniciar a programar y otorgan la ayuda necesaria para seguir avanzando cada nivel e incluso se puede desarrollar sin conexión a internet.

Por tanto, esta herramienta utiliza la técnica de la gamificación con el propósito de generar impactos positivos al implementarla en el aprendizaje desde diversos ángulos, como en logros, participación y motivación en las actividades de aprendizaje para aprender a programar.

Reflexiones

En la actualidad, los videojuegos forman parte esencial del día a día de niños y adolescentes, quienes utilizan una diversidad de dispositivos para su entretenimiento y educación. La gamificación se convierte en una técnica esencial en este contexto, mejorando la capacidad y motivación de los estudiantes de forma continua, sin desviar el interés en el aprendizaje y practicar habilidades de lo aprendido de forma que directamente satisface ciertas necesidades y deseos básicos. Esto requiere que los docentes estén capacitados para que adopten nuevas técnicas y herramientas que facilite el aprendizaje de la programación

Considerando el capítulo expuesto, mediante su uso y su funcionamiento, se estima que Grasshopper es una herramienta extraordinaria con un interfaz amigable para quienes deseen iniciar o reforzar sus conocimientos de programación; puesto que es motivadora, favorece la adquisición de conocimientos, alienta a la práctica y la habilidad para solucionar problemas de maneras sencillas, sin importar su nivel

de experiencia. Además, puedan tomarse todo el tiempo necesario en descubrir y mejorar su conocimiento en aprender a programar con JavaScript, HTML Y CSS. Como trabajos futuros, se plantean revisar otras herramientas enfocadas en la programación, basados con la técnica de la gamificación que permitan una mejor aproximación al mundo de la programación de manera divertida.

Bibliografía

- Astudillo, G. J., Bast, S. G. y Willging, P. A. (2016). Enfoque basado en gamificación para el aprendizaje de un lenguaje de programación. *Virtualidad, Educación y Ciencia*, 7(12), 125-142.
- Educativa. (2013). Gamificación: el aprendizaje divertido. <https://www.educativa.com/blog-articulos/gamificacion-el-aprendizaje-divertido/>
- Estévez, J. (2020). Aprende a programar con esta app gratuita de Google. El Grupo Informático. <https://www.elgrupoinformatico.com/noticias/grasshopper-app-google-para-aprender-programar-t77385.html>
- Instituto Nacional de Tecnologías Educativas y Formación del Profesorado [INTEF]. (2018). GrassHopper. https://code.intef.es/prop_didacticas/grasshopper/
- Mora, M. J. C., Murillo, M. G. E., Murillo, R. D. L. Á. B., & Moyano, M. Y. C. (2022). La Gamificación como herramienta metodológica en la enseñanza. *Polo del Conocimiento: Revista científico-profesional*, 7(1), 43. <https://dialnet.unirioja.es/servlet/articulo?codigo=8331458>
- Noguera, T. (2018). Google quiere enseñarte a programar con Grasshopper, una app para iOS y Android. Xatakamovil. Recuperado de <https://www.xatakamovil.com/aplicaciones/google-quiere-ensenarte-a-programar-con-grasshopper-una-app-para-ios-y-android>

Queiruga, C. A., Fava, L. A., Gómez, N. S., Kimura, I., & Brown Bartneche, M. (2014). El juego como estrategia didáctica para acercar la programación a la escuela secundaria. In XVI Workshop de Investigadores en Ciencias de la Computación. <https://sedici.unlp.edu.ar/handle/10915/41365>

Rasgado, G. J. (2018). La programación como fuente motivadora para la construcción del conocimiento y el desarrollo de habilidades del pensamiento. RIESED-Revista Internacional De Estudios Sobre Sistemas Educativos, 2(8), 269-278. <http://www.riesed.org/index.php/RIESED/article/view/114>

Revelo, O., Alberto, C. y Alejandro, J. (2017). La gamificación como estrategia didáctica para la enseñanza/aprendizaje de la programación: un mapeo sistemático de literatura. Lámpsakos, 19, 31–46. <https://www.redalyc.org/jatsRepo/6139/613964506004/html/index.html>

Tadeu, P. (2020). La competencia científico-tecnológica en la formación del futuro docente: algunos aspectos de la autopercepción en respeto a la integración de las TIC en el aula. Educatio Siglo XXI, 38(3 Nov-Feb), 37-54. <https://revistas.um.es/educatio/article/view/413821>

Palma Suárez, C. A. y Sarmiento Porras, R. E. (2015). Estado del arte sobre experiencias de enseñanza de programación a niños y jóvenes para el mejoramiento de las competencias matemáticas en primaria. Revista mexicana de investigación educativa, 20(65), 607-641. https://www.scielo.org.mx/scielo.php?pid=S1405-66662015000200013&script=sci_arttext

Pulido, V. (2015). El uso de herramientas de programación gráfica en alumnos de CFGM en sistemas microinformáticos y redes. Unir.net. <https://doi.org/https://reunir.unir.net/handle/123456789/2952>

González, J. (2016). Gamificación: hagamos que aprender sea divertido. <https://academica-e.unavarra.es/bitstream/handle/2454/21328/TFM15-mpes-%20ege-gonzalez-68030.pdf?sequence=1&isAllowed=y>

Google LLC. (s.f.). Grasshopper. <https://learn.grasshopper.app/project/fundamentals>

Gutiérrez, A., Pinedo, R. y Gil, C. (2022). Competencias TIC y mediáticas del profesorado. Convergencia hacia un modelo integrado AMI-TIC. Comunicar: Revista Científica de Comunicación y Educación (70) 21-33. <https://dialnet.unirioja.es/servlet/articulo?codigo=8198766>

Capítulo 12

SoloLearn: plataforma móvil para aprender a programar en varios lenguajes

*Michay Caraguay Gloria Cecibel, Alvarado Montesinos Luis Gabriel
Universidad Nacional de Loja.*

luis.g.alvarado@unl.edu.ec - cecibel.michay@unl.edu.ec

Introducción

Hoy en día, la tecnología se define como la aplicación práctica de conocimientos y habilidades con el objetivo de facilitar los problemas de la sociedad hasta lograr satisfacerlas en ámbitos concretos. Estos avances comprenden en dispositivos y herramientas que permiten realizar tareas de manera ubicua, es decir, en cualquier lugar y momento. Estos aparatos se caracterizan por su tamaño pequeño, capacidades de procesamiento adecuadas y la habilidad de conectarse a redes de manera permanente o inminente para una función. Su versatilidad les posibilita ejecutar una amplia gama de funciones.

En este contexto de innovación constante, los teléfonos móviles han experimentado una notable transformación, evolucionando de simples dispositivos de comunicación a plataformas multifuncionales que ofrecen una variedad de aplicaciones preinstaladas, como mensajería, registro de llamadas, agenda, calculadora, juegos, entre otros. Por ende, el propósito de esta investigación es demostrar que la programación es posible en dispositivos, proporcionando un aporte de conocimiento sobre cómo se puede programar en móviles, siendo esto posible con la ayuda de aplicaciones específicas como SoloLearn.

Introducción a las tecnologías móviles

Los dispositivos móviles hoy en día son una parte muy esencial en nuestra vida diaria, utilizados no solo como fuentes de información sino también como herramientas versátiles para una amplia gama de funciones. Al integrarse con equipos de profesionales capacitados potencia la eficiencia y productividad en una variedad de organizaciones; estas tecnologías facilitan la recolección de datos minimizando el tiempo y el esfuerzo requerido, lo que se traduce en una significativa reducción de costos (Hernández, 2006). En consecuencia, los dispositivos son poderosas que, al estar en manos adecuadas, pueden desbloquear un potencial limitado hacia la innovación y crecimiento.

Cabe resaltar que los dispositivos móviles generalmente incluyen una “tienda de aplicaciones” donde podemos instalar cualquiera de una amplia variedad de aplicaciones útiles tanto para actividades cotidianas o fines educativos. Así mismo, como podemos instalar aplicaciones, también es posible crear aplicaciones mediante programación, usando herramientas de desarrollo que emplean distintos lenguajes de programación. Bertone et al. (2020) resalta que el mobile learning (mLearning) se puede incorporar como un complemento al aprendizaje formal, ampliando la oferta educativa y la modalidad. Lo importante es fomentar la innovación en el proceso educativo en su totalidad, más allá del uso de dispositivos tecnológicos.

Clasificación de los dispositivos móviles

La diversidad de dispositivos móviles en el mercado es amplia, abarcando distintos tamaños, tecnologías y capacidades. Critza (2016) señala que, cada uno está diseñado con un propósito y funcionalidad específicos, en general, comparten similitudes en sus funciones básicas. Sin embargo, es su rendimiento lo que a menudo los distingue unos de

otros; el aprovechamiento óptimo de estos dispositivos depende en gran medida del usuario, porque no todos utilizan al máximo de su potencial. Por esta razón, la Tabla 12.1 detalla cada uno de estos dispositivos según su propósito inicial.

Tabla 12.1. Clasificación de los dispositivos móviles según su propósito.

Dispositivos	Descripción
Dispositivos de trabajo	Estos dispositivos están diseñados con la capacidad de manejar una amplia gama de información, resultan idóneos en la realización de tareas generales. Los usuarios, como los estudiantes, pueden aprovechar estos dispositivos para una variedad de necesidades, desde la realización de informes y tareas escolares hasta el entretenimiento y entretenimiento, etc.
Dispositivos de entretenimiento	Tienen el principal propósito de entretenimiento (especialmente por medio de videojuegos), pero tienen como objetivo apoyar fácilmente el uso de otros medios de entretenimiento, como el escuchar música o incluso pueden tener servicio de comunicaciones.
Dispositivos de control e información	Se centran en ofrecer una experiencia e información específica para el usuario, y si tiene funciones adicionales, los elementos son muy secundarios.

Fuente: Adaptado a partir de Critza (2016).

¿Por qué son necesarias las tecnologías móviles en la educación?

La tecnología móvil se ha integrado en numerosos sectores, destacando su creciente influencia en la educación. Critza (2016) describe que esta tecnología facilita la conexión a reuniones, la realización de trabajos y consultas académicas. Una implementación efectiva puede potenciar un aprendizaje más significativo, aprovechando la capacidad de adaptación rápida de los estudiantes a herramientas familiares. Además,

al ofrecer métodos de estudio innovadores, la tecnología móvil rompe con la rutina tradicional; somos conscientes que la tecnología tiene sus pros y contras, pero tratamos de sacarles el mayor provecho en cuanto al aprendizaje nos referimos.

Además, la búsqueda de información y la portabilidad son aspectos que no presentan problemas al momento de usar esta tecnología, disponemos toda la información al alcance de la mano con estos dispositivos, a diferencia de los ordenadores, los cuales podemos utilizar en cualquier lugar del mundo (Avendaño, 2018). Por otro lado, es necesario señalar que estos dispositivos continúan avanzando, mejorando su eficiencia y, en algunos casos superando a los ordenadores en términos de rendimiento.

De esta manera Pulido et al. (2016) menciona que el sistema educativo se encuentra inmerso en un proceso de cambios, enmarcados en el conjunto de transformaciones sociales propiciadas por la innovación tecnológica, pero sobre todo por el desarrollo de las Tecnologías de la Información y de la Comunicación (TIC). Estos cambios están relacionados con las transformaciones en las relaciones sociales y con una nueva concepción de la interacción entre tecnología y sociedad, que a su vez impacta en la relación entre tecnología y educación.

En consecuencia, es necesario ir incorporando progresivamente las tecnologías móviles dentro de sus programas de enseñanza actuales. Esta inclusión constante no solo facilitará que los estudiantes desarrollen habilidades digitales cruciales, sino que también asegurará que dichas competencias se conviertan en un componente integral de su perfil profesional.

Eficiencia en el aprendizaje con tecnología

El avance de la tecnología móvil ha generado en la actualidad una mejora en la educación, brindando apoyo a los docentes en sus prácticas de enseñanza. Según Zamora (2020), este desarrollo tecnológico resulta clave en la superación de los métodos de enseñanza tradicionales. El empleo de estas tecnologías representa un avance significativo al superar las barreras de tiempo y espacio que anteriormente limitaban el proceso educativo.

Dentro de este marco, la tecnología ha ayudado mucho en este aspecto, porque hoy en día los alumnos tienen acceso al internet muchos sitios públicos; cada vez son más las aulas repletas de alumnos con laptops y tablets. La búsqueda de información se hace casi instantánea, sin necesidad de leer un libro entero. La enseñanza ya no se limita a un espacio físico; las clases virtuales son posibles gracias a una simple conexión a Internet (TECNO2015, 2015). Los dispositivos móviles, cuando se usan adecuadamente y en conformidad con las normativas de cada institución educativa, pueden ser herramientas poderosas para mejorar la calidad de la enseñanza.

Por ello, los autores Chacaguasay y Suárez (2017) enfatizan la importancia del desarrollo de habilidades digitales y tecnológicas en estudiantes, así como la integración efectiva de herramientas digitales en el entorno educativo con la finalidad de promover estas competencias esenciales. La utilización de estas herramientas tecnológicas como apoyo en la educación, especialmente en el contexto de colaboración, ha demostrado ser un factor que optimiza el proceso de enseñanza-aprendizaje, permitiendo un acceso al conocimiento flexible, en cualquier momento y lugar.

Inclusión de la tecnología móvil en el aprendizaje

La inclusión de tecnología móvil en el aprendizaje, conocida como mLearning, es un proceso educativo que utiliza redes inalámbricas para facilitar una interacción inmediata entre docentes y alumnos, enfoque que no solo promueve el desarrollo de competencias digitales, sino que también fortalece las habilidades en el manejo de las TIC según Santiago et al. (2015). Con un crecimiento significativo en su adopción dentro de entornos educativos, las tecnologías móviles están impulsando metodologías más dinámicas, flexibles y accesibles, como señalan González y Sosa (2021), pero al mismo tiempo la implementación del mLearning requiere una estrategia bien definida para asegurar su integración efectiva en el currículo y en la práctica pedagógica.

En este aspecto, el mLearning se perfila como una metodología con un futuro prometedor y adaptable, capaz de completarse con diversas técnicas pedagógicas. Según Aznar et al. (2019), los maestros en formación reconocen las posibilidades que otorgan los dispositivos en el aprendizaje, favoreciendo así una gran variedad de elementos para enriquecer el aprendizaje.

Con respecto, la incorporación de dispositivos móviles representa un beneficio significativo, especialmente al considerar que los estudiantes dedican gran parte de su tiempo al uso del móvil, tiempo que puede ser empleado en aprendizaje (López, 2018). Por ejemplo, la enseñanza de la programación puede realizarse mediante aplicaciones móviles diseñadas para codificar directamente desde el teléfono, en conjunto con una excelente guía los estudiantes podrán comprender fácilmente este ámbito (Falconí, 2017). Otra forma de incluir está a través de videos y actividades en clases, en conjunto con el docente.

Implementación de la programación móvil con SoloLearn

SoloLearn destaca como una de las aplicaciones más utilizadas y efectivas para aprender programación desde cero en dispositivos móviles. Esta plataforma ofrece una experiencia de aprendizaje dinámica y creativa que abarca tanto los fundamentos como los aspectos más avanzados de diversos lenguajes de programación (SoloLearn, 2021). Su relevancia en el ámbito educativo se evidencia en su amplia selección de cursos gratuitos, disponibles para cualquier usuario interesado.

Aprende a programar con SoloLearn

Embarcarse en el aprendizaje de la programación con SoloLearn es una experiencia emocionante y accesible, disponible en múltiples plataformas como iOS, Android y Windows Phone permitiendo a los usuarios aprender en el dispositivo de su elección. De acuerdo con Moncada (2017), esta herramienta no solo introduce a los usuarios a los fundamentos de la programación estructurada, orientada a objetos y basada en eventos en distintos lenguajes, sino que también se distingue por enseñar la estructura esencial de estos lenguajes. SoloLearn proporciona una base sólida que muchas otras aplicaciones omiten, asegurando que los estudiantes no solo memoricen, sino que realmente entiendan los conceptos clave de la programación.

Por medio de esta App, se puede aprender al ritmo que uno desee, en cada curso hay preguntas y ejercicios específicos que ayudan a comprender y aprender programación. A medida que se avanza, se puede realizar ejercicios prácticos que refuerzan el aprendizaje. Por ejemplo, en la Figura 12.1, se puede observar un ejemplo de declaraciones de variables, ofreciendo una guía paso a paso para su correcta utilización. Si pulsamos la pantalla podremos realizar varios intentos con esas variables; en esta misma instancia, Code Coach, nos proporciona una

compresión más profunda de los cursos de programación, accediendo a realizar ejercicios basados en intentos previos y avanzar en el curso tras completar la práctica.



The screenshot shows a mobile application interface for learning Java. At the top, the title 'Variables' is displayed in bold black font. Below it, a white box contains the text 'Ejemplos de declaraciones de variables:' in a smaller black font. Underneath this, there is a code editor window with a dark background and light-colored text. The code is a Java class named 'MyClass' with a main method containing variable declarations. A small blue 'JAVA' badge is located in the top right corner of the code window. At the bottom of the code window, the text 'Pulsa para intentarlo |' is visible, indicating an interactive feature. The overall interface has a clean, modern design with a light gray header bar.

```
class MyClass {  
    public static void main(String[] args) {  
        String name = "David";  
        int age = 42;  
        double score = 15.9;  
        char group = 'Z';  
    }  
}
```

Figura 12.1. Ejemplo de declaraciones de variables. Fuente: Adaptado a partir de SoloLearn (2021).

Métodos de enseñanza

Los métodos de enseñanza son componentes dinámicos que implican interacciones entre docentes y estudiantes, al respecto Moncada (2017) señala que SoloLearn no solo enseña los conceptos básicos, sino que también ofrece pruebas tipo examen para avanzar al siguiente módulo y como característica muy particular, esta herramienta integra un compilador, sin afectar su funcionamiento, lo cual es poco común. SoloLearn es ampliamente utilizada debido a su efectividad: cada módulo comienza con conceptos fundamentales seguidos de una evaluación breve, si no se supera la prueba, se redirige al usuario a los conceptos básicos para reforzar el aprendizaje. El compilador también permite a los usuarios aclarar dudas y experimentar con diferentes variables.

Cursos disponibles



Figura 12.2. Cursos que ofrece SoloLearn. Fuente: Adaptado a partir de SoloLearn (2021).

SoloLearn como plataforma ofrece una amplia gama de cursos sobre programación y desarrollador web como PHP, JavaScript, CSS, HTML, entre otros, los mismos se pueden explorar en la Figura 12.2. Además contempla cursos ideales para principiantes que necesitan comenzar desde cero o usuarios más avanzados que buscan profundizar sus conocimientos en programación, en cualquier caso todos los contenidos que ofrece SoloLearn son completamente gratis y sin ninguna limitante.

Aprendizaje constante

Una vez seleccionado un curso, se presentarán preguntas, ejercicios y prácticas que marcarán el comienzo de la formación elegida. Como se ha indicado anteriormente, es el usuario quien gestiona su propio avance, decidiendo cuánto desea progresar o aprender cada día. Es importante destacar la existencia de una barra de progreso que indica en qué punto del curso nos encontramos; esto sirve como un valioso indicador para monitorear nuestro desarrollo, ya que la App no cuenta con un sistema de calificación, pero sí con un sistema de seguimiento.

El contenido que ofrece SoloLearn está en español; este se actualiza constantemente, asegurando que los usuarios tengan acceso a tareas recientes y relevantes acorde al curso elegido. Al tratarse de una App comunitaria, siempre podremos encontrar contenidos y desafíos de otros programadores para poder comprobar nuestro aprendizaje.

Comunidad de SoloLearn

SoloLearn se destaca en el aprendizaje digital al promover la adquisición de conocimientos en programación y fomentar una comunidad activa y colaborativa, sino que también fomenta una comunidad activa y participativa. A través de su foro de discusión enriquece la experiencia de aprendizaje permitiendo a los usuarios plantear dudas y preguntas, las cuales son resueltas por otros miembros de la comunidad de programadores. Además, la plataforma facilita crear eventos de desafío contra otros programadores y completar desafíos de código con distintos niveles de dificultad, lo que enriquece la experiencia de aprendizaje (SoloLearn, 2021). La Figura 12.3 se puede visualizar de una mejor manera estas características, mostrando la naturaleza colaborativa y el soporte comunitario que ofrece.



Figura 12.3. Comunidad de SoloLearn. Fuente: Adaptado a partir de SoloLearn (2021).

Código desarrollado en SoloLearn

En la hoja HTML de la Figura 12.4 incluye elementos como <cont> que identifica la caja de imágenes. Se usa el método <draggable> para poder mover cualquier tipo de elemento sea arrastrable. Además, se han añadido imágenes del almacenamiento del dispositivo, y en la línea 23 se utiliza la etiqueta con el propósito de referenciar y mostrar una imagen descargada previamente.

	HTML	CSS	JS	SALIDA
1	<!DOCTYPE HTML>			
2	<html>			
3	<head>			
4	<title>Arrastrar y soltar</title>			
5	</head>			
6				
7	<body>			
8	<h1>Arrastrar y soltar</h1>			
9	<h2>Pon los elementos del contenedor de la izquierda en el de la derecha</h2>			
10	<div id="cont0">			
11				
12				
13	<div id="cont1" >			
14				
15	<div><p draggable="true" ondragstart="evdragstart(event,this)" id="e1">Elemento arrastrable</p></div>			
16	<div><p draggable="true" ondragstart="evdragstart(event,this)" id="e2">Google:			
17	ir a Google			
18	</p></div>			
19	<div><ul draggable="true" ondragstart="evdragstart(event,this)" id="e3">			
20	Primer elemento			
21	Segundo elemento			
22	</div>			
23	` para aplicar nuestros estilos definidos. Además, propiedades `<width>`, `<height>`, `<background-color>`, etc; son fundamentales con la finalidad de determinar el tamaño y color de las imágenes.			

HTML	CSS	JS	SALIDA
1	* { margin: 0 auto; padding: 0;		
2	}		
3			
4	h1 { font:bold 1.5em helvetica; text-align:		
5	center; color: navy; margin: 0.5em;		
6	}		
7			
8	h2 { font: bold 1em helvetica; text-align:		
9	center; margin: 0.5em;		
10	}		
11			
12	#cont0 { width: 80%; height: 300px;		
13	}		
14			
15	#cont1 { width: 50%; height: 300px;		
16	background-color:#ccffff; float: left;		
17	overflow: auto;		
18	}		
19			
20	#cont2 { width: 50%; height: 300px;		
21	background-color:#ffccff; float: left;		
22	overflow: auto;		
23	}		

Figura 12.5. Hoja de CSS. Fuente: Elaboración propia.

Cabe resaltar que existen algunos atributos que podemos usar en los elementos del HTML para configurar el proceso de una operación arrastrar y soltar, pero puede ser hecho desde código Javascript. En la Figura 12.6, se muestran algunas funciones como dragstart(), en sí este evento se inicia cuando el proceso de arrastre comienza. Otra función que se puede observar es la de dragover() en cambio este se da durante la operación de arrastre. Así mismo, la función drop() comienza cuando se suelta el elemento que se estaba arrastrando; estas 3 funciones son imprescindibles para su funcionamiento. Finalmente, podemos observar el objeto dataTransfer(), siendo este el que contendrá la información de las operaciones de arrastrar y soltar.

HTML	CSS	JS	SALIDA
		<pre> 1 function evdragstart(ev,el) { 2 cont1=el.parentNode; 3 4 ev.dataTransfer.setData("text",ev.target.id); 5 6 function evdragover (ev) { 7 ev.preventDefault(); 8 } 9 10 function evdrop(ev,el) { 11 ev.stopPropagation(); 12 ev.preventDefault(); 13 var data=ev.dataTransfer.getData("text"); 14 15 mielem=ev.target.appendChild(document.getElementById(data)); 16 cont1.appendChild(mielem); 17 mielem2=mielem.cloneNode(true); 18 mielem2.setAttribute("draggable","false"); 19 el.appendChild(mielem2); 20 } </pre>	

Figura 12.6. Hoja de JavaScript. Fuente: Elaboración propia.

La interactividad enriquece la navegación web al involucrar al usuario como actor principal en lugar de ser simplemente un observador. Esto se evidencia en la Figura 12.7, donde se muestra el resultado de programar interacciones de arrastrar y soltar en las hojas de estilo, otorgando mover texto e imágenes libremente a diferentes áreas de la interfaz.



Figura 12.7. Resultado. Fuente: Elaboración propia.

Mediante la exploración y realización de actividades, SoloLearn se destaca como una plataforma accesible y gratuita que abarca una amplia gama de lenguajes y conceptos de programación. Esta aplicación no solo ofrece una variedad de lenguajes y conceptos de programación, sino que también fomenta un ambiente de aprendizaje interactivo y colaborativo. Con recursos valiosos como foros de discusión, desafíos de programación y un compilador integrado, SoloLearn permite a los usuarios aprender a su propio ritmo, adaptándose a sus necesidades individuales y haciendo que la programación sea una habilidad al alcance de todos.

Reflexiones

La incorporación de dispositivos móviles en la educación no es solo una tendencia, sino una evolución lógica y necesaria en la sociedad moderna. No se trata sólo de integrar la tecnología en la educación, sino de reconocer su papel esencial en nuestras vidas. Los dispositivos móviles, se han convertido en herramientas indispensables, ofreciendo una ventaja significativa al ampliar el acceso a la educación para estudiantes y docentes en cualquier contexto. No obstante, la adopción de estas tecnologías implica un proceso de adaptación constante, donde aprender, desaprender y reaprender se convierte en esencial al navegar por entornos emergentes.

Finalmente, a través de la aplicación SoloLearn, podemos aprender a programar desde cero en dispositivos móviles, considerando que tiene una gran variedad de cursos completamente sin costo, lecciones interactivas y una comunidad siempre dispuesta a ayudar. La misma mantiene un método de enseñanza adaptable que permite que cada usuario personalice su experiencia educativa, eligiendo libremente su horario y ritmo de estudio. Esta flexibilidad, no solo promueve la inclusión y la accesibilidad, sino que también capacita a los estudiantes en convertirse en aprendices de por vida, preparados para enfrentar los desafíos de la era digital.

Bibliografía

Avendaño, C. (2018). Introducción a las Tecnologías y Herramientas Móviles. Scribd. <https://bit.ly/3hBjN6J>

Aznar, I., Cáceres, M., Trujillo, J. y Romero, J. (2019). Mobile learning y tecnologías móviles emergentes en Educación Infantil: percepciones de los maestros en formación. Revista Espacios, 40(05). <https://www.revistaespacios.com/a19v40n05/19400514.html>

Bertone, R., Filippi, J., Lafuente, G., Ballesteros, C., Lafuente, G., Pérez, D. y Mansilla, A. (2020). Incorporación de la tecnología móvil en el proceso educativo. In XXII Workshop de Investigadores en Ciencias de la Computación (WICC 2020, El Calafate, Santa Cruz). Repositorio Institucional de la UNLP. <https://bit.ly/3yNuydn>

Chacaguasay, R. y Suárez, Jhon. (2017). Los dispositivos móviles en el proceso de enseñanza aprendizaje. [Tesis de grado, Universidad de Guayaquil, Guayaquil]. Repositorio de la Universidad de Guayaquil. <https://bit.ly/3xTZU1y>

Critza. (2016). Clasificación de las Aplicaciones Móviles. <http://www.critza.com/post/clasificacion-de-las-aplicaciones-moviles>

Falconí, A. (2017). Inclusión de la tecnología móvil de información y comunicación educativa como estrategia pedagógica. Revista Universidad y Sociedad, 9(4), 82-89.<https://bit.ly/361vdWR>

González, A. y Sosa, M. (2021). Aspectos pedagógicos, tecnológicos y de interacción social del aprendizaje móvil: Revisión Sistemática de Literatura. Educatio Siglo XXI, 39(1), 257-280. <https://revistas.um.es/educatio/article/view/469271>

- Hernández, N. (2006). Programación De Aplicaciones Para Dispositivos Móviles. Universidad De El Salvador, Ciudad Universitaria. [Tesis de grado, Universidad de El Salvador]. <https://core.ac.uk/download/pdf/11228513.pdf>
- López, J. (2018). El uso de las aplicaciones móviles en el sector educativo. Revista de ciencia y tecnología Milenio. <https://bit.ly/3heSOiu>
- Moncada, M. (2017). SoloLearn, la mejor app para aprender programación. Wiki Estudiantes. Recuperado de <https://bit.ly/3xN6V3w>
- Pulido, D., Nájar, O. y Guesguán, L. (2016). Vivamos la innovación de la inclusión de dispositivos móviles en la educación. Praxis & Saber, 7(14), 115-140. <https://bit.ly/3kz0uy3>
- Santiago, R., Trabaldo, S., Kamijo, M. y Fernández, A. (2015). Mobile Learning-nuevas realidades en el aula. Barcelona. Editorial Oceano. <https://bit.ly/3At0ftZ>
- SoloLearn. (2021). SoloLearn: Learn to Code. <https://www.sololearn.com/>
- TECNO2015. (2015). Tecnología: ¿un beneficio o una adicción? La dependencia tecnológica. <https://n9.cl/ufg5f>
- Zamora, R. (2020). Las ventajas de la utilización de dispositivos móviles en el proceso de aprendizaje en la educación básica. Revista de Ciencias Humanísticas y Sociales (ReHuSo), 5(1), 82-91. <https://www.redalyc.org/pdf/6731/673171024008.pdf>

Capítulo 13

Experiencia de Enseñanza de Programación con Realidad Aumentada

Michay Caraguay Gloria Cecibel - Coello Ruiz Paola Lizzet - Jiménez Gaona Rosela del Cisne- Quezada Vera Lida Andreina.

Universidad Nacional de Loja.

cecibel.michay@unl.edu.ec, paola.coello@unl.edu.ec, rosela.jimenez@unl.edu.ec, laquezadav@unl.edu.ec

Introducción

La integración de la realidad aumentada en la enseñanza de programación representa un avance significativo en la educación tecnológica. La realidad aumentada, al combinar elementos virtuales con el mundo real, ofrece una experiencia inmersiva y dinámica que potencia el aprendizaje de los discentes. En este contexto, la aplicación de la realidad aumentada en la enseñanza de programación no solo facilita la comprensión de conceptos abstractos, sino que también promueve la creatividad, la interactividad y el compromiso activo de los estudiantes en su proceso de aprendizaje. En el presente capítulo, se explorará cómo la realidad aumentada enriquece la enseñanza de la programación, brindando nuevas herramientas y oportunidades para la formación de habilidades tecnológicas y el desarrollo de proyectos innovadores.

Enseñanza del lenguaje de programación en la educación

La tecnología ha transformado el panorama educativo, y la informática se ha convertido en un pilar fundamental del aprendizaje moderno. Las instituciones educativas, conscientes de este cambio, implementan

nuevos métodos de enseñanza y aprendizaje que integran la informática como herramienta esencial para el futuro (Rosas et al., 2018). Es por ello, que los docentes de informática para estar a la par de esta transformación, se capacitan constantemente con la finalidad de ampliar y mejorar sus habilidades en el uso de las nuevas tecnologías. Este proceso les permite dominar las funciones de los programas informáticos y utilizarlos de forma eficaz en el aula.

La enseñanza de lenguajes de programación comienza desde temprana edad, incluso desde la educación primaria, a través de la asignatura de Ciencias Informáticas. Esta temprana introducción permite a los estudiantes familiarizarse con los conceptos básicos de la programación y desarrollar habilidades, siendo así, que los estudiantes al llegar a bachillerato han logrado adquirir un aprendizaje sólido en la programación lo que les permitirá crear sus propios programas (Fracchia et al., 2016). Es por ello, que la Ciencias de la Computación, orienta al desarrollo de una serie de habilidades que permitan dar soluciones generales aplicables a distintos problemas del mundo real, resaltando la creatividad y la innovación.

En un mundo cada vez más digital, los sistemas educativos tienen la responsabilidad de preparar a los jóvenes para el futuro. Para ello, deben dominar un nuevo lenguaje de programación para evitar el analfabetismo digital. Por tanto, la escuela debe asumir estos desafíos que se presentan y complementar la formación de alfabetismo lingüístico y numérico con la alfabetización digital, según lo afirma Llorens (2015) donde destaca la importancia de que los estudiantes no sientan miedo a aprender lenguaje algorítmico, sino que vean como un mecanismo de aprendizaje que puede facilitar el acceso a un futuro prometedor.

Programar es más que escribir código, no se limita a la simple tarea de escribir un programa. Según Rosas et al. (2018) lo definen como el “uso

simultáneo de la creatividad, un conjunto de conocimientos técnicos asociados y la aptitud de trabajar con abstracciones, tanto simbólicas como mentales”. Es un proceso complejo que exige la interacción armoniosa de la creatividad, el conocimiento técnico y la capacidad de trabajar con abstracciones.

En conexión a lo abordado con anterioridad acerca del uso de la programación, los autores Dapozo et al. (2019) van más allá y señalan que la programación involucra:

- **Entender un problema:** Descifrar su naturaleza, sus componentes y sus relaciones.
- **Plantear soluciones efectivas:** Diseñar estrategias que resuelvan el problema de forma eficiente.
- **Manejar lenguajes para expresar una solución:** Dominar lenguajes de programación para traducir las ideas en código.g
- **Utilizar herramientas que entiendan esos lenguajes:** Emplear software y plataformas para ejecutar el código.
- **Probar que la solución sea válida:** Evaluar si la solución funciona correctamente y cumple con los objetivos.
- **Justificar las decisiones tomadas:** Explicar las razones detrás de las elecciones tomadas durante el proceso.

El aprendizaje de lenguajes de programación de acuerdo a Cobo (2019), es “fundamental desarrollar habilidades a prueba de futuro, no limitadas a ciertas herramientas, instrumentos o metodologías y adaptables a diferentes contextos y actualizables durante el aprendizaje con otros” (pág. 30), debido a que la tecnología avanza a pasos agigantados por ende es importante la educación asuma este rol ya que permite desarrollar en los estudiantes las competencias necesarias para el ámbito profesional.

Beneficios de la enseñanza de lenguajes de programación

La enseñanza de lenguajes de programación en la educación se perfila como una inversión crucial en el futuro de los estudiantes. Díaz et al. (2018), indican que esta práctica no solo se limita a la adquisición de habilidades para escribir código, sino que va más allá al fomentar el desarrollo de capacidades fundamentales para el siglo XXI, debido a que la enseñanza de la programación tiene, entre sus objetivos esenciales, la formación de:

- **Desarrollo de habilidades:** La programación ayuda a desarrollar habilidades como el pensamiento computacional, la lógica, la resolución de problemas, la creatividad, la comunicación y el trabajo en equipo.
- **Preparación para el futuro:** En un mundo cada vez más digital, las habilidades de programación son cada vez más demandadas en el mercado laboral.
- **Empoderamiento:** La programación permite a los estudiantes crear sus propios proyectos y soluciones, lo que les da un sentido de empoderamiento y control sobre su aprendizaje.
- **Mayor acceso a la información:** La programación facilita el acceso a la información y al conocimiento, ya que permite a los estudiantes crear sus propias herramientas y aplicaciones.

Realidad Aumentada en la educación

La visualización de la realidad aumentada, que combina el mundo real con elementos virtuales, debe ser coherente y consistente. Por tal razón, Aguilar et al. (2023) manifiestan que “la realidad aumentada es una tecnología que agrega en tiempo real una capa adicional de información virtual a la percepción del mundo real”, lo que proporciona a los usuarios

la idea de que los objetos virtuales y reales existen juntos en el mismo espacio, al mismo tiempo.

La implementación de realidad aumentada es un lenguaje de programación útil, en actividades educativas, como el aprendizaje multimedia, la teoría del aprendizaje basado en experiencias y la teoría de la visión animada. Además, Salazar (2015) señala que la Realidad Aumentada (RA) es una tecnología que complementa la percepción e interacción con el mundo real y permite al usuario estar en un entorno real aumentado, con información adicional generada por la computadora, posibilitando el desarrollo de aplicaciones interactivas que combinan la realidad con información sintética tal como imágenes 3D, sonidos, videos, textos, sensaciones táctiles en tiempo real desde el punto de vista del usuario. Por ello, la RA es sencilla y manejable ya que facilita la aplicación y utilización para realizar proyectos.

Una experiencia en la que se utiliza la RA para la realización de actividades educativas en el contexto de un laboratorio de ciencias, y se analizan los resultados en relación a las habilidades y actitudes de los alumnos en el desarrollo de dichas actividades (Salazar et al., 2016). Los resultados muestran que esta tecnología impacta positivamente en los estudiantes, ya que los ayuda a vivenciar experimentos que no podrían suceder en un laboratorio real.

La necesidad de adaptar los procesos de enseñanza y aprendizaje a los tiempos actuales es cada vez más necesario. Es aquí donde la RA se presenta como una herramienta poderosa para la innovación educativa. Más allá de una simple tecnología móvil, tiene el potencial de transformar la forma en que enseñamos y aprendemos (Aguilar et al., 2023). Su capacidad para crear experiencias inmersivas y contextualizadas permite al estudiante ser el protagonista de su propio aprendizaje, interactuando con el mundo real de una manera completamente nueva

Esta tecnología en el ámbito educativo permite crear experiencias de aprendizaje innovadoras y atractivas que cumplen con las expectativas de los estudiantes nativos digitales. En este contexto, RA emerge como una herramienta capaz de acaparar la “atención, interés, iniciativa y autonomía” (Marín et al., 2018 citado en López et al., 2020) del estudiante facilitando el desarrollo de contenidos educativos y también abre un abanico de posibilidades para la creación de estrategias didácticas innovadoras. Esta tecnología se presenta como un aliado invaluable para la educación del futuro ya que preparar a los estudiantes a los desafíos del mañana.

Beneficios de la RA

El uso de la RA está creciendo en importancia en el ámbito educativo, especialmente en disciplinas prácticas, según Pimentel et al. (2023), esto es crucial ya que permite a los usuarios proyectar objetos y contenido en tres dimensiones, mejorando así la experiencia de aprendizaje. Así mismo, como estrategia de aprendizaje fomenta la interacción con lo virtual, permiten obtener mejores resultados en la formación de los estudiantes dentro del proceso de aprendizaje; a la vez se sienten motivados por el uso de nuevas tecnologías.

En este sentido, la integración de la realidad aumentada en la enseñanza educativa ha revelado diversos beneficios; entre ellos se encuentra el aumento del interés de los estudiantes en su propio proceso de aprendizaje, el fortalecimiento del aprendizaje mediante el uso de juegos y la generación de interacción en tiempo real (Mesía, 2016). Además, permite la conexión entre el conocimiento intelectual y la experiencia física, lo que mejora la asimilación y comprensión de los contenidos educativo.

Herramienta Metaverso

El término “metaverso” o “meta-universo” se originó en la novela “Snow Crash”, publicada en 1992 por Neal Stephenson, y se emplea regularmente para describir una visión de trabajo en espacios tridimensionales (3D). Este concepto se refiere a un mundo virtual ficticio o un espacio virtual colectivo y compartido. Según Basogain et al. (2007), el metaverso se crea a menudo mediante la convergencia y la compatibilización con aspectos de la realidad externa, así mismo, Peña (2023) señala que el término “Metaverso” es una palabra simple y fácil de recordar, pero encierra un enigma, ya que pocos conocen su significado.

Dentro de este orden de ideas en relación a esta herramienta, para Sánchez (2022), el término “metaverso” hace referencia a un mundo virtual paralelo en el que las personas interactúan a través de avatares, representaciones digitales de sí mismos. Un “avatar” es la identidad que adoptamos en este entorno virtual. El metaverso se presenta como una plataforma intuitiva y accesible para aquellos que deseen explorar un mundo tridimensional (3D). Es una herramienta diseñada para facilitar la inmersión y el juego espontáneo en el universo de los videojuegos.

En el ámbito educativo, el metaverso puede transmitir ideas innovadoras y facilitar el desarrollo de habilidades, permitiendo a los estudiantes expresar sus talentos e interactuar con sus docentes y discentes de clase (Peña, 2023). Esta aplicación beneficia tanto a los estudiantes como a los educadores, al ofrecer características indispensables al proceso educativo como se detalla en la Tabla 13.1, brindando un enfoque más dinámico y participativo.

Tabla 13.1. Características de Metaverso.

Características	Descripción
Interactividad	Los usuarios pueden comunicarse entre sí y participar en interacciones dentro del metaverso.
Corporeidad	Los usuarios están representados por avatares y están sujetos a ciertas reglas y recursos dentro del entorno virtual.
Persistencia	Los usuarios están representados por avatares y están sujetos a ciertas reglas y recursos dentro del entorno virtual.

Fuente: Adaptado a partir de Sánchez (2022).

En cuanto a la creciente importancia de los dispositivos tecnológicos en la educación, Barráez (2022) alude que resulta fundamental garantizar una adecuada conectividad para perfeccionar las experiencias virtuales propias del metaverso, ya que proporciona estabilidad en la señal y permite la transmisión de audio y video en tiempo real.

Reflexiones

La enseñanza de lenguajes de programación en la educación se ha convertido en un pilar esencial del aprendizaje moderno, gracias a la transformación que la tecnología ha provocado en el panorama educativo. Las instituciones educativas, conscientes de esta evolución, se encuentran implementan métodos de enseñanza que integran la tecnología como una herramienta clave para el futuro.

En esencia, la RA representa una poderosa herramienta que transforma la enseñanza de programación en los centros educativos, ofreciendo una experiencia inmersiva y práctica para los estudiantes, debido a que la programación con realidad aumentada permite a los estudiantes desarrollar habilidades de resolución de problemas de manera espontánea y adquirir experiencia práctica para abordar situaciones del mundo real en un entorno digital.

Es así, que la enseñanza de programación con realidad aumentada no solo fomenta la creatividad y la innovación, sino que también promueve la colaboración entre docentes y estudiantes, ofreciendo un enfoque motivador e innovador en el proceso de aprendizaje. Esta experiencia práctica en la enseñanza de programación con realidad aumentada, busca formar programadores exitosos que puedan aprovechar al máximo su potencial en un mundo cada vez más digitalizado.

Bibliografía

- Aguilar, F., Flores, J. A., Pacheco, D. y Caldera, J. (2023). Perspectiva tecno-pedagógica de la realidad aumentada en la educación. *Investigación y Ciencia de la Universidad Autónoma de Aguascalientes*, 31(90). <https://doi.org/10.33064/iycuaa2023904252>
- Barráez, D. (2022). Metaversos en el Contexto de la Educación Virtual. *Revista Tecnológica-Educativa Docentes 2.0*, 13(1), 11-19. <https://doi.org/10.37843/rted.v13i1.300>
- Basogain, X., Olabe, M., Espinosa, K., Rouèche, C. y Olabe, J. (2007). Realidad Aumentada en la Educación: una tecnología emergente. Escuela Superior de Ingeniería de Bilbao, EHU. <http://bit.ly/2hpZokY>
- Cobo, C. (2019). Realidad Aumentada en la Educación: una tecnología emergente. DocPlayer. <https://www.educ.ar/recursos/152203/acepto-las-condiciones-usos-y-abusos-de-las-tecnologias-digitales-madrid-2019>
- Dapozzo, M. J., Greiner, C., Petris, G., Espíndola, M. V. y Coto, C. (2019). Enseñanza de la Programación en la Universidad. Factores que Inciden en el Buen Desempeño de los Estudiantes. *Educación en Ingeniería*, 1122-1128. Santa Fe, Argentina
- Díaz, K., Martín, M. y Muñoz, M. (2018). La enseñanza de la programación. Una experiencia en la formación. *Educación XXVII* (53), 73-91. <https://revistas.pucp.edu.pe/index.php/educacion/article/view/20307/20260>

- Fracchia, C., Kogan, P. y Amaro, S. (2016). Competir, motivar hornero aprender programación. Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología, (18), 19-29. <https://teyet-revista.info.unlp.edu.ar/TEyET/article/view/383/16>
- López, J., Pozo, S., Fuentes, A. y Romero, J. M. (2020). Eficacia del aprendizaje mediante flipped learning con realidad aumentada en la educación sanitaria escolar. Journal of Sport and Health Research. 12(1):64-79.
- Llorens, F. (2015). Dicen por ahí. . . que la nueva alfabetización pasa por la programación. Revista de Investigación em Docencia Universitaria de la Informática 8(2), 11-14. https://rua.ua.es/dspace/bitstream/10045/49092/1/2015_Llorens_ReVision.pdf
- Mesía, M. A. (2016). Realidad aumentada en el laboratorio de ciencias: ¿una herramienta para mejorar el aprendizaje? Revista Eureka sobre Enseñanza y Divulgación de las Ciencias, 13(3), 409-424.
- Peña, M. (2023). Hacia una ontología del metaverso: crítica a sus concepciones y sus características e implicaciones [Trabajo de maestría, Universidad Industrial de Santander]. <https://noesis.uis.edu.co/server/api/core/bitstreams/a4bde619-351c-4693-a963-68bd0894c3ca/content>
- Pimentel Elbert, M. J., Zambrano Mendoza, B. M., Mazzini Aguirre, K. A. y Villamar Cárdenas, M. A. (2023). Realidad virtual, realidad aumentada y realidad extendida en la educación. RECIMUNDO, 7(2), 74-88. [https://doi.org/10.26820/recimundo/7.\(2\).jun.2023.74-88](https://doi.org/10.26820/recimundo/7.(2).jun.2023.74-88)

- Rosas, R., Zúñiga, R., Fernández, M. C. y Guerrero, L. A. (2018). Pensando computacionalmente: ¿cómo, ¿cuándo y dónde? y... ¿Quiénes? en Universidad Nacional de Misiones, XIII Congreso de Tecnología en Educación y Educación en Tecnología (1a ed.). Universidad Nacional de Misiones. <https://www.fceqyn.unam.edu.ar/teyet2018/wp-content/uploads/2018/07/teyet2018-acta.pdf>
- Salazar, N. (2015). Realidad Aumentada en la Enseñanza de Conceptos Básicos de Programación [Tesis de licenciatura, Universidad Nacional de la Plata]. https://sedici.unlp.edu.ar/bitstream/handle/10915/71302/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y
- Salazar, N., Sanz, C. y Gorga, G. (2016). Experiencia de enseñanza de programación con realidad aumentada. Actas de las Jornadas sobre la Enseñanza Universitaria de la Informática (JENUI), (1), 209-216. <https://dialnet.unirioja.es/servlet/articulo?codigo=8478288>
- Sánchez, M. (2022). El metaverso: ¿la puerta a una nueva era de educación digital? Investigación En Educación Médica, 11(42), 5-8. <https://doi.org/10.22201/fm.20075057e.2022.42.22436>

Revisores Externos

Rosemary de Lourdes Samaniego Ocampo



Datos personales:

Nacionalidad: Ecuatoriana
Dirección: 214 San Martín y 10 de agosto, Machala, Ecuador
Teléfono: +593983601137
Correo electrónico: rsamaniego@utmachala.edu.ec

Formación académica:

2015 - 2019	Universidad de las Islas Baleares, España Doctora en Tecnología Educativa
2013 - 2015	Universidad de las Islas Baleares, España Maestría en Tecnología Educativa: e-learning y gestión del conocimiento
2010	Universidad de Málaga Máster universitario en Nuevas Tecnologías Aplicadas a la Educación
2001 - 2004	Universidad Técnica de Machala Maestría en Gerencia de Proyectos Educativos y Sociales, 2004 Diploma Superior en Diseño de Proyectos, 2002 Especialista en Liderazgo y Gerencia, 2001

Experiencia Laboral:

Fecha:	1996 - presente
Institución:	Universidad Técnica de Machala / Educación superior. Sector Público.
Cargos y responsabilidades:	<ul style="list-style-type: none">• Vicerrectora Académica• Decana de la Facultad de Ciencias Sociales• Subdecana de la Facultad de Ciencias Sociales• Coordinadora Académica de la Facultad de Ciencias Sociales• Coordinadora de la Carrera de Docencia en Informática. 2014• Representante del Grupo de Investigación de Tecnología Educativa• Organizadora del I, II y III Congreso de Tecnología Educativa• Participante del Proyecto de Vinculación con la Sociedad “Herramientas Tecnológicas para el aula”. 2014-2015• Capacitadora a Docentes en curso de NTIC. 2013• Jurado Calificador de la Primera Feria Distrital Juvenil de Innovación Ciencia y Tecnología. 2013• Subdecana de la Facultad de Ingeniería Civil. Enero/Agosto 2012• Organizadora del 3er. Encuentro Internacional de Tecnologías de Información y 4to. Congreso Informática UTMACH 2012.• Capacitadora a Docentes Titulares en curso de TIC. 2012• Organizadora del 2do. Encuentro Internacional de Tecnologías de Información y 3er. Congreso Informático UTMACH 2011.• Coordinadora General del curso Pre Universitario de la Facultad de Ciencias Empresariales de la Universidad Técnica de Machala. Período 2001 – 2002, 2002 - 2003.• Autora de libros, capítulos de libros, ha escrito más de 30 artículos científicos en revistas indexadas de alto impacto.

Revisores Externos

Janio Lincon Jadán-Guerrero



Datos personales:

Nacionalidad: Ecuatoriana
Dirección: Calle Río Cachabi y Av. Machala, Conjunto Akhenaton, Torre 6
Teléfono: +593996339372
Correo electrónico: jjadan@gmail.com

Formación académica:

2013 - 2016	Universidad de Costa Rica Doctor en Computación e Informática
2001 - 2003	Universidad Tecnológica Indoamérica Magíster en Administración y Marketing
1998 - 2000	Universidad de Costa Rica Magíster Scientiae en Computación e Informática
1987 - 1995	Universidad Central del Ecuador Ingeniero en Informática

Experiencia Laboral:

Fecha: 1987 - presente
Institución: Universidad Tecnológica Indoamérica / Educación superior. Sector Privado
Cargos y responsabilidades:

- Vicerrector de Investigación
- Director del Instituto de Investigación, desarrollo e Innovación
- Director del Centro de Investigación en Mecatrónica y Sistemas Interactivos-MIST
- Editor Asociado de la Revista Cienciamérica
- Coordinador del proyecto Tecnología Educativa para Discapacidad – TEDI-Actual
- Coordinador de varios proyectos semilla
- Miembro activo de la Academia Bolivariana, ABA
- Organizador de varios congresos CISMA. redEC
- Capacitador a Docentes en varios cursos
- Autor de libros, capítulos de libros, ha escrito más de 100 artículos científicos en revistas indexadas de alto impacto



Universidad
Nacional
de Loja

1859

Herramientas digitales para la enseñanza de la programación en el ámbito educativo

Herramientas digitales para la enseñanza de la programación en el ámbito educativo, es una iniciativa con el propósito de impulsar la programación en los entornos formativos y en los diferentes niveles de estudio, además busca desarrollar habilidades como el pensamiento computacional de forma interactiva, a través de diferentes recursos y plataformas on-line, e impulsar el uso de estrategias activas para solventar problemas en el aula con el uso de la tecnología.

Es por eso, que el presente libro se enfoca en las herramientas digitales que impulsen la programación educativa, donde se analizan las funciones y características que permiten desarrollar las habilidades informáticas de los estudiantes, además brinda a los docentes una guía para fomentar diferentes estrategias de enseñanza con el uso de las Tecnologías de la Información y Comunicación (TIC); recalando que cada capítulo esta enriquecido con información esencial para usar las tecnologías educativas de forma innovadora.

