



Revista EIA
ISSN 1794-1237
e-ISSN 2463-0950
Año XIX/ Volumen 22/ Edición N.44
Julio - diciembre 2025
Reia4418 pp. 1-28

Publicación científica semestral
Universidad EIA, Envigado, Colombia

PARA CITAR ESTE ARTÍCULO / TO REFERENCE THIS ARTICLE /

Ramírez Ramírez, D. J.; Marín Ríos, L.
A.; Correa Álvarez, C. D.

Fortalecimiento del Pensamiento
Computacional en Estudiantes de
Secundaria Mediante la Programación
con Arduino UNO

Revista EIA, 22(44), Reia4418 pp. 1-28
<https://doi.org/10.24050/reia.v22i43.1871>

✉ *Autor de correspondencia:*

Ramírez Ramírez, D. J.
Ingeniero Químico
Correo electrónico:
ddramirezra@unal.edu.co

Recibido: 05-03-2025

Aceptado: 10-06-2025

Disponible online: 01-07-2025

Fortalecimiento del Pensamiento Computacional en Estudiantes de Secundaria Mediante la Programación con Arduino UNO

✉ DIÓGENES DE JESÚS RAMÍREZ RAMÍREZ¹

LAURA ANDREA MARÍN RÍOS¹

CRISTIAN DAVID CORREA ÁLVAREZ¹

1. Universidad Nacional de Colombia, Sede Manizales

Resumen

El Pensamiento Computacional se ha convertido en una competencia esencial para la educación en la era de la Industria 4.0, especialmente en estudiantes de secundaria. El objetivo del artículo es fortalecer el Pensamiento Computacional de estudiantes de secundaria mediante programación informática utilizando el microcontrolador Arduino UNO, integrándolo en actividades pedagógicas que fomentan la resolución de problemas y la automatización. La metodología del trabajo tuvo un enfoque cualitativo no experimental, donde los estudiantes con conocimientos previos en programación diseñaron, construyeron y probaron prototipos físicos que solucionaran problemas reales, iniciaron con una prueba diagnóstica sobre lógica de programación, luego una introducción, sintaxis de programación e interacción con el microcontrolador. Los resultados mostraron un aumento significativo en las competencias técnicas y en la motivación de los estudiantes hacia el aprendizaje activo de la programación y la robótica. Asimismo, se destacó el valor del aprendizaje colaborativo, la creatividad y la capacidad de resolución de problemas, ya que los estudiantes trabajaron en equipo para desarrollar sus proyectos. No obstante, se identificaron algunas limitaciones, como la disponibilidad de recursos materiales y la necesidad de mayor capacitación para los docentes. En conclusión, el uso de Arduino UNO como herramienta educativa contribuye al desarrollo de habilidades tecnológicas, ofreciendo un entorno accesible y efectivo del aprendizaje activo crucial en la preparación de los estudiantes para los desafíos de la Industria 4.0 y un mundo cada vez más digitalizado.

Palabras clave: Educación; pensamiento computacional; programación informática; aprendizaje activo; resolución de problemas; Arduino UNO; lógica de programación; microcontrolador; industria 4.0; circuitos electrónicos.

Strengthening Computational Thinking in High School Students through Programming with Arduino UNO

Abstract

Computational Thinking has emerged as an essential skill in education, particularly in the context of Industry 4.0, especially for high school students. This article aims to enhance computational thinking among secondary school students through computer programming using the Arduino UNO microcontroller, integrated into pedagogical activities that foster problem-solving and automation. The research methodology adopted a non-experimental qualitative approach, in which students with prior programming knowledge designed, constructed, and tested physical prototypes to address real-world problems. The process began with a diagnostic test to assess programming logic, followed by an introduction to programming syntax and interaction with the microcontroller. The results demonstrated significant improvements in students' technical skills and their motivation for active learning in programming and robotics. Additionally, collaborative learning, creativity, and problem-solving abilities were highlighted as students worked in teams to develop their projects. However, certain limitations were identified, including the availability of material resources and the need for further teacher training. In conclusion, the use of the Arduino UNO as an educational tool effectively contributes to the development of technological competencies and provides an accessible and engaging environment for active learning. This approach is vital in preparing students for the challenges of Industry 4.0 and an increasingly digitized world.

Keywords: education; computational thinking; computer programming; active learning; problem-solving; Arduino UNO; programming logic; microcontrollers; industry 4.0; electronic circuits.

1. Introducción

La rápida evolución de la tecnología ha transformado significativamente el panorama educativo, lo que hace necesaria la integración de métodos y herramientas de enseñanza innovadoras para preparar a los estudiantes frente a las demandas del siglo XXI. La Industria 4.0, caracterizada por la fusión de los mundos físico, digital y biológico, destaca la necesidad de habilidades en programación, Pensamiento Computacional (PC) y resolución de problemas (Narong y Hallinger, 2024; Rodriguez-Sanchez et al., 2024). En consecuencia, las instituciones educativas deben adaptarse incorporando estas competencias en sus planes de estudio, especialmente en la educación básica secundaria, donde se cultivan los conocimientos e intereses fundamentales (Dimitrova y Marinova, 2024; Umbara, 2024). Por lo anterior, este artículo presenta una propuesta práctica con actividades pedagógicas para fortalecer el PC en estudiantes de secundaria mediante programación con Arduino UNO.

La Industria 4.0, también conocida como la cuarta revolución industrial, representa un cambio significativo en el ámbito industrial a nivel mundial y la creación de sistemas autónomos e inteligentes (Mercorelli et al., 2024). Esta nueva era digital se caracteriza por la implementación de tecnologías avanzadas, que permiten la automatización de procesos, contribuyendo al desarrollo y mejora de la manufactura, el incremento de las utilidades empresariales y la satisfacción de los clientes. Las tecnologías emergentes que impulsan esta Industria incluyen el internet de las cosas (IoT), la inteligencia artificial (IA), los sistemas ciberfísicos y la robótica (Manikanawar et al., 2021; Udvaros et al., 2023). Estas innovaciones no solo permiten una mayor eficiencia, sino que también transforman la manera en que las personas y las máquinas interactúan en los entornos productivos (Gajdzik y Wolniak, 2021).

Existen cuatro enfoques conceptuales que se destacan en el marco de esta transformación: el primero, relacionado con el impacto social que genera la automatización en los métodos de producción y en la participación humana; el segundo, centrado en las competencias del personal necesario para operar y mantener estos sistemas; el tercero, relacionado con la modernización de los procesos gracias a la automatización; y el cuarto, que aborda la reciprocidad y

colaboración entre humanos y máquinas (Deepti Raj et al., 2024; Kumar et al., 2021).

La transformación digital en la educación es esencial para alinearse con estas demandas. Dimitrova y Marinova (2024) analizaron la transformación digital de la educación superior, destacando la necesidad de marcos educativos modernos. Rodriguez-Sanchez et al. (2024) examinaron los enfoques de aprendizaje transformador, subrayando la importancia de la Educación 4.0 para fomentar aprendices adaptables e innovadores. Umbara (2024) discutió la revolución de la Educación 4.0, enfatizando el impacto transformador en los sistemas educativos tradicionales.

En Colombia, la Industria 4.0 ha comenzado a desarrollarse en los últimos años, aunque su adopción aún se encuentra en una fase temprana. El Ministerio de Comercio, Industria y Turismo (2023) ha destacado que la Industria 4.0 representa una oportunidad para que el país aumente su competitividad, mejore la productividad y genere empleo de calidad. A través de políticas y programas como el Programa Nacional de Transformación Productiva (PNT), el Plan Nacional de Desarrollo y la Alianza para la Cuarta Revolución Industrial, el gobierno ha impulsado la implementación de tecnologías avanzadas en las industrias (Gobierno de Colombia, 2023). También, se busca impulsar esas tecnologías desde la interacción en el aula fortalecer competencias informáticas desde la educación STEM.

Por otro lado, la educación STEM es un enfoque educativo que se centra en la enseñanza de carreras en campos de alta demanda en la industria, las ciencias, la tecnología, la ingeniería y las matemáticas para aportar a la solución de problemas del mundo real (Mater et al., 2023; Nguyen et al., 2021; Udvaros et al., 2023). En el año 2016 el ministerio de Educación de Colombia adopta el modelo educativo STEM para promover y estimular la innovación en las estrategias educativas de los docentes, mediante el uso de tecnologías digitales que promuevan habilidades técnicas esenciales para la Industria 4.0 (Balon et al., 2021; Marín-Marín et al., 2024), con la aplicación de metodologías innovadoras de enseñanza y aprendizaje, donde los

estudiantes sean más receptivos y puedan desarrollar sus habilidades a partir de proyectos, prácticas de laboratorio y herramientas tecnológicas (de Educación, 2016).

El PC, por su parte, es una habilidad clave en el contexto de la Industria 4.0. Este enfoque promueve el uso de estrategias computacionales para resolver problemas, incluyendo la descomposición de problemas, la abstracción, el reconocimiento de patrones y la creación de algoritmos (Corrales-Álvarez et al., 2024; Montiel y Gomez-Zermeño, 2021). Esta habilidad es fundamental para enfrentar los desafíos tecnológicos del siglo XXI, donde la globalización y la digitalización son pilares de los nuevos sistemas de producción. Alsina y Acosta (2022) subrayan la importancia de fomentar competencias matemáticas mediante la aplicación de habilidades de PC a través de enfoques lúdicos y actividades específicas para resolver problemas.

En ese sentido, es necesario integrar el PC y el Pensamiento Matemático desde la educación básica, como lo propone la Alianza para el Aprendizaje del siglo XXI (Akib y Muhsin, 2019) para el éxito en disciplinas STEM (Science, Technology, Engineering, and Mathematics) y más allá (Avalos Dávila et al., 2021; Lee y Kwon, 2024; Narong y Hallinger, 2024). Sin embargo, muchos sistemas educativos luchan por integrar efectivamente el PC en sus prácticas de enseñanza, a menudo debido a la falta de recursos, experiencia o enfoques pedagógicos adecuados (Cabezas y Carrera, 2024; Ruiz-Bolivar y Ríos-Cabrera, 2020; Soenarto et al., 2020). Por lo cual, en este artículo se busca fortalecer el PC de estudiantes de secundaria a través de la programación informática con el microcontrolador Arduino UNO.

Por su parte, Arduino UNO, una plataforma de microcontrolador de código abierto muy popular, presenta una solución viable para mejorar el PC en los estudiantes de secundaria. Su interfaz amigable y el extenso apoyo de la comunidad en línea lo convierten en una herramienta accesible tanto para docentes como para estudiantes (Hurtuk et al., 2017; Kondaveeti et al., 2021). A través de proyectos prácticos con Arduino, los estudiantes pueden aprender conceptos de programación y electrónica digital, identificar patrones, crear algoritmos, abstraer información compleja y descomponer problemas

en partes manejables (Dimitrova y Marinova, 2024; El Hadi et al., 2020; Hernández et al., 2024).

El uso de Arduino en entornos educativos ha sido ampliamente explorado, destacando su versatilidad e impacto en el compromiso y comprensión de los estudiantes. Buachoom et al. (2019) desarrollaron un sistema de medición de distancia utilizando una placa Arduino con un sensor ultrasónico, demostrando su aplicación en proyectos en tiempo real. Sulayman et al. (2024) mejoraron la precisión de monitoreo y el tiempo de respuesta mediante un sistema de monitoreo en tiempo real basado en un microcontrolador Arduino. Bianchi et al. (2024) mejoraron el compromiso de los estudiantes en el ensamblaje de circuitos con BlinkBoard, una herramienta educativa que guía y monitorea las tareas de ensamblaje de circuitos.

En consecuencia, la integración de Arduino UNO en entornos educativos se alinea con el cambio pedagógico hacia el aprendizaje experiencial, debido a que permite involucrar a los estudiantes en la construcción y programación de dispositivos electrónicos y mejorar su comprensión de conceptos abstractos en ciencia e ingeniería (Marín-Marín et al., 2024). Además, herramientas como Arduino se destacan como un recurso valioso para desarrollar el PC a través del diseño y construcción de proyectos interactivos, que fomentan la creatividad e innovación en los estudiantes, desde sistemas de automatización simples hasta robots complejos (Kondaveeti et al., 2021; Sulayman et al., 2024).

Este artículo contiene entre otras las siguientes secciones: en la Sección 2 se presentan los materiales y método, donde se describen los participantes, los instrumentos de recolección de datos y los criterios de evaluación. Luego, en la Sección 3, se muestran los resultados del presente trabajo dividido en 6 subsección relacionadas al PC, con Arduino UNO y la discusión relacionando los resultados del estudio con algunos autores de la literatura. Finalmente, en la Sección 4 se observa las conclusiones generales del artículo.

2. Materiales y métodos

El estudio se llevó a cabo con un enfoque cualitativo no experimental, en el cual se implementaron actividades de

programación con Arduino UNO en un grupo de estudiantes de secundaria (Guevara et al., 2020). Participaron 8 estudiantes de grado 11 de una institución educativa pública en Colombia, con edades entre 15 y 17 años, y un nivel básico de conocimientos en programación y electrónica durante el primer semestre de 2023. La selección de los participantes se realizó de manera intencional, considerando su interés en actividades relacionadas con la tecnología y la programación.

Los estudiantes realizaron actividades prácticas para familiarizarse con el entorno de programación de Arduino y los componentes electrónicos, incluyendo ejercicios básicos de programación y la construcción de circuitos simples utilizando sensores y actuadores. Posteriormente, aplicaron los conocimientos adquiridos para diseñar y construir un prototipo funcional que abordara el problema identificado en la fase de planificación. Seguidamente, se realizaron evaluaciones formativas para medir el progreso de los estudiantes en el desarrollo de competencias en programación y PC. Estas evaluaciones incluyeron pruebas prácticas y la revisión de los prototipos desarrollados. Los instrumentos de recolección de datos y los criterios de evaluación.

2.1. Instrumentos de Recolección de Datos

Se utilizaron varios instrumentos para la recolección de datos, entre los cuales se incluyeron los siguientes: observación directa, que consistió en registrar el comportamiento y la participación de los estudiantes durante las sesiones de trabajo; entrevistas semiestructuradas, realizadas tanto a estudiantes como a docentes para profundizar en sus experiencias y percepciones sobre el uso de Arduino en el aula; y análisis de productos, en el cual se evaluaron los prototipos desarrollados por los estudiantes para determinar su nivel de complejidad y la aplicación de conceptos de programación y electrónica.

2.2. Criterios de evaluación

Los criterios de evaluación de los productos de los estudiantes se definieron con diferentes instrumentos, de acuerdo con la actividad desarrollada. La prueba diagnóstica de lógica de programación

se evaluó con 8 preguntas: dos preguntas relacionadas con la representación de algoritmos a través de diagramas de flujo y seis preguntas relacionadas con la representación de algoritmos a través de pseudocódigos en la plataforma *Pseint*. En cuanto a la introducción a Arduino UNO y el reconocimiento de los dispositivos del kit de electrónica, inicialmente se realizó una presentación y luego se formularon preguntas a los participantes, asociadas a los componentes. Adicionalmente, se llevó a cabo una dinámica con globos; en su interior había preguntas escritas en trozos de papel de diferentes tipos y con diferente puntuación.

Por otra parte, la actividad relacionada con la sintaxis de programación en Arduino UNO y el uso de simuladores se evaluó mediante un proceso estructurado de la siguiente manera: inicialmente, los estudiantes leyeron una serie de procesos de automatización que debían ser representados en códigos de programación; luego, analizaron y seleccionaron las estructuras de control adecuadas para representar dichos procesos; seguidamente, implementaron la representación de estos procesos mediante códigos de programación en el entorno de desarrollo de Arduino UNO. Además, simularon en Tinkercad el montaje electrónico de cada proceso y comprobaron su funcionamiento.

Para evaluar los fundamentos de electrónica, los estudiantes se organizaron en grupos de dos y se les solicitó que organizaran un montaje de circuitos electrónicos en serie y en paralelo, realizaran la medición de corriente, voltaje y resistencia con un multímetro, representaran los circuitos mediante diagramas eléctricos y determinaran valores teóricos de corriente, voltaje y resistencia mediante la Ley de Ohm, para compararlos con los valores medidos. Finalmente, la interacción con Arduino UNO fue evaluada de forma práctica con códigos de programación usando distintas estructuras de control que permitieron la automatización de los procesos planteados en sus proyectos.

3. Resultados y discusión

En esta sección se presenta inicialmente una prueba diagnóstica de lógica de programación para evaluar cómo los estudiantes

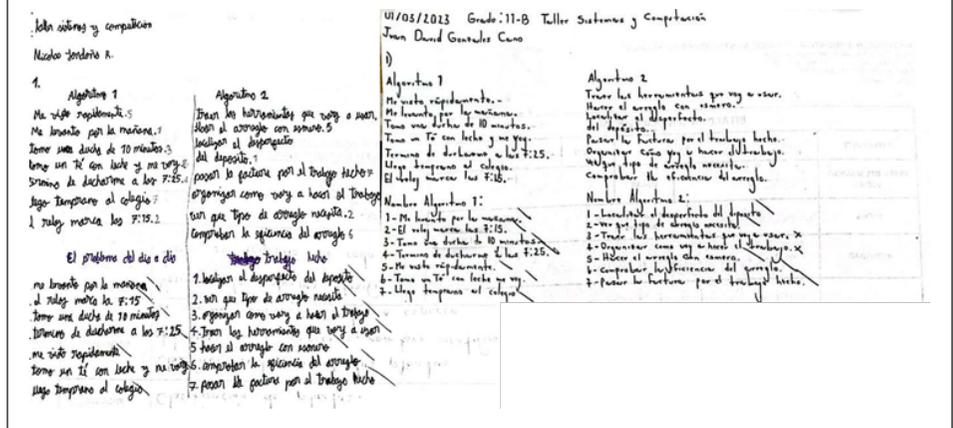
comprenden los procedimientos esenciales al plantear un algoritmo (subsección 3.1). A continuación, se introduce Arduino UNO mediante una presentación que abarca su definición, reconocimiento de los componentes de la placa, funcionamiento de microcontroladores, configuración del entorno de desarrollo y se identifican los componentes electrónicos digitales del kit incluido con Arduino UNO (subsección 3.2). Luego, se enseña la sintaxis de programación en el entorno de Arduino UNO, impartiendo sus fundamentos a los estudiantes (subsección 3.3). Seguidamente, se realiza una actividad sobre conceptos eléctricos básicos como voltaje, corriente, resistencia y su aplicación en la ley de Ohm (subsección 3.4). Finalmente, se plantea a los estudiantes diseñar y desarrollar procesos de automatización y control para solucionar problemáticas sociales, ambientales, económicas, educativas o lúdicas utilizando Arduino UNO (subsección 3.5). Finalmente, se presenta la discusión de los resultados en la subsección 3.6

3.1. Prueba diagnóstica sobre lógica de programación:

La prueba diagnóstica de programación consistió en evidenciar los conocimientos básicos de programación de los estudiantes relacionados con: variables, sintaxis de programación en pseudocódigos y uso de estructuras de control fundamentales. Constaba de 8 preguntas, 2 relacionadas con la representación de algoritmos a través de diagramas de flujo y 6 preguntas relacionadas la representación de algoritmos a través de pseudocódigos en la plataforma *Pseint*. Durante el desarrollo, se evidenció que los estudiantes siguieron correctamente las instrucciones y hubo un buen comportamiento durante la misma. La mayoría de los estudiantes solicitaron ocasionalmente ayuda para la comprensión de algunas preguntas y su interpretación en forma de pseudocódigo.

La primera pregunta estaba relacionada con la organización de dos algoritmos de lógica de programación, no generó mayor dificultad para los estudiantes, pues evidenciaron fácilmente el orden correcto en que se debía escribir cada uno como se evidencia en la Figura 1.

Figura 1. Organización de algoritmos de la prueba diagnóstica de lógica de programación



La segunda pregunta relacionada con la representación de algoritmos a través de diagramas de flujo se observa en la Figura 2, en esta surgieron algunas dificultades por parte de los estudiantes:

- Cuatro estudiantes desarrollaron el diagrama de flujo, pero con errores de sintaxis, mal uso de la simbología y / o flechas de flujo.
- Dos estudiantes no desarrollaron este punto debido a que el año anterior no estaban en el taller y por lo tanto no tuvieron esta formación en diagramas de flujo.
- Solo un estudiante desarrolló el diagrama de flujo correctamente, siguiendo las normas de sintaxis, haciendo un buen uso de los símbolos y flechas de flujo.

Figura 2. Representación de algoritmos con diagramas de flujo por parte de los estudiantes



Con respecto a la tercera y cuarta pregunta, que consistían en desarrollar pseudocódigos en *Pseint* para solicitar datos por teclado y mostrar mensajes en consola, no hubo dificultad para los estudiantes, usaron correctamente la lógica y sintaxis de programación en el desarrollo de estos. Como se observa en la Figura 3, el propósito del programa es pedirle al usuario su edad y peso, y luego mostrar esos datos en pantalla. Para esto, se declaran dos variables llamadas edad y peso de tipo entero; estas variables servirán para guardar los datos que ingrese el usuario, se muestra un mensaje por pantalla preguntándole al usuario su edad y peso, el programa se detiene y espera a que el usuario ingrese un valor que se almacenará en las respectivas variables. Por último, se imprime en pantalla un mensaje que combina texto y el valor que el usuario ingresó para la edad y el peso.

Figura 3. Desarrollo de pseudocódigo en *Pseint* para solicitar datos por teclado

```

1 // Desarrollo de pseudocódigo en Pseint para solicitar datos por teclado
2 Algoritmo Edad_Peso // Se inicia la definición del algoritmo y se le da el nombre Edad_Peso
3   Definir edad,peso Como Entero; //Se declaran dos variables llamadas edad y peso,tipo entero
4   Escribir '¿Cual es tu edad?'; // Se muestra un mensaje por pantalla preguntándole al usuario su edad
5   Leer edad; // El programa se detiene y espera a que el usuario ingrese un valor (número entero)
6   Escribir '¿Cual es tu peso?'; // Se muestra otro mensaje por pantalla para preguntar el peso del usuario.
7   Leer peso; // El programa espera que el usuario escriba su peso (un número entero)
8   Escribir 'Tu edad es:',edad,'años'; // Se imprime en pantalla un mensaje que combina texto
9   Escribir 'Tu peso es:',peso,'kg'; // Se muestra otro mensaje que incluye el peso ingresado por el usuario
10 FinAlgoritmo
11

```

Las respuestas de la quinta y sexta pregunta, que consistían en desarrollar pseudocódigos usando condicionales, no hubo dificultad para la mayoría de los estudiantes, sin embargo, si se requirió de un mayor apoyo para tres estudiantes. La Figura 4 muestra este algoritmo, el cual tiene como objetivo comparar dos números y determinar cuál es el mayor, a través de una estructura condicional “Sino”. Para esto se declaran dos variables llamadas “num1” y “num2”, ambas de tipo entero. Estas variables van a almacenar los dos números que el usuario ingresará, luego el programa espera que el usuario escriba dos números, uno por uno. Seguidamente, comienza una estructura condicional (“si” – “Sino”) para comparar si el primer número es mayor que el segundo. Si la condición anterior es verdadera, entonces el programa muestra en pantalla que “num1”

es el número mayor. Si la condición anterior no se cumple, entonces el programa muestra que “num2” es el número mayor.

Figura 4. Desarrollo de pseudocódigo en Pseint usando condicionales

```

1 Algoritmo Condicional_SiNo // Inicia el algoritmo y le da un nombre: Condicional_SiNo
2 definir num1 Como Entero; // Se están declarando dos variables llamadas num1 y num2, ambas de tipo Entero
3 definir num2 Como Entero;
4 // Muestra un mensaje en pantalla para indicarle al usuario que debe escribir dos números
5 Escribir "Dime dos numeros y te diré cual es el mayor";
6 leer num1; // El programa espera que el usuario escriba dos números, uno por uno
7 leer num2;
8 // Se compara si el primer número (num1) es mayor que el segundo (num2).
9 si num1>num2 Entonces
10     //Si la condición anterior es verdadera, el programa muestra en pantalla que num1 es el número mayor.
11     Escribir "El numero mayor es:" num1;
12 SiNo
13     // Si la condición anterior no se cumple, el programa muestra en pantalla que num2 es el número mayor.
14     Escribir "El numero mayor es:" num2;
15 FinSi // Indica el fin de la decisión entre las dos opciones.
16
17 FinAlgoritmo
18

```

En la séptima pregunta, los estudiantes enfrentaron un desafío considerable al tener que utilizar una estructura de repetición conocida en otros lenguajes de programación, pero poco familiar en la plataforma *Pseint*. Para resolverlo, debieron relacionar el funcionamiento de esta estructura tal como la conocían y adaptarla a la sintaxis específica de *Pseint*. Tras varios intentos con el apoyo ocasional de la docente, todos lograron superar el reto. Como se muestra en la Figura 5, comienza la definición del algoritmo, que tiene por nombre “Edad_sexo”. Se declaran dos variables, de tipo entero y carácter, cada una. El bloque “repetir hasta cuando” asegura que el usuario ingrese una edad válida (mayor que cero) y un sexo válido (“f” para mujer ó “m” para hombre). Luego el programa muestra en pantalla “Usted es mujer” ó “Usted es hombre”. La estructura condicional verifica si la persona tiene 18 años o más y si esta condición se cumple imprime “Usted puede votar”, Si no, imprime “Usted no puede votar”.

Figura 5. Desarrollo de pseudocódigo usando condicionales en Pseint más complejos y el otro muestra una operación aritmética

```

1 Algoritmo Edad_sexo // definición del algoritmo, que tiene por nombre Edad_sexo
2   definir a como entero; // Se declara la variable a como un número entero
3   definir sexo Como Caracter; // Se declara la variable sexo como un caracter
4   Repetir
5     Escribir "Su edad es"; // Muestra el mensaje "Su edad es"
6     leer a; // Lee la edad en la variable a
7   Hasta Que a>0 //Repite hasta que el valor ingresado sea mayor que cero
8
9   Repetir
10    Escribir "Su sexo es"; // Muestra el mensaje "Su sexo es"
11    leer sexo; // Lee el sexo en la variable sexo
12  Hasta Que sexo="f" o sexo="m" // "f" para mujer y "m" para hombre
13
14  si sexo= "f" Entonces // Si es "f", muestra el mensaje "Usted es mujer"
15    Escribir "Usted es mujer";
16  SiNo // En cualquier otro caso, muestra "Usted es hombre"
17    Escribir "Usted es hombre";
18  FinSi
19
20  si a≥ 18 Entonces // Si la edad es mayor o igual, imprime el mensaje en pantalla
21    Escribir "Usted puede votar";
22  SiNo // Sino cumple la anterior condición, imprime el siguiente mensaje en pantalla
23    Escribir "Usted no puede votar";
24  FinSi

```

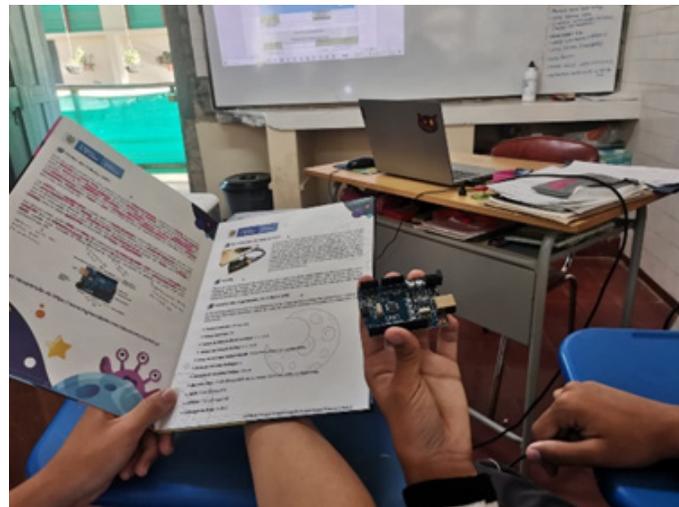
Ocho estudiantes presentaron la prueba, de los cuales tres requirieron un poco más de apoyo que los demás. Según la variable establecida para esta prueba, representa coherentemente algoritmos a través de diagramas de flujo y pseudocódigos, el 42% de los estudiantes lograron cumplir con esta competencia en lo referente a la representación de algoritmos mediante diagramas de flujo, mientras que el 95% alcanzó la competencia en la representación de algoritmos mediante pseudocódigos, con correcta compilación y funcionamiento. Sin embargo, ¿a qué se debe el bajo porcentaje de competencia en la representación de algoritmos mediante diagramas de flujo? Esto se atribuye a vacíos de aprendizaje y a la escasa práctica de los estudiantes en el uso de diagramas de flujo para describir algoritmos en orden secuencial, aplicando las normas adecuadas de representación.

3.2. Introducción a Arduino UNO y reconocimiento dispositivos kit de electrónica

En esta actividad se presentó Arduino UNO, abordando su definición, los componentes de la placa de hardware, el funcionamiento de los microcontroladores y la configuración del entorno de desarrollo, como se muestra en la Figura 6. Además, se destacó su valor como herramienta para la enseñanza de la

programación y la interacción con el entorno digital, ilustrando sus aplicaciones en diversos campos como la industria, la medicina y el entretenimiento. Durante la actividad, también se identificaron los componentes electrónicos digitales incluidos en el kit de electrónica que acompaña al Arduino UNO. Se realizaron reconocimientos tanto a nivel físico como funcional de dispositivos como sensores, actuadores, motores DC, resistencias y otros elementos.

Figura 6. Reconocimiento de dispositivo electrónicos por los estudiantes

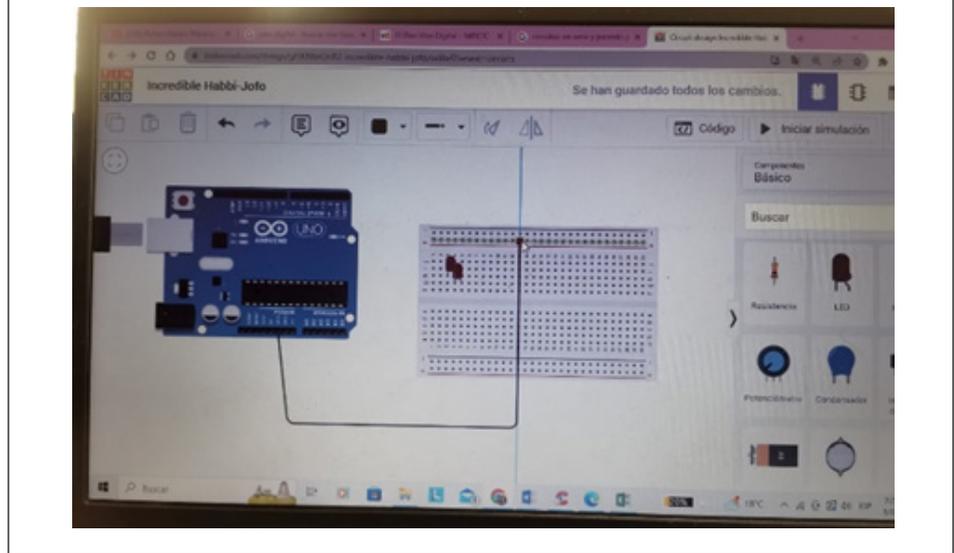


Durante la valoración del momento evaluativo, se realizaron diversas preguntas a los estudiantes sobre los componentes de la placa de hardware Arduino UNO. Se evidenció que las preguntas abiertas representaron un mayor desafío, ya que exigían un nivel más alto de análisis e interpretación por parte de los estudiantes. Por otro lado, las preguntas cerradas, como las de selección múltiple y las de verdadero o falso, resultaron ser de gran ayuda porque los estudiantes se sintieron más seguros al responder. Algunas de estas preguntas cerradas incluían distractores, lo que demandaba de los estudiantes la habilidad de identificarlos y evitar caer en ellos. Ante estos distractores, los estudiantes lograron identificarlos y superarlos satisfactoriamente. En las preguntas de completar definiciones con selección de múltiples opciones, se observó en los estudiantes una habilidad para interpretar, relacionar y construir cada definición de manera correcta.

De acuerdo con la variable establecida para este momento evaluativo, reconoce la función de los componentes que conforman la placa de hardware Arduino UNO y los dispositivos electrónicos digitales del kit de electrónica, el 91% de los estudiantes aplicaron de manera adecuada la habilidad de PC relacionada con el reconocimiento de patrones. Esta habilidad les permitió identificar y asociar términos, lo que facilitó el reconocimiento de los componentes de la placa Arduino UNO y el funcionamiento de los dispositivos electrónicos incluidos en el kit de electrónica.

3.3 Sintaxis de programación en Arduino UNO y uso de simuladores

Durante esta actividad, se impartieron a los estudiantes los fundamentos de la sintaxis en el entorno de programación de Arduino UNO. Estos conceptos incluyeron la identificación de datos y tipos de datos, así como el uso de variables y operadores aritméticos, lógicos y de comparación. Además, se exploró el funcionamiento de estructuras de control como *“if - else”*, *“for”*, *“switch”*, *“while”* y *“do while”* para la creación de códigos de programación. También se presentó la plataforma en línea Tinkercad (Autodesk, 2024), utilizada para la simulación de circuitos electrónicos, presentado en la Figura 7. Los estudiantes ingresaron, se registraron y exploraron su entorno visual, realizando prácticas sencillas que les permitieron familiarizarse con las diferentes herramientas del simulador para la representación de estos circuitos.

Figura 7. Uso de simuladores de circuitos electrónicos

Para el momento evaluativo de estas actividades, se pudo observar que, seis de los ocho estudiantes lograron analizar, interpretar y seleccionar correctamente los diagramas de flujo que representaban cada proceso de automatización según su estructura de control. Aquellos estudiantes que inicialmente cometieron errores en la elección de las estructuras de control para cada algoritmo identificaron sus fallos por sí mismos y realizaron las correcciones necesarias.

Por otra parte, durante el desarrollo de los códigos de programación, los estudiantes recibieron un apoyo inicial de la docente para reconocer la sintaxis de programación que debían emplear en la representación de cada algoritmo, utilizando las estructuras de control previamente seleccionadas en los diagramas de flujo. Posteriormente, los estudiantes desarrollaron de forma autónoma los códigos para cada algoritmo, los resultados se muestran en la Figura 8. Se observó que, todos los estudiantes lograron representar en el simulador Tinkercad el circuito electrónico correspondiente a cada proceso automatizado mediante los códigos de programación.

Figura 8. Desarrollo de código de programación

```

Ciclo_Else_If$
int i=0;
void setup() {
  Serial.begin(9600);
  pinMode (13,OUTPUT);
}

void loop() {
  if (i, i++)
  { Serial.println("Hola mundo");
  digitalWrite (13,HIGH);
}

else (i=20);
digitalWrite (13,LOW);

delay(1000);
}

ciclo_switch
int temperatura=32;
void setup() {
  Serial.begin (9600);
  pinMode (13, OUTPUT);
  pinMode (12, OUTPUT);
  pinMode (11, OUTPUT);
}

void loop() {
  switch(temperatura)
  { case 18:
    Serial.println("Dia frio");
    digitalWrite (13, HIGH);
    delay(2000);
    break;

  case 28:
    Serial.println("Dia agradable");
    digitalWrite (12, HIGH);
    delay(2000);
    break;

  default:
  case 32:
    Serial.println("Dia caluroso");
}

ciclo_for$
void setup() {
  Serial.begin(9600);
  pinMode (13,OUTPUT);
}

void loop() {
  for (int i=0;i<=20;i++){
    Serial.println("hola mundo");
    digitalWrite (13, HIGH);
    delay(1000);

    Serial.println (i=20; digitalWrite(13,LOW);
  }
}
    
```

Según la variable establecida para este momento evaluativo, interpreta diagramas de flujo para el desarrollo de códigos de programación y simula el funcionamiento de un circuito electrónico, el 90% de los estudiantes aplicaron de manera adecuada la habilidad de Pensamiento Computacional relacionada con el desarrollo de algoritmos. Esta habilidad les permitió interpretar diagramas de flujo y crear algoritmos a partir de ellos, facilitando la automatización de un proceso en electrónica y la simulación de su respectivo circuito electrónico.

3.4 Fundamentos de electrónica

La actividad comenzó con una introducción a conceptos eléctricos básicos, como voltaje, corriente, resistencia, y su aplicación en la Ley de Ohm. Luego, se proyectaron y retroalimentaron dos videos que explican el funcionamiento, características y diferencias entre la electrónica analógica y digital. Posteriormente, se estudió el funcionamiento de los principales dispositivos electrónicos analógicos y digitales, tales como diodos, transistores, circuitos integrados, resistencias y relevadores. Finalmente, se aplicaron estos conceptos para diseñar circuitos en serie y en paralelo, permitiendo el correcto funcionamiento de diversos dispositivos electrónicos.

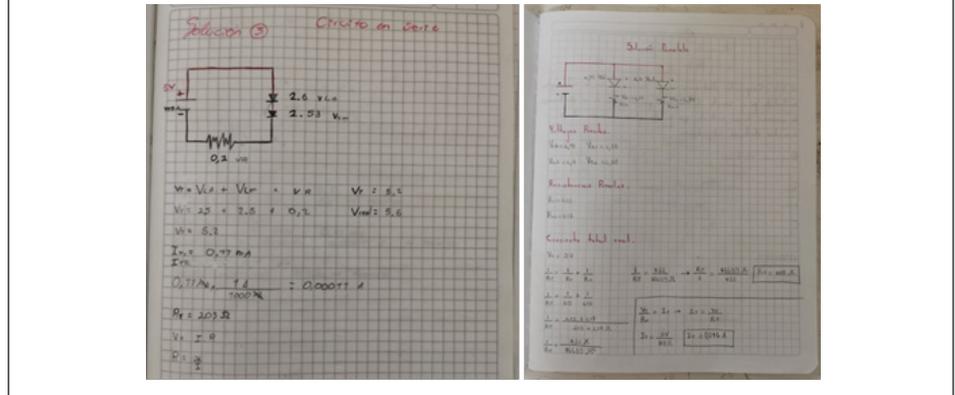
El momento evaluativo de esta actividad consistió en la creación de montajes de circuitos electrónicos en serie y en paralelo, la medición de corriente, voltaje y resistencia con un multímetro, la

representación de los circuitos mediante diagramas eléctricos y la determinación de valores teóricos de corriente, voltaje y resistencia mediante la Ley de Ohm, para compararlos con los valores medidos. Los estudiantes, organizados en parejas, utilizaron recursos como computadoras portátiles, conexión wifi, el simulador Tinkercad, la placa Arduino UNO, dispositivos del kit de electrónica, un multímetro, cuadernos para registrar diagramas eléctricos, valores teóricos y reales. Durante la actividad, la docente brindó acompañamiento continuo para resolver dudas y verificar el funcionamiento de los circuitos y su diagramación eléctrica.

En el montaje de los circuitos electrónicos para encender dos LEDs en serie y paralelo, seis de los ocho estudiantes realizaron correctamente las conexiones entre LEDs y resistencias, considerando la alimentación de voltaje y la conexión a tierra de cada componente. En cuanto a la medición de voltaje y resistencia con el multímetro, los estudiantes emplearon correctamente el dispositivo, ajustando la escala de medición y ubicando los terminales de acuerdo con la polaridad de los componentes. Para la medición de corriente, algunos estudiantes necesitaron apoyo de la docente para posicionar correctamente los terminales en los circuitos.

En la Figura 9 se observa la representación del circuito en serie mediante un diagrama eléctrico, solo cuatro estudiantes lograron completarlo sin mayores dificultades; los demás requirieron apoyo de la docente. Sin embargo, al repetir el proceso para el circuito en paralelo, todos los estudiantes lograron su representación sin errores. Además, todos hicieron un uso adecuado de la Ley de Ohm para calcular los valores teóricos de voltaje, corriente y resistencia en los circuitos.

Figura 9. Representación de diagramas eléctricos y uso ley de ohm por los estudiantes



De acuerdo con la variable establecida para este momento evaluativo, realiza circuitos electrónicos en serie y paralelo, representa estos circuitos a través de diagramas eléctricos y analiza su funcionamiento a través de la Ley de Ohm, el 87% de los estudiantes aplicaron de manera adecuada la habilidad de Pensamiento Computacional relacionada con la abstracción. Esta habilidad les permitió seleccionar la información relevante, lo cual facilitó la construcción de los circuitos, su correcta representación mediante diagramas eléctricos y la medición posterior de valores eléctricos utilizando la Ley de Ohm.

3.5 Interactuando con Arduino UNO

En esta actividad se propuso a los estudiantes diseñar y desarrollar procesos de automatización y control para abordar problemáticas de tipo social, ambiental, económico, educativo o lúdico, utilizando sensores y otros componentes electrónicos del kit Arduino UNO, junto con su respectiva programación en el entorno de desarrollo de Arduino UNO. Una vez socializada la actividad, los estudiantes se organizaron en parejas e iniciaron una investigación para identificar problemas que pudieran ser solucionados o mejorados mediante la plataforma Arduino UNO. Los proyectos seleccionados por los estudiantes fueron los siguientes:

- Sistema de control y automatización para una puerta de garaje vehicular.
- Sistema de control para un juego lúdico de memoria y concentración.
- Sistema que proporciona un microclima adecuado para un invernadero de manera automática.
- Sistema de control para el manejo de luces estroboscópicas en respuesta a ondas sonoras.

A continuación, los estudiantes desarrollaron los códigos de programación usando distintas estructuras de control que permitieron la automatización de los procesos planteados en sus proyectos. Estos códigos fueron verificados mediante el compilador del software de Arduino UNO y luego cargados en el microcontrolador de sus respectivas placas de hardware. Los estudiantes clasificaron los componentes electrónicos digitales del kit Arduino UNO necesarios para implementar los circuitos en sus proyectos, empleando elementos como diodos LED, pulsadores, bocinas, cables jumpers, protoboards, sensores de ultrasonido, servomotores, transistores, relevadores y potenciómetros.

Posteriormente, se indicó a los estudiantes que diseñaran los diagramas eléctricos de los circuitos, calculando los valores adecuados de resistencia, corriente y voltaje para su funcionamiento mostrados en la Figura 10. Tras analizar los diagramas junto a la docente, se procedió a verificar su correcto funcionamiento en el simulador Tinkercad. Luego, realizaron el montaje físico de los circuitos electrónicos utilizando los componentes seleccionados y un multímetro para medir valores como resistencia, corriente, voltaje y continuidad, garantizando así el correcto funcionamiento de los montajes.

Figura 10. Desarrollo de códigos de programación por los estudiantes de proyectos Arduino UNO

```

//máquina_mquina_estado
const int entradaPin = 7; // pin de entrada para la variable high/low
const int salidaPin = 9; // pin de salida para la salida 9
const int salidaPinB = 8; // pin de salida para la salida 8

// Estados de la máquina de estados
enum Estado {
    ESTADO_1,
    ESTADO_2,
};

Estado estadoActual = ESTADO_1; // Estado inicial

void setup() {
    pinMode(entradaPin, INPUT); // Configurar el pin de entrada como entrada
    pinMode(salidaPin, OUTPUT); // Configurar el pin de salida 9 como salida
    pinMode(salidaPinB, OUTPUT); // Configurar el pin de salida 8 como salida
}

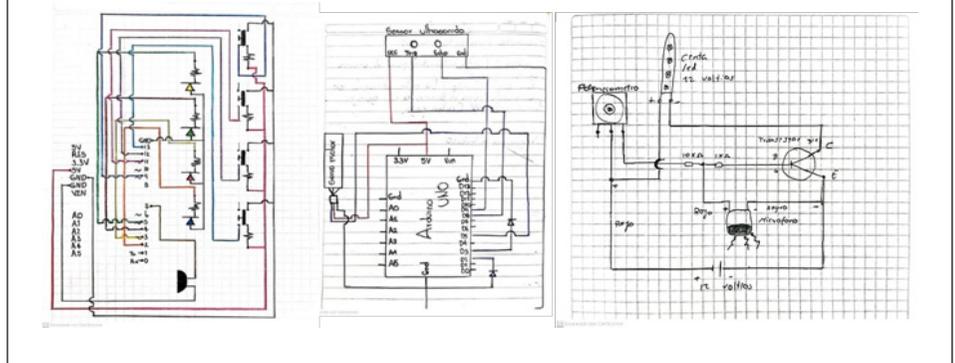
void loop() {
    int estadoEntrada = digitalRead(entradaPin); // Leer el estado de la entrada
}

// Se le define a los leds que se encienden segun al funcionamiento de la puerta
void onOffLeds() {
    digitalWrite(2,HIGH);
    digitalWrite(3,HIGH);
    delay(100);
    digitalWrite(2,LOW);
    digitalWrite(3,LOW);
    delay(100);
    digitalWrite(2,HIGH);
    digitalWrite(3,HIGH);
    delay(100);
    digitalWrite(2,LOW);
    digitalWrite(3,LOW);
}

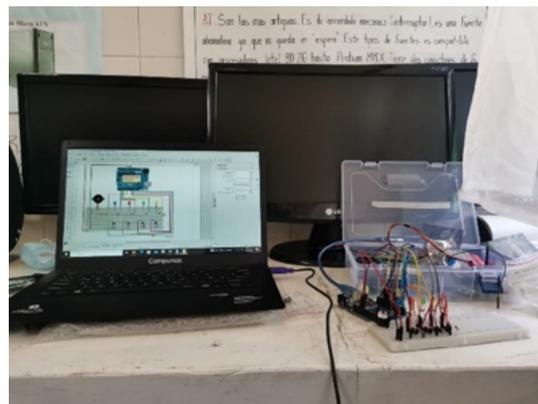
// Funcion que permite cerrar la puerta a travez de un estructura de funcion do while
void cerrarPuerta() {
    digitalWrite(2,LOW);
    digitalWrite(3,HIGH);
    int angulo=0;
}
    
```

En el desarrollo de los códigos de programación mostrados en la Figura 10, los estudiantes demostraron una adecuada comprensión de las estructuras secuenciales, condicionales y repetitivas necesarias para el control o la automatización que deseaban implementar en sus proyectos. Durante la compilación de sus códigos, la mayoría identificó y corrigió errores de programación con el apoyo de sus compañeros y la docente. Sin embargo, algunos estudiantes presentaron dificultades para desarrollar el código correcto y requirieron un mayor acompañamiento de la docente y el uso de libros de referencia.

Además, seis de los ocho estudiantes lograron representar e interpretar correctamente las conexiones entre los componentes electrónicos digitales y la placa Arduino UNO, teniendo en cuenta la resistencia, corriente y voltaje en los circuitos. Asimismo, demostraron un manejo adecuado de las herramientas de simulación en Tinkercad como se observa en la Figura 11.

Figura 11. Diagramas eléctricos de proyectos Arduino UNO realizado por los estudiantes

Una vez realizados y analizados los diagramas eléctricos junto a la docente, se procede a verificar su funcionamiento en el simulador Tinkercad. Posteriormente, los estudiantes montan los circuitos electrónicos utilizando componentes seleccionados, como diodos LED, entre otros, mencionados anteriormente. Con el uso de un multímetro, miden valores como resistencia, corriente, voltaje y continuidad para asegurar el correcto funcionamiento del montaje, como se muestra en la Figura 12. Durante la construcción de las maquetas, se evidenció que la mayoría de los estudiantes mostró un notable empeño e interés en participar activamente en su elaboración. Cada grupo gestionó los materiales necesarios y empleó su creatividad en el diseño y ejecución de las maquetas.

Figura 12. Simulación y realización de circuitos electrónicos en maquetas de proyectos Arduino UNO

De acuerdo con la variable establecida para este momento evaluativo, diseña, desarrolla y ejecuta procesos de automatización y control mediante Arduino UNO, el 87% de los estudiantes aplicaron de manera adecuada la habilidad de Pensamiento Computacional relacionada con la descomposición de problemas. Los sistemas de automatización y control suelen ser complejos y están compuestos por múltiples componentes interconectados. Al descomponer el problema en partes más pequeñas, los estudiantes lograron abordar cada componente por separado, lo cual facilitó el diseño, desarrollo y ejecución de sus proyectos.

3.6 Discusión

Los resultados obtenidos en este estudio demuestran que la implementación del microcontrolador Arduino UNO en la educación secundaria puede contribuir significativamente al fortalecimiento del Pensamiento Computacional de los estudiantes. La mayoría de los estudiantes participaron activamente con gran interés en las diferentes actividades, mostraron una mejor comprensión y habilidad para la representación de algoritmos a través de pseudocódigos y diagramas de flujo, a pesar de las dificultades iniciales con estos últimos. Estos hallazgos son consistentes con estudios previos como el de Marín-Marín et al. (2024) que destacan cómo el uso de plataformas de programación facilita la adquisición de competencias lógicas y de resolución de problemas. Además, la introducción de conceptos de programación en un entorno práctico como el de Arduino promovió un aprendizaje más activo y basado en la experiencia, lo cual ha sido identificado como un enfoque efectivo para la enseñanza de disciplinas STEM.

En términos de aprendizaje colaborativo, los estudiantes lograron trabajar de manera efectiva en equipos para diseñar y desarrollar proyectos de programación informática con Arduino UNO. Este tipo de actividades grupales no solo fomenta la cooperación, sino que también impulsa la creatividad y la innovación en la solución de problemas reales, aspectos claves según la investigación de Nguyen et al. (2021). Al respecto, Cuásquer-Viveros y Moreno-Cortés (2021), en su estudio sobre diagramas de flujo, sugieren que la práctica constante y el análisis de errores son cruciales para superar estas

barreras y desarrollar una representación algorítmica sólida, coincidiendo con lo observado por Balon et al. (2021) sobre la enseñanza de habilidades computacionales.

En comparación con otros estudios que abordan el uso de herramientas de programación, como Tinkercad y otras plataformas de simulación, los resultados del presente estudio son coherentes al evidenciar un incremento en la motivación y el compromiso de los estudiantes hacia la programación. Por ejemplo, investigaciones recientes como la de Sulayman et al. (2024) encontraron que el uso de simuladores junto con hardware de código abierto permite a los estudiantes experimentar de manera tangible los conceptos aprendidos, mejorando así la comprensión y el aprendizaje autónomo. Esta observación se alinea con los hallazgos de Moncayo et al. (2023) y Tupac-Yupanqui et al. (2022) quienes coinciden en que la experimentación y el enfoque basado en proyectos con Arduino, son cruciales para desarrollar competencias de programación. Estos superan las limitaciones de la programación tradicional al hacer los conceptos más prácticos, lo que resulta en una mejor comprensión del aprendizaje asociado con la electrónica y la programación.

4. Conclusión

La implementación de la programación informática con Arduino UNO en el entorno educativo de secundaria ha demostrado ser una estrategia efectiva para fortalecer el Pensamiento Computacional en los estudiantes. Los resultados indican que esta herramienta no solo facilita el aprendizaje activo de conceptos básicos de programación, sino que también promueve competencias transversales como el trabajo en equipo, la creatividad y la capacidad de resolución de problemas. Sin embargo, las dificultades encontradas en la representación de algoritmos mediante diagramas de flujo sugieren la necesidad de ajustar las estrategias de enseñanza para abordar estas áreas críticas con mayor profundidad. En general, los hallazgos respaldan la idea de que el aprendizaje con tecnologías accesibles como Arduino puede desempeñar un papel crucial en la preparación de los estudiantes para los desafíos de la Industria 4.0 y un mundo cada vez más digitalizado.

Finalmente, los resultados presentados en este artículo ofrecen un punto de partida para futuras investigaciones centradas en la optimización de metodologías de enseñanza y la incorporación de tecnologías emergentes en el currículo de educación secundaria. Además, destacan la importancia de brindar una capacitación adecuada a los docentes para maximizar el impacto de estas herramientas en el desarrollo de competencias digitales y computacionales en los estudiantes.

5. Referencias

- Akib, E.; Muhsin, Muh. A. (2019). Assessment of teaching in 21st century. *Journal of Physics: Conference Series*, 1179(1), 012065. <https://doi.org/10.1088/1742-6596/1179/1/012065>
- Alsina, A.; Acosta, Y. (2022). Conectando la educación matemática infantil y el pensamiento computacional: Aprendizaje de patrones de repetición con el robot educativo programable Cubetto®. *Innovaciones Educativas*, 24(37), 133-148. <https://doi.org/10.22458/ie.v24i37.4022>
- Autodesk, I. (2024). *Tinkercad* [Simulador en línea]. <https://www.tinkercad.com>
- Ávalos Dávila, C.; Arbaiza Lecue, N. Z.; Ajenjo Servia, P. (2021). Calidad educativa y nuevas metodologías de enseñanza-aprendizaje: Retos, necesidades y oportunidades para una visión disruptiva de la profesión docente. *Innovaciones Educativas*, 23(35), 117-130. <https://doi.org/10.22458/ie.v23i35.3477>
- Balon, B.; Duric, J.; Simic, M. (2021). Arduino platform as learning tool in high school and college education. *44th International Convention on Information, Communication and Electronic Technology (MIPRO 2021)*, 688-693. <https://doi.org/10.23919/MIPRO52101.2021.9597141>
- Bianchi, A.; Moon, K. J.; Dementyev, A.; Je, S. (2024). BlinkBoard: Guiding and monitoring circuit assembly for synchronous and remote physical computing education. *HardwareX*, 17, e00511. <https://doi.org/10.1016/j.ohx.2024.e00511>
- Buachoom, A.; Thedsakhulwong, A.; Wutti-prom, S. (2019). An Arduino board with ultrasonic sensor investigation of simple harmonic motion. *Journal of Physics: Conference Series*, 1380(1), 012098. <https://doi.org/10.1088/1742-6596/1380/1/012098>
- Cabezas, C.; Carrera, O. (2024). Enfoque STEM en la educación y formación docente en el Distrito Noroccidente de la Mancomunidad del Chocó Andino. *Mamakuna*, 23, 48-62. <https://revistas.unae.edu.ec/index.php/mamakuna/article/view/946>

- Corrales-Álvarez, M.; Ocampo, L. M.; Cardona-Torres, S. A. (2024). Definiciones del pensamiento computacional. Una revisión de la literatura. *Revista EIA*, 21(42), 1–24. <https://doi.org/10.24050/reia>
- de Educación, S. (2016). *Plan nacional decenal de educación 2016–2026* (1–84). Ministerio de Educación Nacional. https://www.mineducacion.gov.co/1780/articles-392871_recurso_1.pdf
- Deepti Raj, G.; Prabadevi, B.; Gopal, R. (2024). Evolution of Industry 4.0 and its fundamental characteristics. En *Disruptive Technologies and Digital Transformations for Society 5.0* (pp. 1–25). Springer. https://doi.org/10.1007/978-981-99-8118-2_1
- Dimitrova, G.; Marinova, M. (2024). Digital transformation of higher education: Model for digital learning. *AIP Conference Proceedings*, 3078(1), 020053. <https://doi.org/10.1063/5.0208135>
- El Hadi, M.; El Moussaouy, A.; Ouariach, A.; Essaadaoui, R.; Hachmi, A.; Laabidi, K.; Magrez, H.; Meziani, Y. M. (2020). Real time free fall investigation for educational purposes using Arduino Uno board. *Physics Education*, 55(5), 055018. <https://doi.org/10.1088/1361-6552/aba12f>
- Gajdzik, B.; Wolniak, R. (2021). Transitioning of steel producers to the steelworks 4.0 – Literature review with case studies. *Energies*, 14(14), 4391. <https://doi.org/10.3390/en14144109>
- Gobierno de Colombia. Ministerio de Comercio, Industria y Turismo. (2023). <https://www.mincit.gov.co>
- Guevara, G.; Verdesoto, A.; Castro, N. (2020). Metodologías de investigación educativa (descriptivas, experimentales, participativas, y de investigación-acción). *RECIMUNDO*, 4(3), 163–173. [https://doi.org/10.26820/recimundo/4.\(3\).julio.2020.163-173](https://doi.org/10.26820/recimundo/4.(3).julio.2020.163-173)
- Hernández, J.; Díaz, S.; Hernández, G.; León, L. (2024). Implementación de microcontrolador Arduino en prácticas de electrónica digital como estrategia de aprendizaje. *Ciencia Latina Revista Científica Multidisciplinar*, 8(4), 13488–13504. https://doi.org/10.37811/cl_rcm.v8i5.13757
- Hurtuk, J.; Adam, N.; Chovanec, M. (2017). The Arduino platform connected to education process. *IEEE 21st International Conference on Intelligent Engineering Systems (INES)*, 71–76. <https://doi.org/10.1109/INES.2017.8118531>
- Kondaveeti, H. K.; Kumaravelu, N. K.; Vanambathina, S. D.; Mathe, S. E.; Vappangi, S. (2021). A systematic literature review on prototyping with Arduino: Applications, challenges, advantages, and limitations. *Computer Science Review*, 40, 100364. <https://doi.org/10.1016/j.cosrev.2021.100364>
- Kumar, G.; Singh, O. P.; Saini, H. (2021). *Cybersecurity: Ambient Technologies, IoT, and Industry 4.0 Implications* (Vol. 1). CRC Press.

- Lee, S. J.; Kwon, K. (2024). A systematic review of AI education in K-12 classrooms from 2018 to 2023: Topics, strategies, and learning outcomes. *Computers and Education: Artificial Intelligence*, 6, 100211. <https://doi.org/10.1016/j.caeai.2024.100211>
- Manikanawar, S.; Kulkarni, V. N.; Gaitonde, V. N.; Satish, G. J.; Kotturshettar, B. B. (2021). A state-of-the-art review on Industry 4.0 and related key technologies. *AIP Conference Proceedings*, 2358(1), 020006. <https://doi.org/10.1063/5.0057994>
- Marín-Marín, J. A.; García-Tudela, P. A.; Duo-Terrón, P. (2024). Computational thinking and programming with Arduino in education: A systematic review for secondary education. *Heliyon*, 10(8), e29177. <https://doi.org/10.1016/j.heliyon.2024.e29177>
- Mercorelli, P.; Nemati, H.; Zhu, Q. (2024). Industry 4.0 more than a challenge in modeling, identification, and control for cyber-physical systems. En *Modeling, Identification, and Control for Cyber-Physical Systems Towards Industry 4.0*. Elsevier. <https://doi.org/10.1016/B978-0-32-395207-1.00010-X>
- Moncayo, H.; Alulima, L.; Mena, M.; Zárate, L. (2023). Empowering electronics learning through a hands-on approach: Arduino projects in the classroom. *IEEE 3rd International Conference on Advanced Learning Technologies on Education & Research (ICALTER)*, 1–4. <https://doi.org/10.1109/ICALTER61411.2023.10372878>
- Montiel, H.; Gomez-Zermeño, M. G. (2021). Educational challenges for computational thinking in K–12 education: A systematic literature review of “Scratch” as an innovative programming tool. *Computers*, 10(6), 71. <https://doi.org/10.3390/computers10060069>
- Narong, D. K.; Hallinger, P. (2024). Traversing the evolution of research on engineering education for sustainability: A bibliometric review (1991–2022). *Sustainability (Switzerland)*, 16(2), 6401. <https://doi.org/10.3390/su16020641>
- Nguyen, T.-T.; Hoang, D.-L.; Nguyen, H.-T. L.; Nguyen, T.-B. (2021). STEM-oriented activities for improving student performance in Chu Van An secondary school, Thai Nguyen Province, Vietnam. *Journal of Physics: Conference Series*, 1835(1), 012053. <https://doi.org/10.1088/1742-6596/1835/1/012053>
- Rodriguez-Sanchez, C.; Orellana, R.; Fernandez, P. R.; Borromeo, S.; Vaquero, J. (2024). Insights 4.0: Transformative learning in industrial engineering through problem-based learning and project-based learning. *Computer Applications in Engineering Education*, 32(4), e22736. <https://doi.org/10.1002/cae.22736>
- Ruiz-Bolívar, C.; Ríos-Cabrera, P. (2020). La innovación educativa en América Latina: Lineamientos para la formulación de políticas públicas. *Innovaciones Educativas*, 22(32), 199–212. <https://doi.org/10.22458/ie.v22i32.2828>

- Soenarto, S.; Sugito, S.; Suyanta, S.; Siswantoyo, S.; Marwanti, M. (2020). Vocational and senior high school professional teachers in Industry 4.0 | Guru profesional di sekolah menengah atas dan sekolah menengah kejuruan di era industri 4.0. *Cakrawala Pendidikan*, 39(3), 655–665. <https://doi.org/10.21831/cp.v39i3.32926>
- Sulayman, A. A.; Araromi, D. O.; Ayodele, O. E.; Araromi, H. O.; Osuolale, F. N. (2024). Arduino microcontroller based real-time monitoring of haemodialysis process for patients with kidney disease. *E-Prime - Advances in Electrical Engineering, Electronics and Energy*, 7, 100403. <https://doi.org/10.1016/j.prime.2023.100403>
- Tupac-Yupanqui, M.; Vidal-Silva, C.; Pavesi-Farriol, L.; Sanchez Ortiz, A.; Cardenas-Cobo, J.; Pereira, F. (2022). Exploiting Arduino features to develop programming competencies. *IEEE Access*, 10, 20602–20615. <https://doi.org/10.1109/ACCESS.2022.3150101>
- Udvaros, J.; Gubán, M.; Gubán, Á.; Sándor, Á. (2023). Industry 4.0 from the perspective of Education 4.0. *International Journal of Advanced Natural Sciences and Engineering Researches*, 7(4), 230–234. <https://doi.org/10.59287/ijanser.705>
- Umbara, D. M. A. (2024). Revolutionizing Education 4.0: Evaluating the role of robots in learning effectiveness. *E3S Web of Conferences*, 482, 05011. <https://doi.org/10.1051/e3sconf/202448205011>