

**SISTEMATIZACIÓN DE UNA EXPERIENCIA DE APRENDIZAJE PARA LA
INTRODUCCIÓN AL PENSAMIENTO ALGORÍTMICO A TRAVÉS DEL USO
TARJETAS PROGRAMABLES EN UN AMBIENTE MAKER EN LA IE DON
BOSCO POPAYÁN**

JOHAO DANIEL CALDÓN MAZABUEL

UNIVERSIDAD ICESI

MAESTRÍA EN EDUCACIÓN MEDIADA POR LAS TIC

ESCUELA DE CIENCIAS DE LA EDUCACIÓN

SANTIAGO DE CALI

2023

Sistematización de una experiencia de aprendizaje para la introducción al pensamiento algorítmico a través del uso tarjetas programables en un ambiente maker en la IE Don Bosco Popayán

JOHAO DANIEL CALDÓN MAZABUEL

MAESTRÍA EN EDUCACIÓN MEDIADA POR LAS TIC

TUTOR DEL PROYECTO

JORGE ALBERTO QUESADA HURTADO

UNIVERSIDAD ICESI

ESCUELA DE CIENCIAS DE LA EDUCACIÓN

SANTIAGO DE CALI

2023

Agradecimientos

Quiero expresar mi más profundo agradecimiento a aquellas personas que han sido parte fundamental en este proceso académico. En primer lugar, agradezco de todo corazón a mis padres, Ariel y Esperanza, quienes han sido mi fuente de inspiración y ejemplo de disciplina y esfuerzo. Su inquebrantable apoyo y creencia en la educación como herramienta para el crecimiento personal me han motivado a esforzarme al máximo en este reto académico.

No puedo pasar por alto el agradecimiento a Andrea, su cariño, su ejemplo de tenacidad y sus palabras de aliento han sido mi mayor motivación para perseverar a pesar de los desafíos.

También deseo expresar mi gratitud a todos los estudiantes de la Institución Educativa Don Bosco en Popayán. Su compromiso y participación activa en este proceso han hecho que la experiencia de ser docente sea verdaderamente maravillosa. Cada día de enseñanza ha sido un aprendizaje mutuo, y su curiosidad y dedicación me han impulsado a esforzarme por ser un mejor educador.

Tabla de contenido

1. Introducción	7
2. Fase 1. Alistamiento: Identificación, precisión y contextualización histórico-situada de la práctica educativa objeto de la sistematización y sus actores.	8
2.1. Identificación y precisión de la práctica educativa.	8
2.2. Contextualización histórico-situada de la práctica educativa.....	9
2.3. Caracterización de los actores	10
3. Fase 2. Diseño del Proyecto: El Problema de la sistematización	11
3.1. Identificación del problema de la sistematización.....	11
3.2. Formulación de la pregunta de la sistematización.....	12
3.3. Justificación de la sistematización.	12
3.4. Objetivo de la sistematización.....	13
3.5. Ejes de la sistematización.....	13
3.5.1. Eje	13
3.5.2. Sub ejes	13
4. Marco analítico.....	14
4.1. Inclusión de las TIC en ambientes educativos	14
4.2. Pensamiento algorítmico	16
4.3. Enseñanza de la programación computacional en la escuela	18
4.3.1. Actividades desconectadas	20
4.3.2. Gamificación	20
4.3.3. Robótica educativa	21
4.3.4. Aprendizaje móvil.....	21
4.4. Espacios creativos (MakerSpaces)	22
4.5. Micro:bit.....	25
5. Fase 3. Recuperación y reconstrucción de la práctica educativa.....	37
5.1. Diseño metodológico.....	37
5.2. Recuperación y reconstrucción de la práctica	40
5.2.1. Diseño	40
5.2.2. Implementación.....	56
6. Fase 4. El análisis, interpretación y reflexión del relato producto de la recuperación y reconstrucción de la práctica.	75

6.1. Uso de plataformas de programación en bloques y dispositivos interactivos para el desarrollo del pensamiento algorítmico como estrategia de aprendizaje en una fase introductoria.	75
6.2. ¿Cuáles son las actividades implementadas por el docente que benefician el desarrollo del pensamiento algorítmico?	77
6.3. ¿Cómo se puede evidenciar el uso de las TIC como elementos de mediación en el aprendizaje del pensamiento algorítmico?	79
7. Conclusiones	82
8. Referencias bibliográficas	87

Índice de figuras

Figura 1. Dimensiones del aprendizaje maker.	23
Figura 2. Características de la tarjeta Micro:bit.	26
Figura 3. Listado de presaberes.....	42
Figura 4. Interfaz de Scratch.	44
Figura 5. Tarjeta Arduino Uno.....	44
Figura 6. Tarjeta Micro:bit v2.	45
Figura 7. Demostración del juego de la “culebrita”.	57
Figura 8. Evidencia de evaluación diagnóstica.	58
Figura 9. Resultados evaluación diagnóstica.	59
Figura 10. Interfaz inicial de la plataforma MakeCode.....	60
Figura 11. Interfaz de trabajo en la plataforma MakeCode.....	61
Figura 12. Evidencia de implementación.	62
Figura 13. Bloques de código para el algoritmo del juego de la “culebrita”.....	64
Figura 14. Evidencia de implementación.	66
Figura 15. Programación de un prototipo de candado digital.	68
Figura 16. Evidencia de preparación de la exposición Tecno Bosco.	70
Figura 17. Evidencia de la exposición Tecno Bosco.	72
Figura 18. Resultados evaluación formativa.	74
Figura 19. Resultados autoevaluación desempeños.	74
Figura 20. Resultados autoevaluación intereses.....	75

Índice de tablas

Tabla 1. Síntesis de las sistematizaciones consultadas.....	28
Tabla 2. Diseño metodológico.	37
Tabla 3. Plan de clase.....	47

1. Introducción

La educación es un campo en constante evolución, y como docentes, estamos comprometidos en proporcionar a nuestros estudiantes experiencias de aprendizaje significativas y efectivas. Para lograr este objetivo, nuestras prácticas educativas deben ser sometidas a un proceso continuo de autoevaluación y reflexión. Este proceso nos permite comprender a fondo los actores involucrados en el proceso de enseñanza-aprendizaje, los elementos que influyen en él y cómo se relacionan entre sí. Además, nos brinda la oportunidad de reconocer y evaluar el impacto de los modelos pedagógicos y enfoques didácticos que aplicamos en el aula.

La importancia de este proceso de reflexión no puede subestimarse, ya que es fundamental para el crecimiento y la mejora constante de nuestra labor docente. Como afirma Jara (2020), esta reflexión puede servir como un punto de partida para identificar áreas de mejora, adaptar nuestras estrategias pedagógicas y, en última instancia, transformar nuestra práctica educativa en beneficio de nuestros estudiantes.

En este documento de sistematización, exploraremos en detalle cómo esta reflexión profunda ha influido en mi práctica docente durante un período específico. Examinaré las estrategias pedagógicas que he empleado, los resultados obtenidos y cómo estas experiencias han moldeado mi labor. A través de este análisis, espero arrojar luz sobre la eficacia de ciertas metodologías y proporcionar ideas valiosas para futuras mejoras en mi enseñanza.

2. Fase 1. Alistamiento: Identificación, precisión y contextualización histórico-situada de la práctica educativa objeto de la sistematización y sus actores.

2.1. Identificación y precisión de la práctica educativa.

La experiencia propone el desarrollo de habilidades del pensamiento algorítmico mediante actividades desarrolladas en las salas de cómputo y laboratorios de la institución educativa, haciendo uso de computadores con acceso a internet y tarjetas de desarrollo. Las tarjetas Micro:bit son pequeñas computadoras programables que se utilizan para enseñar a los estudiantes los conceptos básicos de la programación.

Realizar la introducción a la programación utilizando tarjetas Micro:bit es una práctica educativa innovadora y divertida que permite a los estudiantes aprender los fundamentos de la programación de manera visual y práctica, experiencia educativa que ha cobrado gran popularidad en las escuelas de todo el mundo debido a sus numerosos beneficios. En primer lugar, es una forma divertida y motivadora de aprender programación, lo que hace que los estudiantes estén más comprometidos y dispuestos a aprender. En segundo lugar, enseña a los estudiantes habilidades importantes como la resolución de problemas, el pensamiento crítico y la colaboración en equipo. Finalmente, les proporciona a los estudiantes una base sólida para continuar su educación en programación en el futuro.

La práctica educativa consiste en generar espacios y dotar de recursos a los estudiantes para que de manera creativa diseñen, codifiquen y prueben sus propios programas. En una primera sesión, se presenta la tarjeta y los estudiantes exploran sus funcionalidades reconociendo sus características técnicas. Luego, se conceptualizan

elementos básicos de la programación, como la lógica y las variables. Observando y modificando ejemplos los estudiantes aprenden a escribir código en un lenguaje de programación de bloques, que es una forma visual y simplificada de la programación.

Una vez que los estudiantes comprenden los conceptos básicos, se les motiva a crear una solución en la que puedan aplicar lo que han aprendido, los estudiantes trabajan en equipo y se les anima a compartir sus ideas y soluciones con sus compañeros.

2.2. Contextualización histórico-situada de la práctica educativa.

La ciudad de Popayán se encuentra ubicada en la zona centro del departamento del Cauca limitada con los municipios de Cajibío, Totoró, Puracé, Sotaró y el Tambo, cuenta con una población de 318000 habitantes aproximadamente y es reconocida a nivel internacional por sus edificios coloniales encalados y por ser un centro religioso con procesiones populares en Semana Santa.

Su población mestiza es predominante, proceso que se acentúa por la interrelación étnica debido al flujo de migraciones de los diferentes municipios del departamento. Esto hace que la ciudad crezca desde la perspectiva demográfica originando en la actualidad riesgos sociales como el desempleo y la delincuencia entre otros.

La institución educativa Don Bosco Popayán es un colegio de carácter público con modalidad técnica cuya misión es “la formación integral para educar, evangelizar y capacitar a niños y jóvenes orientando proyectos de vida fundamentalmente en los principios y valores del sistema preventivo con una espiritualidad salesiana. Formamos buenos cristianos y honestos ciudadanos.”

La comunidad educativa es animada espiritualmente y dirigida estratégicamente por miembros (sacerdotes y laicos consagrados) de la comunidad salesiana en Colombia desde el año 1961 quienes, como su fundador, San Juan Bosco se proponen “ser signos y portadores del amor de Dios a los jóvenes, especialmente a los más pobres”.

La IE desarrolla un proyecto de formación técnica que permite a estudiantes de educación media graduarse con dos títulos: el primero que los acredita como bachilleres y el segundo como técnicos, esto permite fortalecer sus competencias laborales y les ofrece oportunidades en el sector productivo del país. Las especialidades técnicas ofertadas en la actualidad son metalmecánica, electricidad, ebanistería, mantenimientos de equipos de cómputo y programación de aplicaciones.

2.3. Caracterización de los actores

En la experiencia de aprendizaje se pueden identificar dos tipos de actores, ambos participan de manera activa en la misma.

Un grupo de 32 estudiantes del grado décimo, que han optado por la especialidad técnica de programación de software, cuyas edades oscilan entre los 13 y 15 años de edad. Estos estudiantes asumen el rol en el contexto del proceso de sistematización de sujetos sobre los cuales se aplica el proceso de enseñanza, lo que significa que no son meros receptores pasivos de información, sino que desempeñan un papel activo en la construcción de su propio conocimiento.

El docente del área de tecnología e informática, con una formación profesional en ingeniería en electrónica y telecomunicaciones y una experiencia en la labor docente de 8 años, asume dos roles dentro del proceso, es el mediador en la enseñanza y aprendizaje para

lo cual diseña, implementa y evalúa las actividades propias de la experiencia, pero a su vez realiza una observación crítica de su propia práctica.

3. Fase 2. Diseño del Proyecto: El Problema de la sistematización

3.1. Identificación del problema de la sistematización.

La institución educativa Don Bosco, de acuerdo a su PEI desarrolla en sus actividades académicas una modalidad técnica, por lo cual los estudiantes al iniciar el nivel décimo realizan la selección de una especialidad. El grupo de estudiantes, con quienes se desarrolla la experiencia de aprendizaje, pertenecen al técnico en programación de software.

El principal objetivo de aprendizaje en la asignatura es desarrollar habilidades para dar solución a problemas utilizando dispositivos de cómputo. La consecución de este objetivo requiere diseñar actividades que no solo se limiten a la enseñanza de un lenguaje de programación en específico sino que le permitan al estudiante aplicar elementos del pensamiento computacional, como la abstracción, reconocimiento de patrones, descomposición, la validación de soluciones, pensamiento algorítmico, entre otros, para estimular su creatividad, potenciar sus habilidades comunicativas y matemáticas y practicar de manera efectiva el aprendizaje colaborativo.

Alcanzar este objetivo requiere por parte del docente diseñar de un entorno de aprendizaje que no se limite a la presentación teórica de conceptos, sino que mediante la construcción de prototipos los estudiantes puedan descubrir e interiorizar la aplicabilidad de las ciencias de la computación. En este contexto, es esencial que las actividades promuevan el

pensamiento crítico y creativo de los estudiantes utilizando medios interactivos para mediar el aprendizaje fomentando la colaboración y el trabajo en equipo.

3.2. Formulación de la pregunta de la sistematización.

¿De qué manera se beneficia el desarrollo del pensamiento algorítmico en los estudiantes del grado décimo del técnico en programación de software de la IE Don Bosco Popayán al usar la tarjeta programable Micro:bit durante la fase de introducción al curso?

3.3. Justificación de la sistematización.

Teniendo en cuenta que la experiencia objeto del proceso de sistematización, se ha implementado durante los últimos 3 años y que existe la necesidad y oportunidad de replicarla nuevamente en igual contexto educativo, se plantean los siguientes argumentos que justifican la selección:

- El proceso de sistematización planeado permite evidenciar posibles actividades, de las desarrolladas durante la experiencia, para ser objeto de transformación a fin de mejorar el aprendizaje de los estudiantes.
- La recopilación de información y su posterior análisis permitirá destacar los elementos que los estudiantes consideran más cercanos o propicios para sus contextos.
- El análisis crítico de la experiencia permite construir elementos para valorar la metodología ejecutada vs la metodología recomendada, es decir identificar aquellas actividades en las cuales se ha innovado frente a las actividades propuestas en las metodologías, para la enseñanza de los algoritmos computacionales.

- Tener la posibilidad de presentar el proceso de sistematización a la comunidad educativa.
- Compartir los resultados y conclusiones con colegas en un evento académico.
- Motiva el proceso continuo de aprendizaje en torno a la profesión docente.

3.4. Objetivo de la sistematización

Establecer la manera en que el uso de la tarjeta Micro:bit beneficia el desarrollo del pensamiento algorítmico en la introducción a la lógica de programación a los estudiantes del grado décimo del técnico en sistemas de la IE Don Bosco Popayán.

3.5. Ejes de la sistematización.

3.5.1. Eje

Uso de plataformas de programación en bloques y dispositivos interactivos para el desarrollo del pensamiento algorítmico como estrategia de aprendizaje en una fase introductoria.

3.5.2. Sub ejes

¿Cuáles son las actividades implementadas por el docente que benefician el desarrollo del pensamiento algorítmico?

¿Cómo se puede evidenciar el uso de las TIC como elementos de mediación en el aprendizaje del pensamiento algorítmico?

4. Marco analítico

4.1. Inclusión de las TIC en ambientes educativos

Mencionar la dependencia que la humanidad ha desarrollado frente a los medios tecnológicos para garantizar su desarrollo y supervivencia es afirmar una verdad absoluta, resulta imposible en la actualidad encontrar aspectos del diario vivir en los cuales no exista al menos una tecnología digital para mediar y facilitar las interacciones.

Como menciona Soza (2020), integrar las TIC en una clase demanda un esfuerzo consciente de los docentes ya que requiere activar e integrar una serie de saberes y habilidades para lograr concretar un uso significativo. Las políticas educativas de TIC en nuestro país definidas para dotar a las instituciones educativas de dispositivos de cómputo y desplegar la red de internet en la mayoría del territorio nacional, incluyen según Rueda y Franco (2018) el Programa Compartel de Telefonía Social implementado entre los años 1999 y 2001, Agenda de conectividad en el año 2000, Plan Estratégico de Educación de Colombia para los años 2000 y 2002, Plan Nacional de Desarrollo 2003 - 2006, Plan Vive Digital 2014-2018, Computadores para educar, entre otras.

Al realizar un paralelismo entre los recursos económicos asignados y los resultados en pruebas externas se puede concluir como indica Sosa (2020) que no existe una relación directa entre el acceso o tiempo de uso de las TIC y la calidad educativa, ya que entre otros factores existe una deficiencia en la formación de los docentes y directivos docentes en el uso pedagógico de los mismos.

La relativa facilidad de acceso, la sobre demanda de recursos tecnológicos como lo presenta Espitia, Sánchez & Cano (2022), y la necesidad propia de actualización ha generado un constante movimiento en los ambientes educativos para generar estrategias y

modelos que guíen la inclusión de las TIC permitiendo mantener un equilibrio entre su uso pertinente y otras formas de enseñanza y aprendizaje. Y es que el uso de las TIC en educación pueden ser una herramienta valiosa para facilitar el aprendizaje y mejorar la calidad de la educación, siempre y cuando se utilicen de manera efectiva y se integren adecuadamente en los procesos educativos. Por ello, es fundamental que los docentes estén capacitados para utilizar estas tecnologías y cerrar las brechas de acceso para que los estudiantes puedan beneficiarse de sus posibilidades.

Ante este panorama se presenta el modelo teórico TPACK formulado en el año 2006 por Punya Mishra y Matthew J. Koehler, el cual constituye un marco conceptual para describir el conocimiento que los docentes deben tener para integrar efectivamente la tecnología en los procesos de enseñanza y aprendizaje. Dicho modelo TPACK se enfoca en la intersección de tres tipos de conocimiento:

- **Conocimiento pedagógico:** Este tipo de conocimiento se refiere a la comprensión que tienen los docentes sobre cómo enseñar de manera efectiva, incluyendo la planificación de lecciones, la gestión del aula y la evaluación del aprendizaje.
- **Conocimiento del contenido:** Este tipo de conocimiento se refiere al conocimiento que los docentes tienen sobre el tema que están enseñando. Incluye tanto conocimientos generales sobre el tema como conocimientos específicos sobre cómo enseñar ese tema.
- **Conocimiento tecnológico:** Este tipo de conocimiento se refiere a la comprensión que tienen los docentes sobre cómo utilizar las herramientas tecnológicas de manera efectiva en el aula. Incluye tanto conocimientos técnicos sobre cómo utilizar las

herramientas como conocimientos pedagógicos sobre cómo integrarlas en la enseñanza.

En síntesis, el modelo TPACK afirma que para integrar efectivamente la tecnología en la educación, los docentes necesitan tener un conocimiento equilibrado de estos tres tipos de conocimiento. Esto significa que los docentes deben comprender cómo la tecnología puede ayudar a mejorar la enseñanza y el aprendizaje en su materia específica, y deben tener las habilidades y conocimientos necesarios para utilizarla de manera efectiva en el aula (Sosa, 2020).

Como concluye Enríquez (2023) en su artículo de revisión de literatura, en la mejora de los procesos de enseñanza y aprendizaje la capacitación de los docentes juega un papel preponderante en aspectos como competencias digitales y la capacidad para integrar adecuadamente la TIC en el aula además se valida que, en la actualidad de acuerdo a los artículos revisados, las TIC son relevantes en distintos niveles, desde infantil hasta educación superior.

4.2. Pensamiento algorítmico

El pensamiento algorítmico (PA) es la habilidad para resolver problemas a través de la creación y aplicación de algoritmos, es decir, una secuencia lógica y estructurada de instrucciones que llevan a cabo una tarea específica. Esta habilidad implica la capacidad de analizar un problema, identificar los pasos necesarios para resolverlo, ordenarlos de manera lógica y secuencial, y luego traducirlos en un lenguaje de programación. El PA como comenta Kožuh et al (2018), es una de las habilidades fundamentales del siglo XXI dado el nivel de inclusión de la informática en todos los campos de la vida.

En el campo educativo, ha cobrado relevancia la inclusión del PA en el currículum académico para ayudar a los estudiantes a desarrollar habilidades de resolución de problemas y pensamiento crítico por ejemplo a través de actividades que involucren la creación de algoritmos y el desarrollo de soluciones lógicas a problemas concretos. Al enseñar pensamiento algorítmico, se busca que los estudiantes aprendan a descomponer problemas complejos en pasos más pequeños y manejables, crear soluciones paso a paso, identificar patrones y establecer reglas. (M. Stephens y Kadjevich, 2020), siendo estas habilidades esenciales para la programación de computadoras y otras áreas de las TIC. De acuerdo a Pérez, González & Palacio (2020), la enseñanza del pensamiento algorítmico también puede ayudar a los estudiantes a desarrollar habilidades importantes para la vida, como la planificación, la organización y la toma de decisiones.

Teniendo en cuenta que el pensamiento computacional PC como lo define Angeli & Giannakos (2020), es un término para referirse a un conjunto de habilidades mentales y procesos cognitivos utilizados para resolver problemas complejos de manera eficiente y efectiva mediante la aplicación de conceptos y técnicas de la ciencia de la computación es posible plantear los siguientes elementos diferenciadores frente al PA:

- El PA se centra en el diseño, aplicación y validación de algoritmos para resolver problemas, mientras que el PC implica una serie de habilidades más amplias, incluyendo la resolución de problemas, el diseño de sistemas y la programación.
- El PA es un proceso cognitivo, mientras que el PC implica el uso de herramientas y técnicas específicas en un contexto computacional.

- El PA puede aplicarse en cualquier área de la vida, no necesariamente relacionado con la tecnología, mientras que el PC se enfoca en la solución de problemas mediante el uso de herramientas y técnicas de la computación.

4.3. Enseñanza de la programación computacional en la escuela

Dada la constante evolución y transformación del mundo, impulsado por el desarrollo tecnológico y la globalización, en la actualidad, es cada vez más importante que en las escuelas se enseñe un currículo que prepare a los estudiantes para el futuro y no solo para el pasado. En este sentido, las habilidades y conocimientos que los estudiantes necesitarán en el futuro serán diferentes a los que se han enseñado en el pasado, se espera que sean capaces de resolver problemas complejos, trabajar en equipo, comunicarse de manera efectiva, tener habilidades digitales y estar preparados para adaptarse a los cambios constantes en el mercado laboral. (Partovi, 2020).

Por lo tanto, las escuelas deben adaptar su currículo y sus métodos de enseñanza en los cuales se privilegie el desarrollo de habilidades y conocimientos necesarios para el futuro. Esto implica una educación más orientada al pensamiento crítico, la resolución de problemas, la creatividad y la colaboración, así como una mayor integración de la tecnología y el aprendizaje práctico en las aulas.

Como muestra de esta transformación en los sistemas educativos, está la inclusión de la programación informática en los planes curriculares, iniciando en los países desarrollados y llegando a ser una tendencia a nivel mundial. Esta rápida adopción por la programación se encuentra justificada, no solo porque prepara a los trabajadores para los

futuros empleos en los cuales se requerirá crear y programar máquinas, sino que mucho más importante da la capacidad de desarrollar en los estudiantes habilidades cognitivas como indica Pedro Jose & Brigas (2022) que lo guíen hacia un rol en el cual reflexione sobre su propio proceso de pensamiento. Seymour Papert, pionero en la inclusión de la programación en la escuela en la década del 60, identificó en la enseñanza de lenguajes de programación una plataforma para generar ideas poderosas y una herramienta para involucrar a los estudiantes en nuevas formas de pensar y reflexionar. (Papert, 2005).

A manera de síntesis se puede afirmar que el principal beneficio de iniciar un proceso de enseñanza de la programación informática en las escuelas es que se desarrolla el pensamiento computacional en los estudiantes. Algunos de los argumentos y beneficios presentados por Pedro Jose & Brigas (2022) sobre el desarrollo del pensamiento computacional en los niveles de educación primaria, básica y media son los siguientes:

- Facilita la creación de actividades transversales a diferentes áreas del conocimiento.
- Promueve la equidad de género.
- Permite el diseño de actividades que brindan motivación y satisfacción a los estudiantes.
- Fomenta el trabajo colaborativo.
- Fortalece la creatividad e innovación a través del diseño y construcción de soluciones que utilizan las artes, las ciencias y la tecnología.

Existen diversos enfoques que facilitan la enseñanza de la programación en el ámbito escolar siendo estos los elementos primordiales para consolidar las ciencias de la computación en los entornos educativos. En la revisión de propuestas realizada por Astudillo, Bast, Segovia & Castro (2019) y en las estrategias recomendadas por Pedro Jose

& Brigas (2022), estos enfoques se agrupan en actividades desconectadas, gamificación, robótica y aprendizaje móvil.

4.3.1. Actividades desconectadas

Las actividades desconectadas son aquellas en las cuales no se depende específicamente de un computador para su realización ni tienen como objetivo la enseñanza de un lenguaje de computación específico. Para el desarrollo de este tipo de actividades se hace uso de recursos como juegos o materiales cotidianos para introducir en los estudiantes conceptos propios de la programación. Juegos de mesa como Robot Turtles¹, son recursos válidos dentro de este enfoque.

4.3.2. Gamificación

La gamificación es una estrategia pedagógica que consiste en aplicar técnicas y mecánicas de juego en entornos educativos para mejorar el aprendizaje. En el contexto de la enseñanza de la programación, la gamificación se utiliza para motivar a los estudiantes a aprender y practicar habilidades de programación.

Una forma común de gamificar la enseñanza de la programación es a través de la creación de juegos. Los estudiantes pueden aprender los conceptos básicos de la programación a medida que crean sus propios juegos, algunas plataformas como GameMaker² les permite poner en práctica los conceptos teóricos que han aprendido y ver cómo funcionan en un entorno práctico.

¹ <https://boardgamegeek.com/boardgame/147370/robot-turtles>

² <https://gamemaker.io/es>

También se pueden utilizar juegos educativos en línea para enseñar programación. Estos juegos suelen presentar desafíos de programación en un entorno lúdico y atractivo, lo que puede motivar a los estudiantes a aprender y practicar habilidades de programación de una manera divertida y entretenida como la plataforma Codecombat³.

4.3.3. Robótica educativa

La robótica educativa es una metodología de enseñanza que utiliza robots como herramientas de aprendizaje para mejorar la comprensión de los conceptos de ciencia, tecnología, ingeniería y matemáticas (STEM, por sus siglas en inglés) y desarrollar habilidades de resolución de problemas, pensamiento crítico, colaboración y creatividad en los estudiantes. A través de la construcción y programación de robots, los estudiantes pueden aplicar los conocimientos adquiridos en estas áreas para resolver problemas y crear soluciones innovadoras. La robótica educativa también fomenta el interés de los estudiantes por las carreras STEM y les permite experimentar de forma práctica y lúdica el mundo de la tecnología y la robótica.

4.3.4. Aprendizaje móvil

El aprendizaje móvil (m-learning, por sus siglas en inglés) es una modalidad de aprendizaje que se realiza a través de dispositivos móviles, como smartphones o tablets, y se lleva a cabo en cualquier lugar y en cualquier momento. Esta modalidad de aprendizaje aprovecha las posibilidades que brindan los dispositivos móviles para crear experiencias educativas personalizadas e interactivas, adaptadas a las necesidades y preferencias del estudiante. El aprendizaje móvil puede incluir una amplia variedad de recursos y

³ <https://codecombat.com/>

herramientas, como videos, podcasts, juegos educativos, aplicaciones de aprendizaje de idiomas, entre otros. Esta modalidad de aprendizaje se ha vuelto cada vez más popular en la actualidad, gracias a la penetración de los dispositivos móviles en la sociedad y a la necesidad de tener acceso a la educación en cualquier momento y lugar.

4.4. Espacios creativos (MakerSpaces)

A consecuencia de las políticas gubernamentales, campañas del sector empresarial o esfuerzos de comunidades educativas, se han dotado ciertas instituciones educativas con elementos tecnológicos adicionales a los computadores, como kits de electrónica o robótica, microscopios digitales, tableros interactivos o inclusive impresoras 3D, con la esperanza de incentivar una mejor calidad en los procesos de enseñanza allí ofertados. Este tipo de elementos brindan la posibilidad de hacer o crear algo físicamente, permitiendo como menciona Rosenheck (2020), desarrollar en los estudiantes habilidades que en una clase tradicional no sería posible.

Es en este contexto donde es posible hablar sobre los makerspaces o espacios de creación, lugares físicos, como salones o laboratorios, en los cuales se disponen de una variedad de materiales y recursos para que los estudiantes puedan diseñar y crear sus propios proyectos escolares. Como lo afirma Rayna & Striukova (2021), su principal objetivo es fomentar las habilidades del pensamiento creativo, la gestión de recursos, planificación o autogestión de tareas y el trabajo colaborativo, las cuales corresponden a algunas de las habilidades del siglo XXI.

Y es que los resultados de la educación Maker no solo se limita por los elementos que se dispongan en un lugar, sino que esta es complementada con el enfoque, mentalidad y

la comunidad que se generan alrededor de esta propuesta emergente. “La Educación Maker tiene como objetivo proporcionar a los estudiantes la autonomía necesaria para explorar sus propias ideas y verse a sí mismos como personas que pueden imaginar, crear, construir y resolver problemas” (Rosenheck, 2020).

En la literatura consultada encontramos un marco de referencia en el cual se describen las habilidades que un estudiante inmerso en makerspace puede llegar a desarrollar, presenta elementos que sirven de guía para el diseño de actividades de aprendizaje y evaluación, además de parámetros para la reflexión sobre cómo las actividades facilitaron o no el aprendizaje. El marco de referencia “Dimensiones del Aprendizaje del Making y el Tinkering” desarrollado por The Tinkering Studio Exploratorium, se presenta en la figura 1.

Figura 1. Dimensiones del aprendizaje maker.



Fuente: Recuperado de Exploratorium. (s.f.). Learning Dimensions: Making and Tinkering.

En el proceso de exploración y análisis de la teoría disponible, se descubrió un listado de aspectos esenciales para la educación maker definidos por Brahms & Wardri (2014). Los autores identificaron una lista de principios de diseño que son fundamentales para desarrollar prácticas de enseñanza efectivas dentro de este enfoque pedagógico. Estos principios se organizan en tres categorías distintas: diseño del espacio, diseño de las actividades y diseño de la instrucción.

Diseño del espacio: Este primer conjunto de principios se centra en la configuración física y el ambiente en el que se lleva a cabo el aprendizaje maker. Implica la disposición de herramientas, materiales y espacios de trabajo de manera que fomenten la creatividad, la experimentación y la colaboración. El diseño del espacio debe ser flexible y adaptable, lo que permite a los estudiantes explorar y crear de manera autónoma.

Diseño de las actividades: Los principios relacionados con el diseño de las actividades se refieren a la planificación de tareas y desafíos que involucran a los estudiantes en procesos de creación y resolución de problemas. Estas actividades deben ser auténticas, significativas y abiertas, lo que significa que permiten múltiples enfoques y soluciones. Deben promover la curiosidad y el pensamiento crítico, y brindar a los estudiantes la oportunidad de aplicar sus habilidades en contextos del mundo real.

Diseño de la instrucción: La tercera categoría de principios se centra en cómo se guía y apoya a los estudiantes en su aprendizaje maker. Esto implica la creación de un entorno de aprendizaje que fomente la autonomía y la autorregulación. Los educadores deben actuar como facilitadores o guías, brindando orientación y retroalimentación en lugar de instrucciones directas. La instrucción debe ser personalizada y adaptada a las necesidades e intereses individuales de los estudiantes.

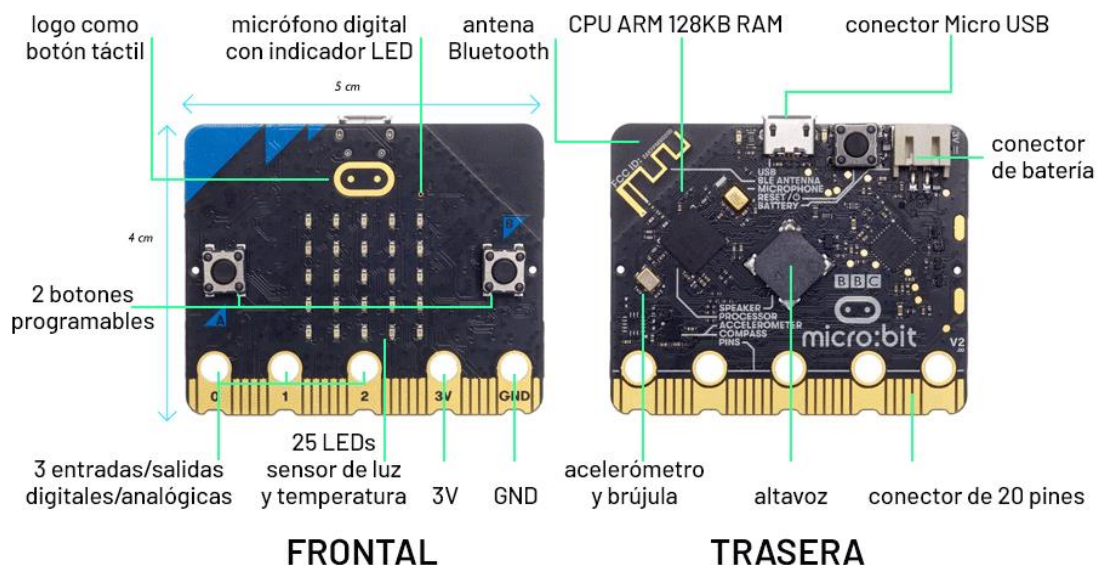
La implementación de una estrategia de makerspace se puede desarrollar en 3 sencillos pasos como menciona Krueger (2022):

- Definir un espacio: Se trata simplemente de formalizar un lugar específico en el cual se va a desarrollar las actividades, no tiene por qué ser un lugar diferente al salón de clase.
- Equipar con recursos: De acuerdo a los objetivos planteados se requiere contar con ciertos elementos, desde materiales para manualidades hasta robots, inicia con lo que ya tienes disponible.
- Invitar a los estudiantes a participar: Aunque es útil desarrollar un plan de clase, también es valioso permitir a los estudiantes la exploración libre.

4.5. Micro:bit

BBC Micro:bit es una pequeña tarjeta programable de 4x5 cm, diseñada para que aprender a programar sea fácil, divertido y al alcance de todos. Tanto el hardware como el software de Micro:bit son de código abierto. Gracias a la gran cantidad de sensores que incorpora, sólo con la tarjeta se pueden llevar a cabo centenares de proyectos. BBC micro:bit también es una plataforma IoT (Internet of Things), lo que la hace muy interesante para usuarios avanzados.

Figura 2. Características de la tarjeta Micro:bit.



Fuente: Recuperado de Wikipedia.

Tiene un entorno de programación gráfico propio: Make Code de Microsoft, un sencillo editor gráfico online muy potente y gratuito que posibilita la introducción en el mundo de la programación de forma intuitiva a través del lenguaje de programación visual o de bloques.

Como herramienta TIC diseñada para su uso en los ambientes educativos, la tarjeta Micro:bit proporciona un medio cognitivo que facilita la enseñanza de la programación al transformar actividades abstractas en actividades lúdicas de exploración y creación además, contribuye al fortalecimiento de la alfabetización digital en los estudiantes generando interacciones que elevan el interés y voluntad de los estudiantes. (Simović, Veskovíc & Purenovic (2022).

Las conclusiones del uso de la tarjeta programable Micro:bit en educación presentadas por Voštinár & Knežník (2020) son:

- Se identificó que la inclusión del Micro:bit tiene una gran influencia en la motivación y actitud de los estudiantes hacia la informática.
- La tarjeta Micro:bit no solo está dirigida para desarrollar el pensamiento computacional de los estudiantes, sino que su uso ayuda a explorar en los docentes nuevos enfoques didácticos de enseñanza, como el trabajo colaborativo, la programación por parejas o aprendizaje basado en problemas.
- Se requieren docentes que hayan desarrollado sus habilidades digitales para que puedan guiar el aprendizaje evitando que la tarjeta se convierta luego de la fase de exploración en un elemento hardware aburrido.

4.6. Revisión de otras sistematizaciones.

Teniendo en cuenta que uno de los objetivos que busca el proceso de sistematización es comprender la propia práctica educativa, identificando fortalezas y posibles debilidades, se hace necesario definir algunas experiencias que sirvan como referentes para la reflexión crítica de la práctica seleccionada.

La consulta de información sobre procesos de sistematización resulta relevante porque permite identificar buenas prácticas al reconocer casos exitosos en contextos educativos similares, en segundo lugar, permite enriquecer el aprendizaje al conocer los desafíos y oportunidades que otros actores educativos han enfrentado en situaciones similares, lo que puede ayudar a anticipar y prevenir posibles problemas o a encontrar soluciones efectivas. De igual manera la información consultada permite validar los hallazgos propios al generar una estructura de comparación con los resultados de otros

autores, de igual manera ayuda a anticipar posibles mejoras en la propia práctica educativa, permitiendo enfocar los esfuerzos en aquellos aspectos que requieran mayor atención.

Para realizar la consulta de información se llevó a cabo una búsqueda de trabajos de grado de maestría en el catálogo público y bases de datos de la universidad ICESI y utilizando los siguientes términos de búsqueda: “Pensamiento computacional”, “Pensamiento algorítmico”, “Sistematización”, “Programación”, “Scratch”.

Los resultados de esta revisión de información se presentan a continuación:

Tabla 1. Síntesis de las sistematizaciones consultadas.

Autor/es	Temática	Conclusiones
Alzate Sánchez, 2021.	Sistematización de una experiencia de aprendizaje que implementa el Aprendizaje Basado en Retos (ABR) como enfoque didáctico para la enseñanza de análisis estadísticos en la asignatura de matemáticas con alumnos de grado décimo.	La estrategia seleccionada facilitó la apropiación del conocimiento en los estudiantes, el uso del recurso narrativo involucró a los estudiantes en una aventura generando un ambiente de aprendizaje donde la motivación y la creatividad fueron elementos preponderantes. Las herramientas TIC

		<p>incluidas en la experiencia fueron acertadas ya que el proceso de planeación estuvo guiado por el modelo de integración curricular TPACK.</p>
<p>Esquea Arrieta, 2021.</p>	<p>Sistematización de una estrategia de aprendizaje que promueve el desarrollo de las habilidades del pensamiento computacional mediante la inclusión de la plataforma Scratch. La estrategia de aprendizaje en la cual se basa es el Aprendizaje Basado en Problemas (ABP).</p>	<p>El proceso de reflexión y análisis realizado durante la sistematización permitió vislumbrar la transformación de la práctica educativa y confirmar los aciertos en las decisiones pedagógicas, didácticas y tecnológicas. También permitió reconocer el rigor con el cual deben ser planeadas las actividades educativas en pro de favorecer aprendizajes coherentes y contextualizados.</p>

		<p>La plataforma Scratch no solo es una plataforma que contribuye al desarrollo del pensamiento algorítmico, sino que funciona como elemento mediador para fortalecer las estrategias de solución de problemas.</p>
<p>Rivera Morcillo, 2020.</p>	<p>Sistematización de una estrategia de aprendizaje para la enseñanza de lógica y geometría, utilizando la plataforma Scratch fundamentada en el enfoque pedagógico Aprendizaje Basado en Retos (ABR).</p>	<p>El uso de la plataforma Scratch en el desarrollo de la actividad permitió analizar los diferentes niveles de inclusión bajo el modelo SAMR, obteniendo como resultado logros superiores a los esperados ya que se encontraron elementos para justificar el impacto desde la fase de Mejorar hasta Transformar. El enfoque y las herramientas TIC</p>

		<p>seleccionadas generaron aprendizajes adicionales a los saberes propuestos, tales como búsqueda de información, habilidades computacionales y sociales.</p>
<p>Ayala Rodríguez, 2022.</p>	<p>Sistematización de una experiencia de aprendizaje en la cual se fomenta el desarrollo del pensamiento computacional con el objetivo de potencializar la expresión oral en estudiantes de grados cuarto y quinto de la básica primaria.</p>	<p>Dejando de lado la estrategia de aprendizaje implementada y las herramientas TIC presentadas, el elemento sobresaliente que les permitió a los estudiantes lograr los aprendizajes esperados fue la contextualización en situaciones cotidianas de las actividades presentadas. La transformación de las prácticas educativas mediante la inclusión de las TIC demanda del docente</p>

		<p>una actitud humilde, en el sentido que se requiere en primer lugar reconocer el desconocimiento en el uso de tecnologías emergentes y en la aplicación de las pedagogías emergentes.</p>
<p>Mondragón Reyes, 2019.</p>	<p>Sistematización de una experiencia de aprendizaje que implementa la Gamificación como enfoque didáctico para el desarrollo de habilidades del pensamiento algorítmico.</p>	<p>La reflexión sobre la práctica educativa implementada permite reconocer la importancia del diagnóstico para dar cuenta de los presaberes y adicionalmente de las características sociales y personales de los estudiantes para así establecer objetivos de aprendizaje que se ajusten a la realidad.</p> <p>Las herramientas TIC facilitan la construcción de</p>

		<p>ambientes de aprendizaje lúdicos que elevan los niveles de atención y motivación en los estudiantes.</p> <p>La implementación de la Gamificación demanda una mayor preparación por parte del docente ya que se requiere una mayor experticia para trasladar las fases de un juego al ambiente educativo.</p>
<p>Daza Alvarado & Sánchez Fernández, 2022.</p>	<p>Sistematización de una experiencia de aprendizaje para el desarrollo de habilidades del pensamiento algorítmico mediante el uso de una plataforma de programación visual o en bloques.</p>	<p>Tras la implementación se concluye que la estrategia de Aprendizaje Basado en Retos y la plataforma Scratch conforman una excelente pareja de recursos que permiten aprovechar el potencial cognitivo.</p> <p>El ABR propicia ambientes</p>

		educativos para desarrollar la investigación y el trabajo colaborativo.
Andersen & Rustad, 2022.	Un estudio que motiva el uso de videojuegos, considerados instrumentos que aportan poco en los ambientes educativos, como medios para desarrollar actividades de aprendizaje significativo y colaborativo.	Reconocer el potencial del uso en educación de herramientas TIC para el desarrollo de habilidades del siglo 21. Se observa que al usar el video juego Minecraft para resolver actividades matemáticas permite el surgimiento del aprendizaje colaborativo.
Eteng, Akpotuzor, Akinola, & Agbonlahor, 2022.	Informe sobre la investigación y análisis de metodologías recomendadas para la enseñanza de la programación de software a estudiantes de países en vía de desarrollo que inician su educación universitaria.	El estudio recomienda diferentes lenguajes de programación para la enseñanza de acuerdo al nivel educativo. Se propone un enfoque eficiente y económico para

	<p>Los resultados del estudio permiten enfocar el diseño de las actividades en mi experiencia de aprendizaje, ya que brinda recomendaciones para abordar la enseñanza de la programación de software</p>	<p>la enseñanza de la programación informática, utilizando una herramienta online para motivar la práctica en los estudiantes, una base teórica y el uso de los celulares</p>
<p>López & Marcos, 2018.</p>	<p>Artículo en el cual se presenta la experiencia y resultados de la transformación de las didácticas para la enseñanza de la programación, aplicando enfoques de las pedagogías emergentes para mejorar el interés de los estudiantes frente a nuevos contenidos.</p>	<p>Se presenta una propuesta para la transformación de la metodología en la enseñanza de programación basada en el enfoque de clase invertida.</p> <p>De acuerdo a los resultados presentados el enfoque didáctico seleccionado permite mejorar el interés de los estudiantes por consultar las soluciones a los problemas planteados</p>
<p>Martínez & Vinuesa, 2018.</p>	<p>Reflexión sobre la implementación de una</p>	<p>Las diferencias de género marcan el grado de interés</p>

	<p>experiencia educativa en la cual se evalúa las posibles consecuencias que la diferencia de género puede evidenciar en el aprendizaje de la programación computacional.</p>	<p>hacia la programación más no la capacidad para su aprendizaje.</p> <p>Una mejora esperada del aprendizaje de la programación es las expectativas socioeconómicas de los estudiantes.</p> <p>La informática se ha ido incorporando de manera desestructurada en el sistema educativo.</p>
--	---	---

Fuente: Elaboración propia.

De los trabajos realizados por los autores es importante mencionar que aportan en gran medida al presente proceso de sistematización, ya que dejan claro la viabilidad para la implementación de una práctica educativa que involucre conceptos de las ciencias de computación con estudiantes de igual rango de edad y similar contexto socio económico. También los autores hacen mención a la importancia de la estimación del tiempo previsto para cada actividad en el diseño tecno pedagógico, ya que este debe tener en cuenta retrasos generados por los componentes tecnológicos u organizativos y finalmente el estudio de estos trabajos ha permitido afianzar el conocimiento sobre diferentes enfoques didácticos y reconocer sus ventajas y debilidades en ambientes reales de formación.

5. Fase 3. Recuperación y reconstrucción de la práctica educativa

5.1. Diseño metodológico.

La ruta metodológica que guía el proceso de la actual sistematización educativa incluye tres momentos o fases, descripción, reflexividad e interpretación. La metodología DRI da coherencia al proceso de sistematización educativa permitiendo una comprensión más profunda de la práctica y facilita la comunicación de los hallazgos y recomendaciones de manera efectiva.

Tabla 2. Diseño metodológico.

Eje	Dimensión de observación y análisis	Fuentes	Instrumento
Uso de plataformas de programación en bloques y dispositivos interactivos para el desarrollo del pensamiento algorítmico como estrategia de	Uso de tecnología y recursos digitales: Evaluar cómo se utilizan las plataformas de programación en bloques y dispositivos interactivos en el aula.	Docente y estudiantes del grado 10 de la especialidad de programación.	Diario de campo del docente. Formulario de autoevaluación. Página web con la descripción del proyecto.

aprendizaje en una fase introductoria.	Diseño de actividades de aprendizaje: Analizar cómo se diseñan y estructuran las actividades de aprendizaje.	Docente.	Diario de campo del docente. Plan de clase.
	Participación y compromiso: Observar la participación y el compromiso de los estudiantes en las actividades con tecnología.	Docente y estudiantes del grado 10 de la especialidad de programación.	Diario de campo del docente. Formulario de coevaluación. Formulario de autoevaluación.
	Desarrollo del pensamiento algorítmico: Evaluar cómo se fomenta y desarrolla el pensamiento	Docente y estudiantes del grado 10 de la especialidad de programación.	Diario de campo del docente. Formulario de autoevaluación. Página web con la descripción del

	algorítmico en los estudiantes.		proyecto.
	<p>Creatividad y resolución de problemas:</p> <p>Observar la capacidad de los estudiantes para aplicar el pensamiento algorítmico en situaciones creativas y desafiantes.</p>	Docente y estudiantes del grado 10 de la especialidad de programación.	<p>Diario de campo del docente.</p> <p>Formulario de autoevaluación.</p> <p>Página web con la descripción del proyecto.</p> <p>Exposición del prototipo construido.</p>
	<p>Impacto en el aprendizaje:</p> <p>Evaluar el impacto general en el aprendizaje de los estudiantes.</p>	Docente y estudiantes del grado 10 de la especialidad de programación.	<p>Diario de campo del docente.</p> <p>Formulario de autoevaluación.</p> <p>Formulario de coevaluación.</p>

Fuente: Elaboración propia.

5.2. Recuperación y reconstrucción de la práctica

5.2.1. Diseño

El objetivo de aprendizaje sobre el cual se desarrolla la práctica educativa pertenece al currículo del curso en programación de software y he tenido la oportunidad de desarrollarlo durante 3 veces en la misma institución educativa. Cada una de las anteriores implementaciones de las actividades propuestas me ha dejado resultados académicos, relatos de mis estudiantes y reflexiones propias que de alguna manera u otra han influido y enriquecido la siguiente implementación.

El proceso de diseño de la presente práctica educativa se inicia con una reflexión pedagógica que reconoce una necesidad o problemática evidente en el proceso de aprendizaje de mis estudiantes, esta identificación se basa en las observaciones y análisis previos de mis intervenciones como docente. Dos elementos críticos emergen con claridad en este proceso de diagnóstico: el desfase en la complejidad de los contenidos abordados y la desmotivación manifestada por los estudiantes en relación con su propio proceso de aprendizaje.

El primer aspecto, el desfase en la complejidad de los contenidos, sugiere que existe una desconexión entre lo que se enseña y la capacidad de comprensión y asimilación de los estudiantes, lo que plantea la necesidad de adaptar la práctica para garantizar que los contenidos sean presentados de manera gradual y accesible, permitiendo que cada estudiante pueda construir su comprensión de forma sólida y progresiva.

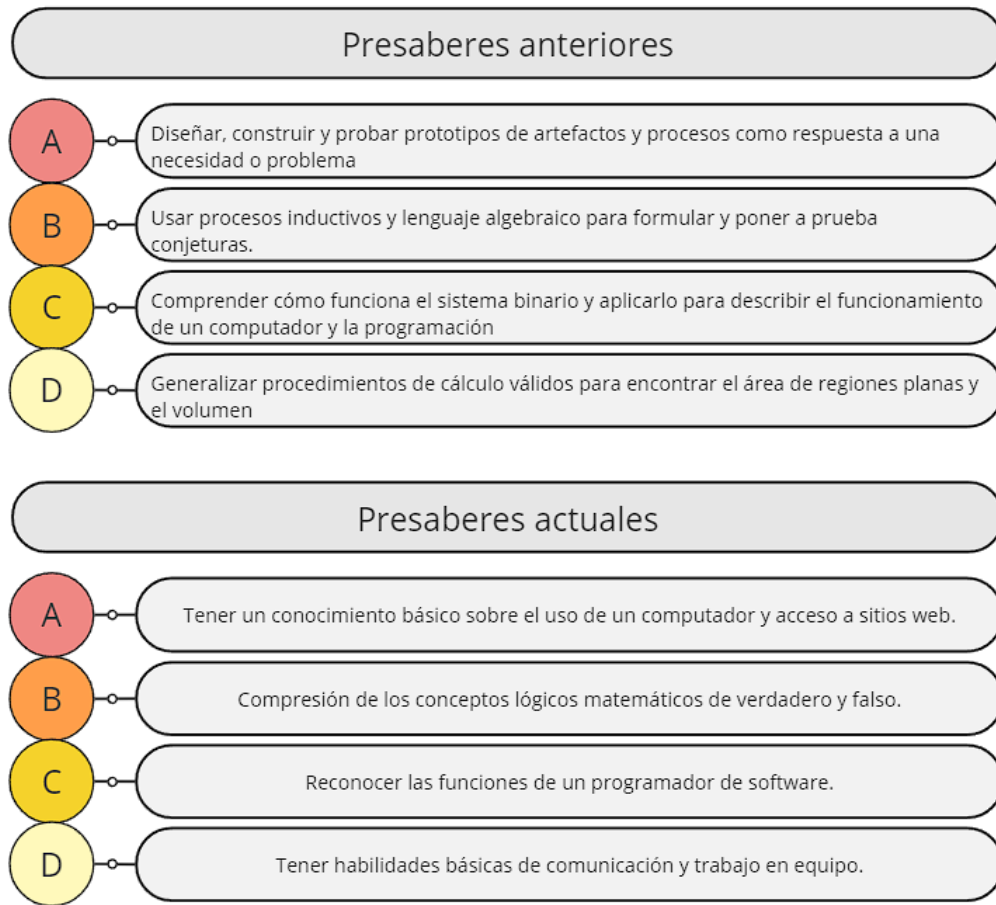
En cuanto a la desmotivación observada entre los estudiantes, este es un aspecto crítico a abordar en el diseño de la práctica puesto que, la desmotivación tiene un impacto significativo en el aprendizaje, disminuyendo el interés y la participación activa de los

estudiantes, por lo tanto, es esencial diseñar estrategias que no solo capten su atención, sino que también fomenten su entusiasmo y compromiso con el proceso de aprendizaje.

El proceso de diseño de la nueva práctica educativa ha sido guiado por la comprensión de las necesidades y desafíos que enfrentan los estudiantes en su proceso de aprendizaje. Identificar y reconocer estos desafíos me motivó a investigar sobre enfoques pedagógicos y didácticos que se aplican con éxito en la enseñanza de la programación computacional, este proceso de auto aprendizaje me permitió no solo adquirir una sólida base teórica, sino también comprender la aplicabilidad práctica de estos enfoques en el contexto de la educación, comprendí que la enseñanza de la programación no solo se trata de proporcionar conocimientos técnicos, sino de cultivar habilidades de resolución de problemas, pensamiento lógico y creatividad en los estudiantes, lo que se convirtió en un principio fundamental en el diseño de la práctica educativa.

Al analizar la pertinencia y el alcance de los objetivos de aprendizaje para la unidad en cuestión, pronto me di cuenta de la importancia de alinear de manera coherente los pre saberes definidos con las expectativas reales de mis estudiantes, de esta manera reconocí que había una brecha significativa entre lo que se esperaba que los estudiantes supieran y lo que realmente entendían y podían aplicar. Este reconocimiento fue fundamental para ajustar el enfoque pedagógico y diseñar una práctica que abordara específicamente estas deficiencias.

Figura 3. Listado de presaberes.



Fuente: Elaboración propia.

Durante el proceso de investigación y consulta sobre los enfoques pedagógicos más adecuados para la enseñanza de la programación, explore diversos enfoques que van desde la gamificación hasta la robótica educativa, cada uno con sus propias ventajas y enfoques únicos. Sin embargo, lo que realmente llamó mi atención y resonó con el contexto de mi institución educativa fueron los entornos de aprendizaje tipo makerspace. Estos entornos ofrecen un enfoque altamente interactivo y práctico para la enseñanza de la programación, permitiendo a los estudiantes experimentar y crear, lo que va de la mano con la filosofía de

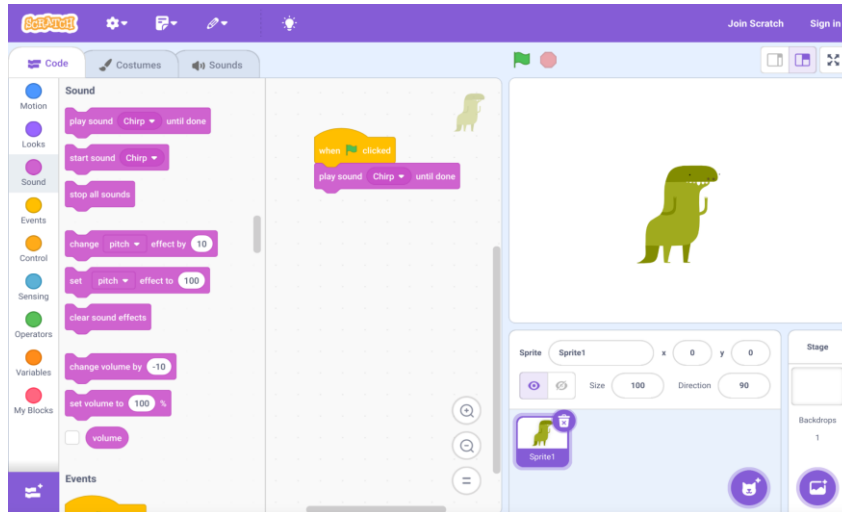
aprendizaje activo que considero esencial para el desarrollo de habilidades de programación.

Este enfoque también se alinea con la idea de que el aprendizaje de la programación va más allá de la adquisición de habilidades técnicas, fomentando la resiliencia frente a desafíos y la capacidad de trabajar en equipo, habilidades que son esenciales en la sociedad actual y futura.

Realizado la selección de un enfoque didáctico acorde y de ajustar los presaberes para el objetivo de aprendizaje, inicié un proceso de curaduría de herramientas TIC disponibles que se alinearan con lo que deseaba que aprendieran los estudiantes, mi conocimiento técnico pedagógico y los recursos disponibles en la IE, en esta búsqueda, identifiqué tres herramientas tecnológicas que se destacaban por su capacidad para cumplir con estos requisitos: Scratch, Micro:bit y Arduino. Cada una de estas herramientas tiene sus propias ventajas y características únicas que las hacen adecuadas para mi contexto educativo.

Scratch, por ejemplo, se caracteriza por su interfaz amigable y su enfoque orientado a niños y principiantes en programación, ofrece una introducción simple a los conceptos a través de bloques visuales, lo que lo convierte en una excelente opción para estudiantes que están dando sus primeros pasos en la programación.

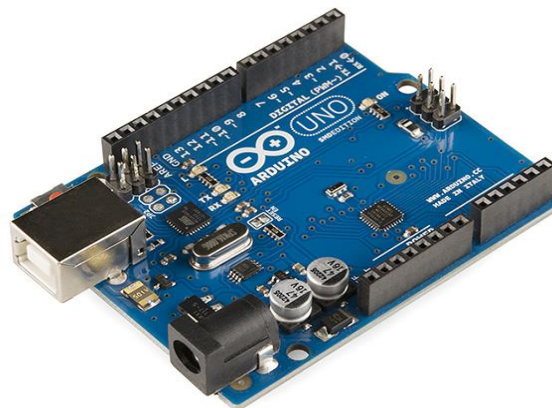
Figura 4. Interfaz de Scratch.



Fuente: Recuperado de Wikipedia.

Arduino, por otro lado, es una plataforma más avanzada que combina la programación con la electrónica, aunque es poderosa y versátil, su curva de aprendizaje puede ser más pronunciada, ya que implica un conocimiento más profundo, por lo tanto, lo consideré más allá del alcance de la unidad en cuestión.

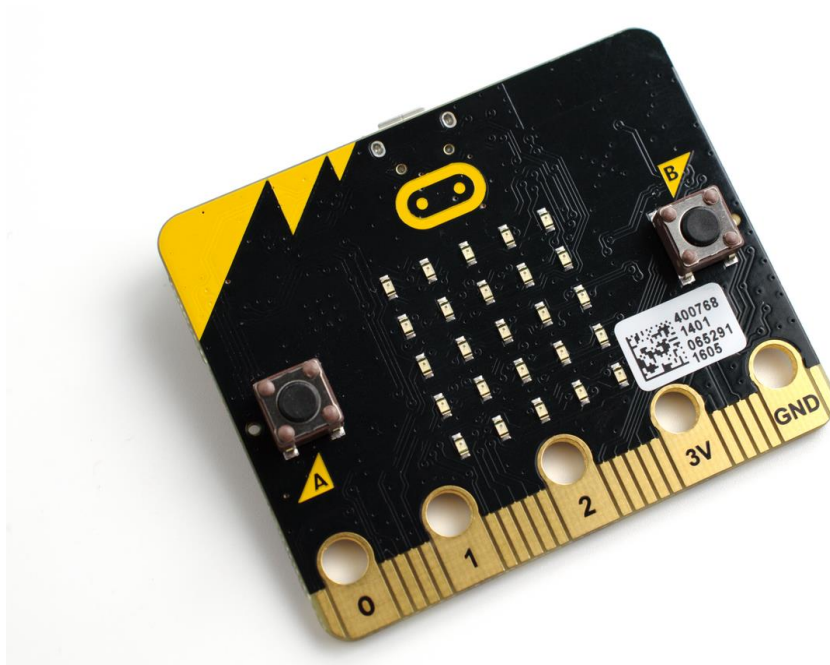
Figura 5. Tarjeta Arduino Uno.



Fuente: Recuperado de Wikipedia.

Finalmente, Micro:bit emergió como la elección ideal. Esta pequeña placa programable ofrece un equilibrio perfecto entre simplicidad y capacidad, es ideal para introducir a los estudiantes en la programación, al tiempo que les permite crear proyectos interactivos y desarrollar habilidades prácticas. Además, Micro:bit estaba disponible en mi institución educativa, lo que facilita la implementación.

Figura 6. Tarjeta Micro:bit v2.



Fuente: Recuperado de Wikipedia.

El proceso de diseño de esta práctica educativa ha estado enriquecido por varias fortalezas que han desempeñado un papel fundamental en mi desarrollo como docente. En primer lugar, quiero destacar mi voluntad de mejorar constantemente mi práctica pedagógica la cual ha sido impulsada en gran medida por los conocimientos adquiridos durante mi experiencia de maestría. A través de esta experiencia académica, he tenido la

oportunidad de explorar teorías y enfoques pedagógicos, lo que ha influido significativamente en mi proceso de enseñanza. Me ha permitido comprender la importancia de ir más allá de la simple transmisión de información y centrarme en la creación de experiencias educativas efectivas.

Además, en el contexto institucional en el que trabajo, se han creado las condiciones propicias para la generación de nuevas experiencias educativas. La institución valora y respeta la autonomía docente, lo que significa que tengo la libertad de diseñar y llevar a cabo planes de enseñanza innovadores que se adapten a las necesidades y expectativas de mis estudiantes. Esta autonomía se combina con un sólido apoyo institucional para la obtención de recursos educativos. Esto es esencial para la implementación exitosa de cualquier práctica pedagógica, ya que brinda acceso a herramientas y tecnologías que enriquecen el proceso de aprendizaje de los estudiantes.

Finalmente, es importante destacar que la institución reconoce y valora la innovación en la enseñanza. Esta cultura de reconocimiento motiva a los docentes a buscar constantemente nuevas formas de involucrar y empoderar a los estudiantes en su proceso de aprendizaje. Como resultado, me siento respaldado y alentado a tomar decisiones pedagógicas y a explorar enfoques educativos que vayan más allá de lo convencional.

Como resultado del proceso de diseño presento a continuación el plan de la práctica educativa, en ella se describen 4 sesiones llamadas: Presentación, exploración, creación de prototipos y comunicación.

Tabla 3. Plan de clase.

Estándares educativos que contribuye a lograr este Plan de Clase	
Desarrollar la solución de software de acuerdo con el diseño y metodologías de desarrollo.	
Objetivos de aprendizaje generales y específicos del Plan de Clase	
<p>General:</p> <ul style="list-style-type: none">● Comprender los conceptos básicos de programación y aplicarlos para crear proyectos simples donde se integre la programación con la interacción física. <p>Específicos:</p> <ul style="list-style-type: none">● Aplicar el pensamiento algorítmico para solucionar problemas cotidianos mediante una secuencia de pasos simples.● Describir el concepto de estructuras de control aplicadas en programación computacional.● Emplear las estructuras de control para diseñar algoritmos que solucionen un problema específico.● Fomentar la persistencia en el diseño de algoritmos reconociendo este enfoque como un proceso accesible y valioso.	
Duración de la implementación (número de sesiones y tiempo de cada sesión).	El plan de clase se implementará en 4 sesiones de 3 horas y media cada una.
Espacio en el cual se realizará la implementación	Sala de sistemas.
Requisitos para implementar el Plan de Clase (<i>se especifican los conocimientos previos necesarios antes de empezar a trabajar en la experiencia educativa -cobrimiento de temas específicos o manejo de herramientas informáticas-</i>)	

<p>Pre saberes:</p> <ul style="list-style-type: none"> ● Tener un conocimiento básico sobre el uso de un computador y acceso a sitios web. ● Comprensión de los conceptos lógicos matemáticos de verdadero y falso. ● Reconocer las funciones de un programador de software. ● Tener habilidades básicas de comunicación y trabajo en equipo. 	
<p>Recursos y materiales necesarios <i>(se especifican los elementos o información necesarios para que el estudiante pueda trabajar adecuadamente en la experiencia educativa -equipos de cómputo, software, Apps, acceso a Internet, enlaces a sitios Web, etc-)</i></p>	
<ul style="list-style-type: none"> ● 30 computadores con acceso a internet. ● Video Beam. ● Tarjetas Micro:bit, con cable USB y porta baterías. 	
<p>Saberes</p>	
<p>Saber conocer. Conceptuales:</p> <ul style="list-style-type: none"> ● Definir los conceptos de algoritmo y pensamiento algorítmico. ● Identificar las estructuras de control condicionales y repetitivas en los algoritmos <p>Saber hacer. Procedimentales:</p> <ul style="list-style-type: none"> ● Emplear características del pensamiento algorítmico para definir y solucionar un problema. ● Aplicar estructuras de control en la construcción de algoritmos. <p>Saber ser. Actitudinales:</p> <ul style="list-style-type: none"> ● Demostrar capacidad para enfrentar desafíos y fracasos al construir algoritmos y aprender de ellos. 	
<p>Interacciones del docente</p>	<p>Interacciones de los estudiantes</p>
<p>Sesión 1 (210 minutos)</p>	<p>Sesión 1 (210 minutos)</p>

- Ambientar el inicio de la unidad comunicando los objetivos y las actividades de aprendizaje.

Actividad 1: Evaluación Diagnóstica. (30 min)

- Guiar a los estudiantes para que participen de la encuesta diagnóstica en la plataforma kahoot.
- Resolver dudas de los estudiantes y justificar cada una de las respuestas.

Actividad 2: Presentación. (180 minutos)

- Presentar a los estudiantes el prototipo del juego Snake o culebrita preparado.
- Presentar la tarjeta Micro:bit y sus características.
- Presentar la plataforma de programación MakeCode.
- Hacer una demostración de cómo cargar un programa desde la plataforma a la tarjeta.

Actividad 1: Evaluación Diagnóstica. (30 min)

- Participar y contestar cada una de las preguntas de la encuesta de acuerdo a su conocimiento actual.
- Solicitar aclaración por parte del docente cuando sea necesario.

Actividad 2: Exploración. (180 minutos)

- Organizar los espacios para desarrollar las siguientes actividades.
- Interactuar con las tarjetas Micro:bit y reconocer sus componentes.
- Explorar la plataforma de programación MakeCode.
- Cargar un ejemplo de algoritmo de la plataforma MakeCode a la tarjeta Micro:bit.

Sesión 2 (210 minutos)

Actividad 3: Conceptos. (90 minutos)

- Explicar los conceptos básicos de programación como secuencias y condicionales.
- Presentar ejemplos sencillos de programas utilizando estas estructuras.
- Guiar a los estudiantes a través de la creación de su primer programa simple.
- Ayudar a los estudiantes a solucionar problemas si tienen dificultades.
- Invitar a los estudiantes a explorar más con las tarjetas Micro:bit y continuar aprendiendo sobre la programación.

Actividad 4: Exploración. (120 minutos)

- Presentar un requerimiento y su algoritmo de solución.

Sesión 2 (210 minutos)

Actividad 3: Conceptos. (90 minutos)

- Analizar los conceptos presentados y aclarar las respectivas dudas.
- Construir los programas de ejemplo observados.
- Recibir instrucciones para la corrección de fallos.
- Validar el correcto funcionamiento del programa y analizar el comportamiento programado.
- Realizar acciones como búsqueda de información adicional sobre la programación o el uso de la tarjeta Micro:bit.

Actividad 4: Exploración. (120 minutos)

- Analizar el requerimiento solicitado.
- Construir un prototipo para dar solución a partir del programa presentado.

- Presentar un plan de pruebas y validar que la solución propuesta cumple con los requerimientos solicitados.
- Incentivar a que los estudiantes tomen el reto de mejorar o proponer nuevas soluciones al mismo requerimiento.

Sesión 3 (240 minutos)

Actividad 5: Crear Prototipos. (240 minutos)

- Motivar la creatividad de los estudiantes para seleccionar un problema del entorno y diseñar una solución.
- Realizar retroalimentación constante a las decisiones tomadas en los grupos de trabajo.

- Validar el funcionamiento del prototipo.

Sesión 3 (240 minutos)

Actividad 5: Crear Prototipos. (240 minutos)

- Generar ideas de posibles problemas susceptibles de solucionar utilizando la programación con los recursos y tiempo asignado.
- Implementar una estrategia para la construcción de un prototipo.
- Validar que el prototipo cumple con los mínimos requerimientos.

Sesión 4 Compartir (150 minutos)

Actividad 7: Exposición (120 minutos)

- Dar la oportunidad a los estudiantes de compartir sus programas con el resto de la clase y explicar cómo funcionan.
- Animar a los estudiantes a que hagan preguntas y comentarios sobre los programas de sus compañeros.

Actividad 8: Final (30 minutos)

- Motivar la participación en un diálogo en el cual los estudiantes reflexionen sobre lo que han aprendido y cómo podrían aplicar estos conocimientos en su cotidianidad.
- Presentar las encuestas de coevaluación y autoevaluación.

Sesión 4 Compartir (150 minutos)

Actividad 7: Exposición (120 minutos)

- Preparar la forma de comunicar a sus compañeros el programa realizado.
- Realizar una presentación que detalle, los resultados, el proceso y los inconvenientes para construir el programa.
- Escuchar con atención las presentaciones de otros grupos.

Actividad 8: Final (30 minutos)

- Expresar sus emociones e ideas acerca de los conocimientos adquiridos.
- Evaluar de manera objetiva el nivel de los conocimientos adquiridos y el desempeño de sus compañeros en la consecución del objetivo grupal.

Sesión 1

Evaluación diagnóstica

Disponible en la plataforma Kahoot: <https://create.kahoot.it/share/evaluacion-diagnostica/7a698571-e5a8-47b6-8fbf-8de51baf6ea8>

¿Qué es un navegador web y para qué se utiliza?

¿Qué es un programa de computador?

¿La representación matemática de falso es 1?

¿Qué entiendes por algoritmo?

¿Cuál es la función de un programador de software?

¿Cuál es el primer paso para crear un programa o aplicación?

¿Se puede considerar una receta de cocina como un ejemplo de algoritmo?

¿En un videojuego, qué elementos se representan con valores numéricos?

¿Cuáles consideras que son los beneficios de aprender a programar?

Lista de Chequeo

Aspecto a evaluar	Criterios	Cumple	No cumple
Compromiso para explorar la plataforma MakeCode.	¿Utiliza los recursos como computadores y tarjetas Micro:bit para desarrollar las actividades propuestas?		
	¿Utiliza la plataforma MakeCode de acuerdo a las indicaciones sugeridas?		
	¿Realiza la verificación de los programas presentados?		
	¿Busca información adicional que le permita conocer mejor la plataforma MakeCode?		
Responsabilidad en el trabajo en equipo.	¿Realiza las tareas asignadas por el grupo en los tiempos acordados?		
	¿Respeta las opiniones de los demás miembros del grupo?		
	¿Lidera las acciones para desarrollar las actividades de la mejor manera posible?		
Lógica y razonamiento:	¿Comprende la diferencia entre declaraciones verdaderas y falsas?		
	¿Reconoce los operadores de comparación para construir condiciones?		

Sesiones 2 y 3. Rúbrica Analítica

Criterios	Superior	Alto	Básico	Bajo
	4,6 - 5	4,1 - 4,5	3,5 - 4,0	0 - 3,4
Identificar claramente las estructuras de control condicionales en los programas.	Identifica claramente y de manera detallada las estructuras de control condicionales, proporcionando ejemplos específicos de su función.	Identifica claramente las estructuras de control condicionales y explica su función.	Identifica las estructuras de control condicionales de manera general, con algunos errores y sin dar una explicación clara.	No identifica de manera clara las estructuras de control condicionales o las identifica de manera incorrecta.
Demostrar capacidad de análisis para ampliar las funcionalidades del programa construido.	Demuestra una gran capacidad de análisis y justifica la utilidad de los cambios en el programa construido.	Demuestra capacidad de análisis para identificar posibles mejoras en el programa construido.	Identifica algunas posibles mejoras para el programa construido, pero no demuestra un análisis profundo.	No demuestra capacidad de análisis para ampliar las funcionalidades del programa construido.
Planear y construir soluciones creativas y efectivas a un requerimiento.	Planifica y construye una solución excepcionalmente creativa y efectiva al requerimiento.	Planifica y construye una solución efectiva y creativa al requerimiento.	Presenta un plan y construye una solución con algunas limitaciones.	No demuestra capacidad para planear y construir soluciones creativas y efectivas a un requerimiento.
Demostrar capacidad de liderazgo en la realización de las actividades propuestas.	Evidencia un alto nivel de liderazgo en la realización de las actividades propuestas, fomentando un ambiente de trabajo positivo y colaborativo.	Evidencia liderazgo en la realización de las actividades propuestas, promueve al equipo de trabajo a alcanzar los objetivos propuestos.	Evidencia liderazgo en la realización de las actividades propuestas, pero de manera limitada, aporta algunas ideas y propuestas al equipo de trabajo.	No se evidencia liderazgo en la realización de las actividades propuestas, no se observa motivación ni influencia en el trabajo en equipo.

Sesión 4

Formulario de autoevaluación: <https://forms.gle/ZWZ5gvJmtuGXXbtk7>

Formulario de coevaluación: <https://forms.gle/aiCUfqc92Kr5z1XD8>

Fuente: Elaboración propia.

5.2.2. Implementación

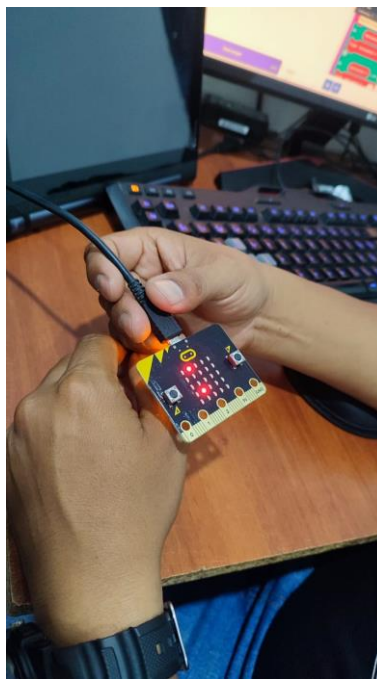
Tras un proceso de diseño el cual generó como resultado un plan de clase ambicioso e innovador, se procedió a la implementación de la actividad de clase durante las dos primeras semanas del segundo periodo académico del año 2023.

En el comienzo de la primera sesión, los estudiantes se reunieron en la sala de sistemas 2 de la IE, el docente comenzó explicando los objetivos de aprendizaje seleccionados para la unidad, proporcionando a los estudiantes una visión clara de lo que se esperaba lograr al final. También brindó un resumen de las actividades que se llevarían a cabo durante la sesión. con el objetivo de que los estudiantes anticiparan el flujo de la clase, se prepararan para las acciones por delante y se sintieran partícipes del proceso de enseñanza.

El docente destacó la importancia de resolver todas las dudas que surgieran durante el proceso de aprendizaje, alentó a los estudiantes a hacer preguntas y buscar claridad en cualquier momento, los estudiantes fueron alentados a no rendirse ante los desafíos y a ver los errores como oportunidades para aprender y mejorar.

Para captar la atención e intriga de los estudiantes desde el principio, el docente presentó una tarjeta Micro:bit previamente programada con un prototipo del juego de la "culebrita", esta intervención práctica y visual generó curiosidad y entusiasmo entre los estudiantes quienes se vieron inmediatamente intrigados por la tarjeta y el juego presentado. La naturaleza visual y práctica de esta introducción capturó su imaginación y los llevó a hacer preguntas sobre cómo funcionaba y qué más podían hacer con ella. Las preguntas planteadas por los estudiantes, "¿Se pueden poner más juegos en la tarjeta?" o "¿Cómo detecta que estoy moviendo la tarjeta?", revelaban su deseo de explorar y entender el funcionamiento, estaban ansiosos por aprender cómo programar la tarjeta y crear sus propios juegos y proyectos, proporcionando una razón motivadora para participar activamente en la lección.

Figura 7. Demostración del juego de la “culebrita”.



Fuente: Elaboración propia.

A medida que los estudiantes planteaban preguntas y expresaban su interés en la tarjeta Micro:bit se crearon las condiciones en un entorno de aprendizaje *maker*, la curiosidad natural es fundamental ya que impulsa a los estudiantes a investigar, experimentar y buscar soluciones por sí mismos. De igual manera, la interacción fomentó la colaboración entre compañeros, ya que no solo estaban aprendiendo de sus preguntas, sino que también estaban aprendiendo unos de otros

A continuación, el docente compartió con el grupo la evaluación diagnóstica preparada en la plataforma Kahoot, instrumento de evaluación que tuvo como objetivo validar los presaberes de la unidad y desencadenar una serie de instrucciones por parte del docente que permitió nivelar dichos conocimientos.

Figura 8. Evidencia de evaluación diagnóstica.











¿Qué es un programa de computador?



Fuente: Elaboración propia.

En la figura 9, se puede observar el resumen de los resultados obtenidos en la evaluación diagnóstica.

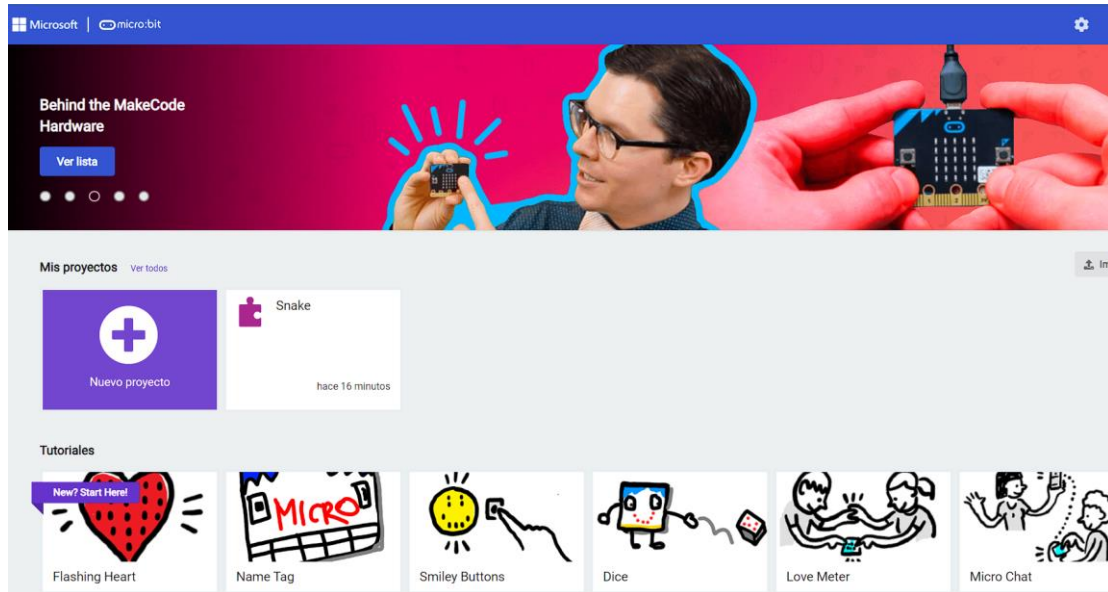
Figura 9. Resultados evaluación diagnóstica.

Pregunta ▾	Tipo ▾	Correcto/incorrecto ▾
1 "¿Qué es el sistema binario?"	Quiz	 68%
2 ¿Qué es un programa de computador?	Quiz	 61%
3 La representación matemática de falso es 1.	Verdadero o falso	 42%
4 ¿Qué entiendes por algoritmo?	Quiz	 65%
5 ¿Cuál es la función de un programador de software?	Quiz	 61%
6 ¿Qué es un lenguaje de programación?	Quiz	 68%
7 El primer paso para crear un programa o aplicación es:	Quiz	 35%
8 ¿Se puede considerar una receta de cocina como un ejemplo de algoritmo?	Verdadero o falso	 87%
9 ¿En un video juego, que elementos se representan con valores numéricos?	Quiz	 61%
10 ¿Cuales son los beneficios de aprender a programar?	Quiz	 71%

Fuente. Plataforma Kahoot.

Se procedió a presentar la plataforma MakeCode, herramienta esencial para la programación de la tarjeta Micro:bit. Primero, proporcionó una visión general de cómo acceder a la plataforma, a continuación, el docente se adentró en los detalles para crear un nuevo proyecto, explicó los pasos de manera clara y concisa, asegurándose de que cada estudiante comprendiera cómo iniciar su propio proyecto. Esta parte de la clase no solo enseñó habilidades técnicas, sino que también fomentó la independencia y la confianza de los estudiantes al permitirles comenzar sus propios proyectos de inmediato.

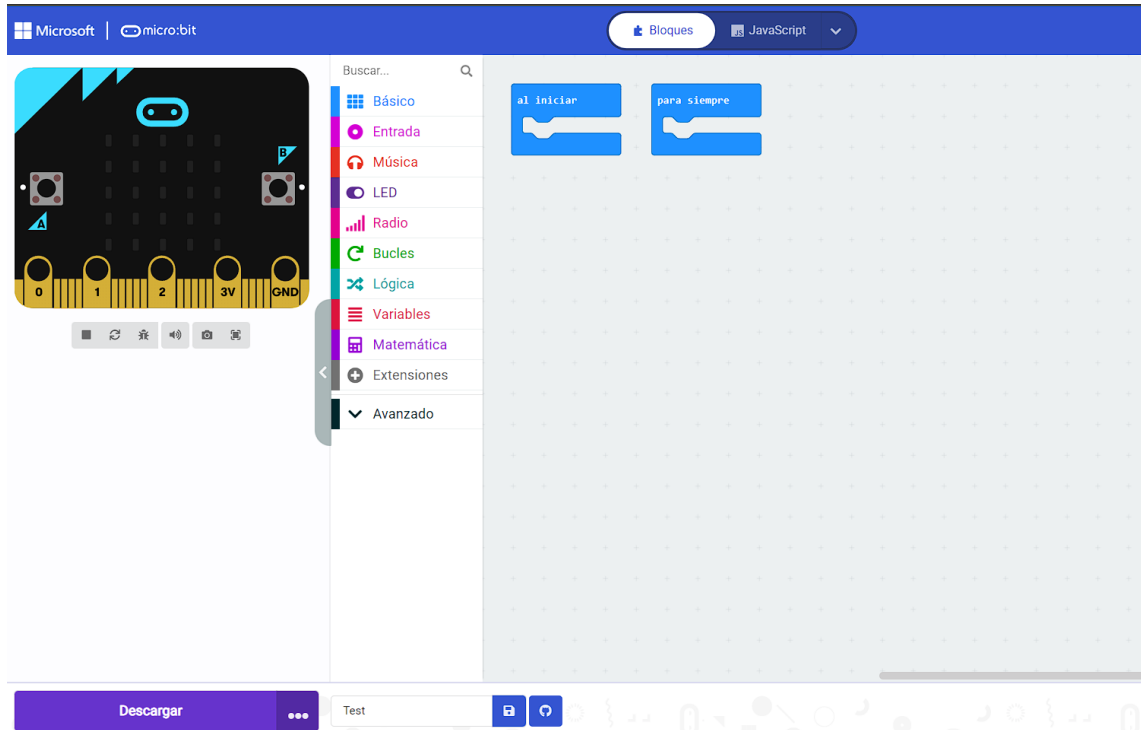
Figura 10. Interfaz inicial de la plataforma MakeCode.



Fuente: Elaboración propia.

Finalmente, el docente describió la disposición de las secciones en la interfaz de MakeCode, ayudando a los estudiantes a familiarizarse con la plataforma y a comprender cómo navegar por ella de manera eficaz esta explicación les brindó una visión general de las herramientas disponibles, lo que despertó su interés por explorar.

Figura 11. Interfaz de trabajo en la plataforma MakeCode.



Fuente: Elaboración propia.

La etapa de experimentación marcó un punto de partida en el proceso de aprendizaje de programación. El docente, después de haber introducido la plataforma MakeCode y haber explicado cómo acceder y crear proyectos, dio un paso adelante para demostrar cómo agregar bloques de programación al espacio de trabajo, esta demostración práctica fue un momento crucial en la lección, ya que permitió a los estudiantes ver cómo tomar sus ideas y convertirlas en código real.

Atendiendo las dimensiones de la educación maker alentó a los estudiantes a seguir su ejemplo mientras él arrastraba y soltaba bloques de programación en la interfaz de MakeCode. A medida que iba construyendo un pequeño programa paso a paso, explicaba el propósito de cada bloque y cómo se relacionaban entre sí para lograr un objetivo específico.

Una vez que había creado un programa simple en MakeCode, el docente invitó a los estudiantes a participar activamente. Los estudiantes se sumergieron en la tarea, entusiasmados por la oportunidad de aplicar lo que habían aprendido hasta ahora. Con la guía del docente y su propia creatividad, comenzaron a construir sus propios programas y a experimentar con diferentes combinaciones de bloques de programación.

Figura 12. Evidencia de implementación.



Fuente: Elaboración propia.

Esta etapa de experimentación no solo se centró en la práctica de codificación, sino que también promovió la indagación y el análisis. Los estudiantes exploraron cómo los cambios en el código afectan el comportamiento de la tarjeta Micro:bit, lo que los llevó a hacer preguntas y buscar respuestas. Se les animó a reflexionar sobre los resultados obtenidos en cada prueba, lo que fomentó su pensamiento crítico y su capacidad para aprender de sus propios errores.

En esta fase se alentó para que los estudiantes compartieran sus observaciones y descubrimientos. Preguntando cosas como: "¿Qué notaron mientras experimentaban con los bloques de programación?", "¿Qué diferencias observaron cuando cambiaron ciertas instrucciones en el código?", "¿Pudieron identificar algún patrón o regla que rige el comportamiento de la tarjeta Micro:bit?".

Estas preguntas abiertas y reflexivas sirvieron como punto de partida para una conversación significativa en el aula. Los estudiantes comenzaron a compartir sus observaciones y a discutir los resultados de sus pruebas a medida que se desarrollaba la discusión, el docente desempeñó el papel de facilitador, ayudando a los estudiantes a conectar sus observaciones con conceptos clave.

Durante este proceso, se construyeron conceptos importantes de programación. Los estudiantes comenzaron a comprender la noción de algoritmo, cómo un conjunto de instrucciones puede llevar a cabo una tarea específica. También se introdujeron conceptos como variables y condiciones a medida que los estudiantes exploraban cómo ciertos cambios en el código podían alterar el comportamiento del programa.

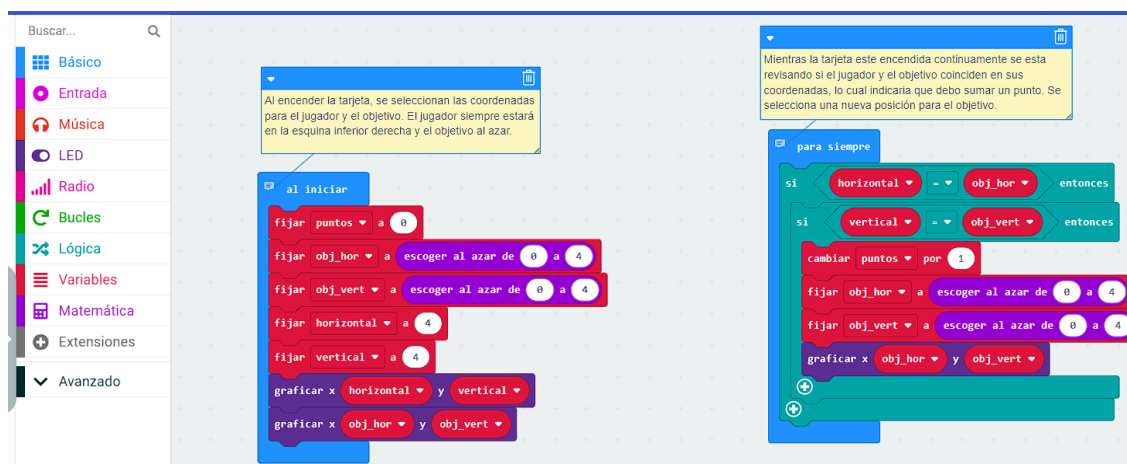
Bajo el enfoque de educación maker, la evaluación formativa se centró en la observación y valoración constante del proceso creativo y constructivo de los proyectos llevados a cabo en las siguientes sesiones. Siguiendo la estrategia de evaluación diseñada, el docente aplicará la rúbrica analítica, que tiene en cuenta no solo los resultados finales, sino también las interacciones y participación de cada estudiante a lo largo del proceso.

Esta metodología de evaluación se alinea con los principios de la educación maker, donde se valora no solo el producto final, sino el proceso creativo y el esfuerzo

colaborativo. La rúbrica proporciona una estructura para evaluar aspectos clave, como la originalidad de la solución, la aplicación de conceptos de programación, la colaboración en equipo y la capacidad para superar desafíos. Asimismo, el docente utilizará sus observaciones directas durante las sesiones para obtener una comprensión más completa del progreso individual de cada estudiante. Esta estrategia de evaluación formativa no solo permite la retroalimentación constante, sino que también promueve la mejora continua y el aprendizaje a partir de la experiencia.

En la siguiente sesión, el docente había preparado un archivo que contenía los bloques de programación del juego de la "culebrita" que había presentado al principio de la unidad, compartió este archivo con los estudiantes, permitiéndoles acceder al algoritmo del juego. Sin embargo, en lugar de simplemente mostrarles los bloques, el docente planteó un desafío interesante. Pidió a los estudiantes que analizaran el algoritmo en busca de variables y condiciones. Esta actividad tenía un propósito claro: ayudar a los estudiantes a reconocer y comprender cómo se habían utilizado conceptos de programación para crear el juego.

Figura 13. Bloques de código para el algoritmo del juego de la “culebrita”.



Fuente: Elaboración propia.

Los estudiantes se sumergieron en el algoritmo, observando cómo se habían utilizado variables para rastrear la posición del jugador y las condiciones para determinar cuándo obtiene un punto. Esta actividad les permitió conectar lo que habían aprendido con un ejemplo práctico y real.

Luego, el docente les propuso un desafío aún mayor. Animó a los estudiantes a realizar modificaciones en el código para transformar el juego de la "culebrita" en algo diferente. Esto requería que aplicaran sus conocimientos recién adquiridos para crear una variante del juego original.

Los estudiantes se dividieron en grupos pequeños y comenzaron a trabajar en sus versiones personalizadas del juego. Algunos optaron por cambiar las reglas, mientras que otros introdujeron nuevos elementos, como obstáculos o bonificaciones. Esta actividad promovió la creatividad y la experimentación, al tiempo que reforzó los conceptos de programación que habían estado explorando.

Al final de la sesión, cada grupo presentó su versión modificada del juego y explicó las decisiones de programación que habían tomado. Esta actividad no sólo consolidó su comprensión de los conceptos, sino que también les dio la oportunidad de compartir y aprender de las ideas de sus compañeros.

En la tercera sesión de esta unidad de aprendizaje, el docente buscó fusionar la imaginación de los estudiantes con las capacidades versátiles de la tarjeta Micro:bit, adoptando un enfoque inspirado en los makerspaces. Este enfoque tenía como objetivo empoderar a los estudiantes para que se convirtieran en creadores y solucionadores de

problemas, lo que promovería aún más su pensamiento algorítmico y habilidades de programación.

La sesión comenzó con una discusión grupal sobre la creatividad y cómo esta podría aplicarse en el mundo de la programación y la tecnología. El docente alentó a los estudiantes a compartir sus ideas y visiones sobre lo que podrían crear utilizando la tarjeta Micro:bit. Esto generó una lluvia de ideas llena de entusiasmo, con sugerencias que iban desde juegos interactivos hasta dispositivos para ayudar a resolver problemas cotidianos.

Después de esta lluvia de ideas, los estudiantes de manera voluntaria se dividieron en grupos de 3 personas pequeños y se sumergieron en la tarea de definir un requerimiento específico para un proyecto. Cada grupo eligió un problema o desafío que quería abordar utilizando la tarjeta Micro:bit. Algunos grupos se centraron en la creación de un juego, mientras que otros se dedicaron a diseñar una solución práctica para un problema cotidiano, como un dispositivo de recordatorio.

Figura 14. Evidencia de implementación.

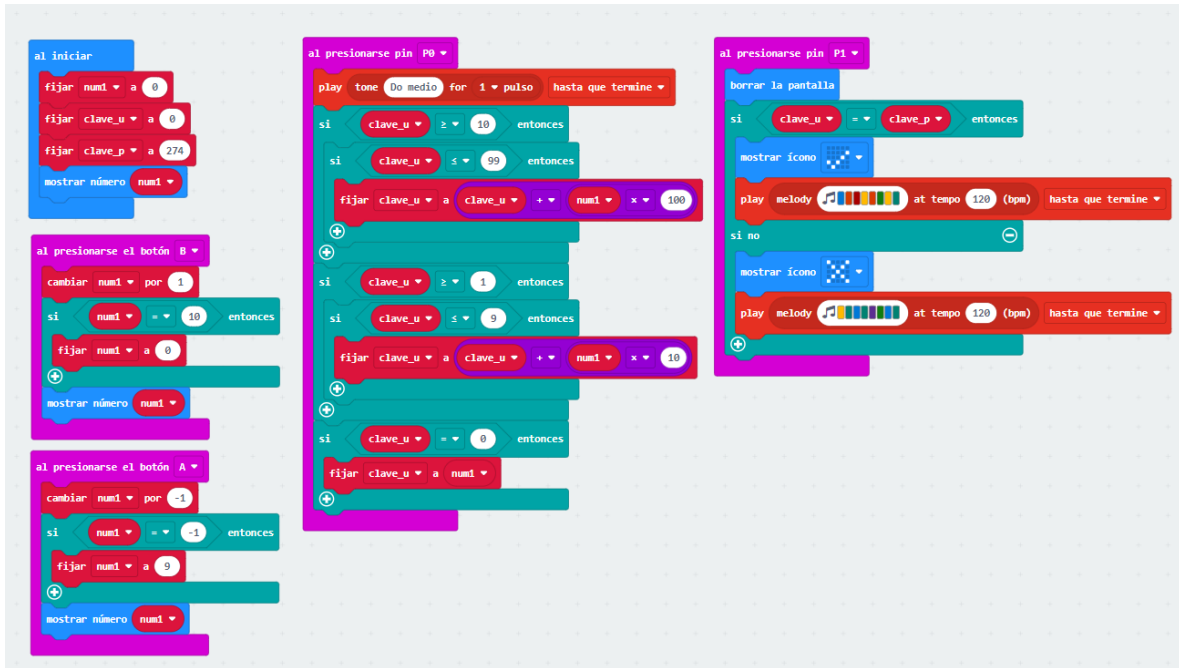


Fuente: Elaboración propia.

Con el requerimiento claramente definido, los grupos comenzaron a trabajar en la construcción de su prototipo utilizando la tarjeta Micro:bit y el entorno de programación que previamente habían explorado en sesiones anteriores. En esta etapa inicial, la mayoría de los estudiantes se sintieron empoderados al asumir la tarea, evidenciando un entusiasmo por el desafío. Sin embargo, a medida que avanzaban en la creación del código de bloques, se encontraron con nuevas situaciones que presentaban problemas lógicos inesperados, lo que generó un valioso espacio de aprendizaje. En este punto los estudiantes se vieron forzados a resolver situaciones como, por ejemplo, comparar tres valores numéricos a partir de un ejercicio en el cual se comparan dos. Otra situación que sirvió de motor para despertar la curiosidad y creatividad en los grupos fue la de superar las limitaciones técnicas que tiene la tarjeta en su número de botones. Estas nuevas subtareas que no estaban presentes en el momento del diseño y que aparecieron en el momento del desarrollo obligaron a que en los grupos se asignaran roles o funciones para resolver una situación en particular y a descubrir que varios de los grupos requerían la misma solución así su prototipo o requerimiento fuera diferente.

Este proceso de enfrentar obstáculos fue fundamental, ya que permitió a los estudiantes experimentar la naturaleza dinámica (de la industria del software) entre las fases de diseño y programación. Este momento de confrontación con problemas reales no solo fortaleció su capacidad para resolver problemas, sino que también fomentó la colaboración dentro de los grupos. Los estudiantes se vieron impulsados a compartir ideas, plantear preguntas y buscar soluciones conjuntas, creando así un entorno de aprendizaje cooperativo.

Figura 15. Programación de un prototipo de candado digital.



Fuente. Elaboración propia.

A manera de ejemplo se presenta en la figura 14, los bloques de código construidos por los estudiantes Duban Muñoz, Daniel Rodríguez y Juan David Toro para simular el comportamiento de un candado digital. Este dispositivo permitirá seleccionar una clave numérica de 3 dígitos y dar una respuesta sonora de éxito o error. En palabras de los estudiantes la principal motivación que tuvieron para el desarrollo de este prototipo “es evitar el robo de las bicicletas o prevenir que nos abran los maletines”.

Para ejemplificar el proceso en general seguido por los estudiantes en esta fase de la práctica educativa, se describirán las acciones y decisiones realizadas por los estudiantes a cargo del prototipo de candado digital. Los estudiantes, impulsados por el deseo de salvaguardar sus bicicletas, encontraron inspiración en la idea de crear un candado digital, estableciendo así una conexión con el juego de la culebrita previamente presentado por el

docente. Inicialmente, idearon la incorporación de un teclado y un motor al Micro:bit para su proyecto, aunque, debido a limitaciones de tiempo, decidieron enfocarse principalmente en la programación. Realizaron modificaciones al código que genera los movimientos de la culebrita, transformándolos en una secuencia de números e implementaron bloques de comparación para determinar cuándo dos números eran iguales. Un paso crucial en su proyecto fue el desarrollo de un método para que el usuario pudiera seleccionar y guardar un número de tres dígitos en una única variable, para lograr superar este desafío requirieron dejar a un lado la plataforma de programación y tomar papel y lápiz para escribir como resolver esta cuestión. Este obstáculo les permitió reconocer la relación entre programación y otras áreas del conocimiento como matemáticas particularmente y la importancia de combinar habilidades como el liderazgo, la resiliencia y la comunicación.

Esta secuencia de decisiones demuestra la habilidad de los estudiantes para adaptar y aplicar concepto, sino también su capacidad para enfrentar desafíos y buscar soluciones creativas.

El docente desempeñó un papel fundamental al ofrecer orientación y apoyo a medida que los grupos trabajaban en sus proyectos. Entre las intervenciones realizadas por el docente se pueden destacar: ayudar a los grupos en la planificación de sus proyectos, proceso que implicó discusiones sobre la viabilidad técnica teniendo en cuenta los recursos y tiempo disponibles y la alineación con los objetivos de aprendizaje. Promovió la colaboración entre los miembros del grupo, enfatizando la importancia del trabajo en equipo y la asignación equitativa de tareas. Motivó a los grupos a mejorar constantemente y a superar obstáculos. Brindó sugerencias constructivas para optimizar el rendimiento de los proyectos y alentó a los estudiantes a asumir desafíos que expandieran sus habilidades.

Promovió la autonomía al proporcionar las herramientas y el conocimiento necesario para que los estudiantes resolvieran problemas por sí mismos. Estimuló la iniciativa y la responsabilidad en el proceso de aprendizaje.

Al llegar al final de esta unidad de aprendizaje, el docente decidió llevar a cabo una actividad que no solo destacaría los logros de los estudiantes, sino que también les proporcionaría la oportunidad de compartir sus proyectos y experiencias con sus compañeros. Esta actividad se denominó "Tecno Bosco", un evento que transformaría el aula en un escenario de presentación y aprendizaje compartido.

En las sesiones previas a Tecno Bosco, los grupos de estudiantes habían estado trabajando diligentemente en sus proyectos, aplicando sus conocimientos recién adquiridos sobre algoritmos, pensamiento algorítmico y programación. Cada grupo había definido un requerimiento específico y había diseñado un prototipo que abordara ese requerimiento de manera creativa y funcional.

Figura 16. Evidencia de preparación de la exposición Tecno Bosco.



Fuente. Elaboración propia.

El día del evento, el aula se transformó en un espacio destinado a una feria o evento expositivo, a cada uno de los grupos se le asignó una mesa a modo de stand y se organizaron para que permitiera visitarlos todos. Los estudiantes llegaron con sus prototipos cuidadosamente preparados y una presentación en mente. A medida que sus propios compañeros o estudiantes de otros cursos los visitaban daban inicio a su presentación explicando el requerimiento que habían abordado. Describieron el problema o desafío que querían resolver y por qué lo habían elegido en particular. Entre las ideas seleccionadas se pueden nombrar, la creación de un candado digital, un sistema de monitoreo de la temperatura ambiente, un juego de agilidad física o un juego de preguntas y respuestas.

Luego, presentaron su solución, que en la mayoría de los casos implicaba el uso creativo de la tarjeta Micro:bit y la programación. Los estudiantes demostraron cómo funcionaba su proyecto, paso a paso. Explicaron las instrucciones que habían programado en la tarjeta Micro:bit y cómo estas instrucciones permitían que el prototipo respondiera al requerimiento. También resaltaron las validaciones que habían realizado para garantizar que su solución funcionara correctamente. Cada grupo compartió su experiencia en el proceso de diseño y construcción, hablaron sobre los desafíos que enfrentaron y cómo los superaron. Estos desafíos no solo estaban relacionados con la comprensión de los conceptos inherentes a los algoritmos, sino también con la capacidad de expresar sus ideas de manera efectiva mediante bloques de código. En este proceso, se evidenció la importancia de la creatividad y la resiliencia, ya que debieron superar obstáculos conceptuales y técnicos. Uno de los mayores inconvenientes que describieron fue la falta de habilidad para plantear una expresión matemática y representarla mediante bloques de código.

Figura 17. Evidencia de la exposición Tecno Bosco.



Fuente. Elaboración propia.

Reflexionaron sobre lo que habían aprendido, no sólo en términos de programación y tecnología, sino también sobre habilidades como la colaboración y la resolución de problemas, dieron informe sobre las tareas que cada uno cumplió en su grupo, desde la fase de diseño, programación, pruebas y presentación.

Los estudiantes evaluaron su propio nivel de conocimiento alcanzado durante la unidad aplicando la encuesta compartida por el docente y de igual manera evaluaron a cada uno de sus compañeros bajo los criterios presentados por el docente en la encuesta de coevaluación preparada para tal fin. Expusieron cómo habían mejorado sus habilidades de

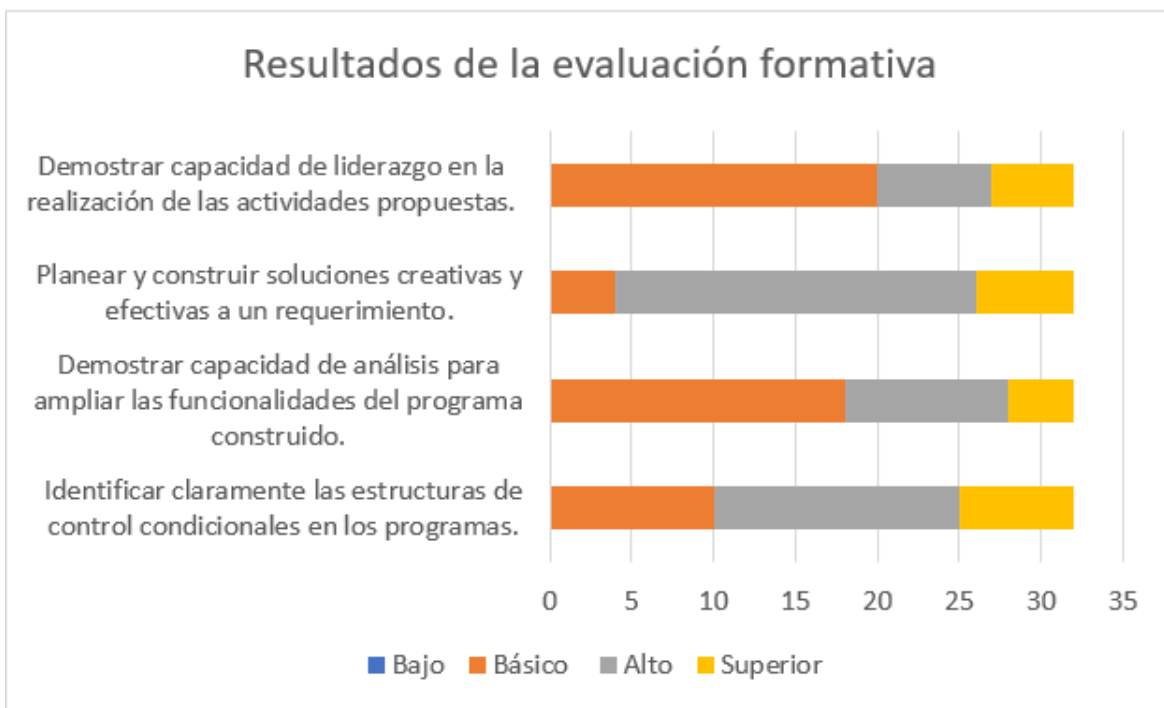
programación y cómo habían aplicado el pensamiento algorítmico en sus proyectos.

También consideraron cómo podrían seguir desarrollando estas habilidades en el futuro.

El evento "Tecno Bosco" no solo brindó una oportunidad para que los estudiantes presentaran sus proyectos, sino que también fomentó la retroalimentación y el aprendizaje entre pares. Los compañeros tenían la oportunidad de hacer preguntas, ofrecer sugerencias y aprender de los enfoques creativos de otros grupos. El docente desempeñó un papel activo como moderador, guiando las discusiones y alentando la participación. También reconoció y elogió los logros de los estudiantes, destacando sus esfuerzos y su crecimiento en el pensamiento algorítmico y la programación. En última instancia, Tecno Bosco fue un testimonio de cómo la combinación de la programación, la creatividad y un enfoque pedagógico efectivo puede inspirar y empoderar a los estudiantes, al tiempo que fomenta habilidades esenciales para el siglo XXI, como la resolución de problemas, la comunicación y la colaboración.

A continuación, se presentan los resultados de aplicar la rúbrica de evaluación formativa y la encuesta de autoevaluación, este resumen brinda una visión valiosa del progreso y destaca algunas áreas donde hemos alcanzado un alto nivel de desempeño.

Figura 18. Resultados evaluación formativa.

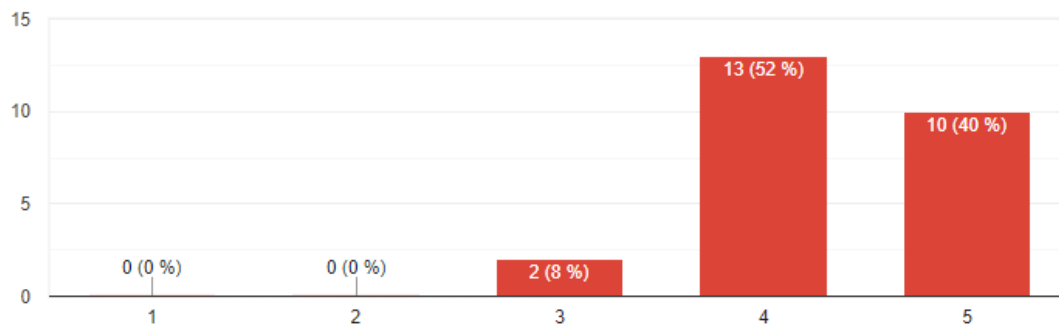


Fuente. Elaboración propia.

Figura 19. Resultados autoevaluación desempeños.

¿Cómo calificarías tu nivel de comprensión de los conceptos básicos de programación antes y después del curso?

25 respuestas

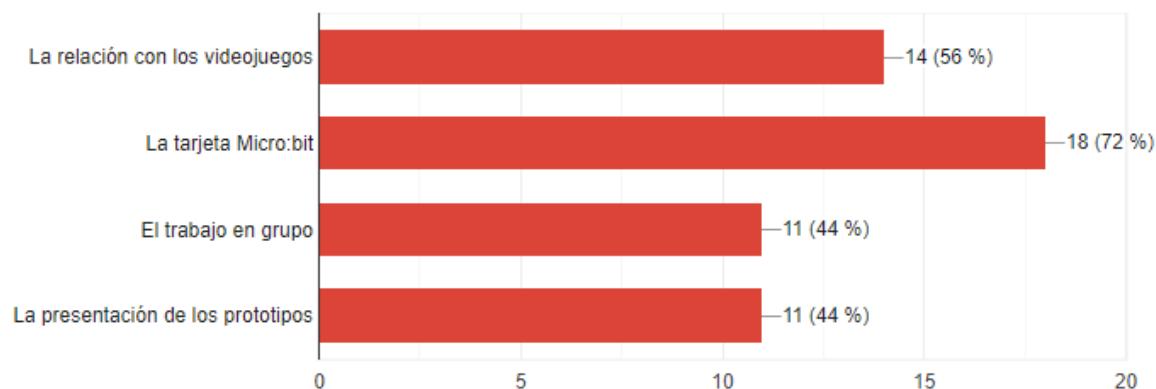


Fuente. Elaboración propia.

Figura 20. Resultados autoevaluación intereses.

¿Cuáles fueron los aspectos más destacados del curso para ti? ¿Por qué?

25 respuestas



Fuente. Elaboración propia.

6. Fase 4. El análisis, interpretación y reflexión del relato producto de la recuperación y reconstrucción de la práctica.

6.1. Uso de plataformas de programación en bloques y dispositivos interactivos para el desarrollo del pensamiento algorítmico como estrategia de aprendizaje en una fase introductoria.

La decisión de incorporar recursos tecnológicos en una práctica educativa debe basarse en una comprensión sólida de la intersección de tres componentes esenciales, según el modelo TPACK de Mishra y Koehler (2006). Estos componentes son el conocimiento tecnológico (TK), el conocimiento pedagógico (PK) y el conocimiento del contenido (CK).

En el caso de la práctica educativa que estamos analizando, se evidencia que para el componente de CK, cuyo contenido específico es el pensamiento algorítmico, el docente tiene un conocimiento sólido de este concepto que le permite enseñarlo de manera efectiva. Para el PK, el docente ha tomado la responsabilidad de profundizar en su conocimiento

sobre los procesos de enseñanza y aprendizaje, lo que le permitió diseñar una actividad desafiante y motivadora para los estudiantes. En cuanto al TK, el docente basado en su formación profesional y la habilidad de autoaprendizaje posee un conocimiento avanzado de los recursos seleccionados, que lo avalan para utilizarlos de manera efectiva para enseñar el contenido.

El uso de la tarjeta Micro:bit como herramienta pedagógica en el desarrollo del pensamiento algorítmico en una fase introductoria se alinea de manera correcta y efectiva con el enfoque de educación maker. A través de un análisis de la práctica realizada, se pueden destacar varios aspectos clave que respaldan esta estrategia de aprendizaje. El ambiente maker se basa en los aspectos del aprendizaje activo y experiencial, los estudiantes no solo recibieron información teórica sobre algoritmos, sino que también participaron activamente en la creación y ejecución de algoritmos utilizando la tarjeta Micro:bit esto les permitió construir conceptos abstractos de una manera tangible y práctica. La programación de la tarjeta Micro:bit requirió que los estudiantes analicen problemas, subdividan tareas en pasos y anticipen resultados, este proceso fomenta los pensamientos algorítmico y computacional, habilidades esenciales en el mundo digital actual. Los proyectos de programación en Micro:bit se realizaron en grupos, los estudiantes aprendieron a comunicarse y a combinar sus habilidades para lograr un objetivo común, lo que es esencial en un entorno educativo moderno. La interacción con la tarjeta Micro:bit, motivó intrínsecamente a los estudiantes, brindándoles la excusa perfecta para que se sintieran empoderados al ver resultados tangibles de su trabajo, aumentando su compromiso y entusiasmo por aprender. El proceso de programación en Micro:bit permitió una evaluación continua a medida que los estudiantes avanzaban en sus proyectos, el docente

pudo proporcionar retroalimentación oportuna para el desarrollo de las competencias propuestas.

6.2. ¿Cuáles son las actividades implementadas por el docente que benefician el desarrollo del pensamiento algorítmico?

Teniendo en cuenta que el enfoque pedagógico seleccionado es la educación maker, las dimensiones descritas en Exploratorium y los principios de diseño definidos por Brahms y Wardrip (2014), las decisiones tomadas por el docente han beneficiado significativamente el desarrollo del pensamiento algorítmico y el aprendizaje de los estudiantes. La elección de adoptar un enfoque de educación maker ha transformado el aula en un entorno donde los estudiantes son los protagonistas de su aprendizaje. Se les ha proporcionado la oportunidad de explorar, crear y aprender a través de la práctica activa, lo que fomenta el compromiso y la motivación intrínseca.

La educación maker promueve la creatividad al alentar a los estudiantes a diseñar y construir proyectos según sus propias ideas. Esto ha llevado a una mayor experimentación y exploración de soluciones, lo que a su vez ha desarrollado la capacidad de los estudiantes para pensar de manera innovadora y resolver problemas de manera independiente.

Los principios de diseño recomendados por Brahms y Wardrip subrayan la importancia del diseño de actividades colaborativas. Esto ha permitido a los estudiantes trabajar juntos en proyectos, compartir ideas y conocimientos, y aprender unos de otros. La colaboración ha mejorado su comprensión y habilidades comunicativas.

El enfoque maker ha brindado a los estudiantes la oportunidad de reflexionar sobre sus proyectos y procesos de diseño. Han aprendido a evaluar y mejorar sus creaciones, lo

que fomenta el pensamiento crítico y la autorreflexión. La incorporación de herramientas como la tarjeta Micro:bit y la plataforma MakeCode ha permitido a los estudiantes relacionar conceptos de programación con proyectos prácticos. Esto ha facilitado la aplicación de pensamiento algorítmico y la adquisición de habilidades técnicas relevantes. Al ver sus proyectos tomando forma y resolviendo problemas del mundo real, los estudiantes han experimentado una mayor motivación intrínseca. El sentido de logro al completar proyectos ha reforzado su autoeficacia y su confianza en sus habilidades.

El docente seleccionó una herramienta TIC interactiva, en este caso, la tarjeta Micro:bit, como plataforma para enseñar programación y algoritmos. Esta elección es beneficiosa porque brinda a los estudiantes una interfaz práctica y visual para experimentar con algoritmos. Los entornos de programación como Microbit suelen ser amigables para principiantes y permiten a los estudiantes ver cómo funcionan los algoritmos en tiempo real, lo que facilita la comprensión de los conceptos.

El docente presentó ejemplos de programas o algoritmos en la tarjeta Micro:bit. Esta estrategia es efectiva porque proporcionó a los estudiantes modelos concretos de cómo se estructuran los algoritmos y cómo se implementan en la tarjeta. Al permitir que los estudiantes modifiquen y experimenten con estos ejemplos, están practicando la modificación de algoritmos existentes, lo que desarrolla su capacidad de comprender y editar algoritmos de manera efectiva.

En conjunto, estas actividades promovieron el desarrollo del pensamiento algorítmico al proporcionar a los estudiantes experiencias prácticas y significativas en las que pudieron aplicar y refinar sus habilidades de programación, comprender cómo

funcionan los algoritmos y aprender a resolver problemas mediante la creación de algoritmos efectivos.

6.3. ¿Cómo se puede evidenciar el uso de las TIC como elementos de mediación en el aprendizaje del pensamiento algorítmico?

El docente, a lo largo de experiencias previas para abordar los mismos objetivos de aprendizaje, basado en su propio recorrido como estudiante, optó por la implementación de actividades que se alineaban con enfoques pedagógicos más tradicionales. Estas actividades se centraron en la presentación de conceptos abstractos a través de métodos convencionales, como el uso de diagramas de flujo y pseudocódigo. En este enfoque, el uso de herramientas o recursos tecnológicos se dejaba de lado en gran medida.

Este enfoque tradicional se caracteriza por una presentación teórica y abstracta de los conceptos, lo que aleja a los estudiantes de una comprensión práctica y significativa. El énfasis recae en la representación visual de algoritmos a través de diagramas de flujo o en la descripción textual mediante pseudocódigo, lo que, aunque es fundamental en la programación, puede resultar abstracto y desafiante para algunos estudiantes, especialmente aquellos que son nuevos en este campo.

Sin embargo, a medida que se evoluciona en la pedagogía, se observó la necesidad de adoptar un enfoque más centrado en el estudiante y en el uso efectivo de recursos tecnológicos. Este nuevo enfoque se alinea con la tendencia educativa que aboga por la práctica activa, permitiendo a los estudiantes involucrarse de manera más práctica y creativa en la construcción de su comprensión.

El docente reconoció que las metodologías tradicionales no siempre eran efectivas para todos los estudiantes, especialmente en el contexto de la enseñanza de la programación. Por lo tanto, se decidió explorar enfoques más innovadores y recursos tecnológicos, como la tarjeta Micro:bit, para mejorar la comprensión y el compromiso de los estudiantes en el proceso de aprendizaje. Esto marcó un cambio importante en la forma en que se abordan los objetivos de aprendizaje y representa un compromiso con prácticas pedagógicas más efectivas y centradas en el estudiante.

De acuerdo al modelo SAMR para la integración efectiva de las TIC en el proceso de enseñanza-aprendizaje, en este caso en particular la tarjeta Micro:bit y la plataforma MakeCode, han demostrado la capacidad de transformar la enseñanza de manera significativa, llevando las actividades a un nivel de redefinición. Como lo menciona Puentedura (2006), al avanzar en la escala propuesta por el modelo SAMR, se alcanza un nivel de redefinición cuando las TIC permiten la creación de nuevas posibilidades educativas que no eran concebibles sin ellas. En el contexto de la enseñanza de la programación, la tarjeta Micro:bit y la plataforma MakeCode han revolucionado la forma en que los estudiantes abordan y comprenden los conceptos del pensamiento algorítmico.

Las herramientas TIC seleccionadas permitieron a los estudiantes interactuar directamente con las herramientas tecnológicas, como la tarjeta Micro:bit, para diseñar, crear y probar algoritmos. Los estudiantes pudieron ver cómo sus instrucciones se traducen en acciones concretas en el entorno digital, facilitando la comprensión de cómo los algoritmos afectan el comportamiento de un sistema y cómo se pueden modificar para lograr resultados específicos.

El entorno de programación visual usado en la plataforma MakeCode, en el cual los estudiantes pueden arrastrar y soltar bloques de código para crear algoritmos en lugar de escribir código complejo, permite que el proceso sea más accesible para los principiantes y les permite enfocarse en los aspectos del pensamiento algorítmico en lugar de preocuparse por la sintaxis de programación.

El uso de los recursos TIC permitieron generar una retroalimentación inmediata a los estudiantes. Cuando crean y ejecutan un algoritmo, pueden ver de inmediato los resultados en la pantalla de la tarjeta Micro:bit, esta respuesta ayuda a los estudiantes a comprender cómo funcionan los algoritmos y a depurar cualquier error que puedan encontrar.

El diseño de las actividades de enseñanza en conjunción con las TIC fomenta la experimentación y la creatividad. Los estudiantes pueden probar diferentes vías para resolver problemas y ver cómo se comportan los algoritmos en situaciones diversas. También tienen la libertad de ser creativos al diseñar algoritmos para solucionar desafíos que les interesen.

7. Conclusiones

El proceso de reflexión sobre los procesos de diseño e implementación de la práctica educativa ha puesto de manifiesto el compromiso del docente en su búsqueda constante por mejorar su labor pedagógica. Reconociendo que los procesos de enseñanza son acciones complejas que requieren una planificación y ejecución cuidadosas, el docente ha asumido una actitud positiva frente a la autoevaluación y mejora continua, esta reflexión ha sido impulsada por la búsqueda de estrategias efectivas que realmente motiven a los estudiantes y les permitan comprender y aplicar los conceptos de manera significativa. A través de esta autoevaluación, el docente ha reevaluado su enfoque anterior basado en métodos tradicionales y ha incorporado enfoques más innovadores y centrados en el estudiante. En este sentido, el docente ha reconocido la importancia de adaptar su enfoque a las necesidades y estilos de aprendizaje de los estudiantes.

La brecha entre los principios de diseño y la realidad de la implementación es un aspecto crítico que ha surgido en el análisis de esta práctica educativa, si bien los principios de diseño inspirados en la educación maker, enfatizan la autonomía del estudiante y un nivel mínimo de instrucción para promover la exploración y el aprendizaje autodirigido, es evidente que el grupo de estudiantes que participaron en el proceso de sistematización aún requiere instrucciones detalladas. Esta discrepancia sugiere varias reflexiones importantes, en primer lugar, es esencial reconocer que cada grupo de estudiantes es único y sus necesidades y capacidades pueden variar ampliamente, mientras que el diseño de la práctica educativa se basa en principios generales, el docente debió ser flexible y capaz de adaptar la instrucción según las necesidades específicas observadas en el aula. En segundo lugar, este hallazgo destaca la importancia de una retroalimentación continua en cada una de las

actividades y la observación sobre cómo los estudiantes interactúan con la tecnología y responden a las actividades para proporcionar información valiosa sobre la efectividad de la instrucción y la necesidad de ajustes. También, es fundamental considerar que la autonomía del estudiante es un objetivo que se desarrolla con el tiempo, en una fase introductoria los estudiantes pueden no estar completamente preparados para la independencia total, por lo tanto, la práctica educativa permite dar el primer paso hacia la construcción de la confianza y la competencia necesarias para el aprendizaje autodirigido en el futuro.

El análisis de esta práctica educativa reveló un aspecto alentador y valioso: la concordancia entre el uso planeado de los recursos TIC y el uso real que los estudiantes hicieron de ellos. Este hallazgo subraya el éxito en la planificación y ejecución de la práctica, así como la efectividad de la estrategia pedagógica que se ha seguido. Mantener esta concordancia implica una comprensión sólida de las necesidades de los estudiantes, el contexto en el cual se desarrolló el proceso y una cuidadosa selección de recursos TIC que se alineen con los objetivos de aprendizaje. La concordancia entre el uso planeado y el uso real también refleja la capacidad del docente para guiar y facilitar el aprendizaje de los estudiantes de manera efectiva, lo que viene a demostrar el equilibrio entre los tres componentes del modelo TPACK (TK, PK y CK).

El proceso de autoevaluación permite evidenciar un logro sobresaliente en la elección y aplicación de recursos TIC en la práctica educativa. Los recursos TIC, en este caso, no sólo desempeñaron un papel de mediadores cognitivos, sino que también ejercieron un impacto positivo en la motivación de los estudiantes, lo que es un hito fundamental en el proceso de enseñanza-aprendizaje. El hecho de que los recursos TIC hayan actuado como mediadores cognitivos implica que cumplieron su función principal de

facilitar la comprensión de conceptos y habilidades relacionados con la programación y el pensamiento algorítmico. Estos recursos proporcionaron herramientas interactivas, ejemplos prácticos y oportunidades de práctica que ayudaron a los estudiantes a internalizar mejor estos conceptos. Al hacer que el contenido sea más accesible y comprensible, los recursos TIC contribuyeron a un aprendizaje más efectivo y significativo. Además de su función cognitiva, es especialmente destacable que los recursos TIC también motivaron a los estudiantes en su proceso de aprendizaje, siendo este un factor crítico en la educación, ya que puede influir en la participación, el compromiso y, en última instancia, en el rendimiento de los estudiantes. Los recursos TIC utilizados en esta práctica educativa lograron despertar el interés de los estudiantes al proporcionar experiencias de aprendizaje atractivas y participativas, evidenciado en el entusiasmo que los estudiantes mostraron al interactuar con las tarjetas Micro:bit y al abordar desafíos a través de la plataforma MakeCode.

En el contexto de la inclusión de las TIC en esta experiencia de aprendizaje, se considera que se ha alcanzado el nivel de redefinición, según el modelo SAMR, teniendo en cuenta que los recursos TIC han transformado (respecto a las anteriores implementaciones) drásticamente el proceso de enseñanza y han brindado nuevas capacidades a los estudiantes para generar cambios significativos en algunas dimensiones como creatividad, autonomía e interés.

El análisis cualitativo a los resultados de aprendizaje y las opiniones de los estudiantes al finalizar el proceso son muy prometedores ya que respaldan el impacto positivo que tuvo la práctica educativa en el aprendizaje del pensamiento algorítmico, especialmente en comparación con estrategias de enseñanza previas que emplearon

diagramas de flujo o pseudocódigo. Este análisis proporciona evidencia de que la introducción de la tarjeta Micro:bit y la plataforma MakeCode en el proceso de enseñanza y aprendizaje resultó en una mejora significativa en la comprensión y aplicación de conceptos algorítmicos ya que les permitió experimentar con conceptos abstractos de una manera más concreta y práctica. El enfoque de aprendizaje maker, en el que los estudiantes pueden ver de inmediato el impacto de sus acciones en el mundo físico a través de la tarjeta Micro:bit, parece haberles ayudado a internalizar mejor los conceptos de algoritmos y pensamiento algorítmico, esto se tradujo en una mejora en su capacidad para diseñar algoritmos, resolver problemas y pensar de manera algorítmica. La comparación con experiencias de aprendizaje anteriores sugiere que los estudiantes encontraron más atractiva y accesible esta forma de aprender, generando mayor participación y motivación, esto puede servir como un modelo inspirador para otros docentes que deseen implementar actividades similares en sus aulas.

Otro aspecto a resaltar de la implementación de la estrategia de aprendizaje fue la eficaz utilización de los recursos donados por el estado en el contexto educativo. Esta práctica demuestra que estos recursos no solo pueden ser adquiridos y almacenados, sino que pueden ser aprovechados de manera real y efectiva en beneficio del proceso de enseñanza y aprendizaje. En muchas ocasiones, es común que las instituciones educativas reciban recursos, ya sean tecnológicos o materiales, destinados a mejorar la calidad de la educación, sin embargo, estos recursos pueden quedar subutilizados o, peor aún, almacenados sin un propósito específico. Al lograr alinear la motivación y determinación del docente para innovar sus prácticas, la confianza de los directivos docentes para alentar estas transformaciones, esta práctica educativa desafía esta tendencia al demostrar que estos

recursos pueden ser incorporados en el proceso de enseñanza de una manera que enriquezca la experiencia de aprendizaje de los estudiantes.

La creación del semillero de investigación en robótica educativa es un testimonio de la inspiración que esta práctica ha generado en el docente y los estudiantes. La experiencia de trabajar con la tarjeta Micro:bit y la plataforma MakeCode despertó la curiosidad y el interés por la tecnología y la programación. Esta inquietud condujo naturalmente al deseo de explorar más a fondo estos temas y abordar cuestiones más profundas en el campo de la robótica educativa. Un semillero proporciona un espacio donde los estudiantes pueden llevar a cabo investigaciones más profundas, experimentar con nuevas ideas y desarrollar proyectos más avanzados. Además, ofrece oportunidades para colaborar, compartir conocimientos y aprender de manera colaborativa. Esta iniciativa no solo enriquece la experiencia de aprendizaje de los estudiantes, sino que también fomenta la autonomía y la creatividad.

8. Referencias bibliográficas

- Alzate Sánchez, A. (2021). Sistematización de una experiencia de aprendizaje basado en retos y gamificación para promover el desarrollo del pensamiento aleatorio y de sistemas de datos en los estudiantes de grado décimo de la Institución Educativa General Santander, de la ciudad de Granada-Meta, en el año escolar 2021.
- Andersen, R., & Rustad, M. (2022). Using Minecraft as an educational tool for supporting collaboration as a 21st century skill. *Computers and Education Open*, 3, 100094.
- Ayala Rodríguez, G. I. (2022). El pensamiento computacional y el pensamiento algorítmico como habilidad para potencializar la expresión oral y escrita en la enseñanza escolar con el uso de la mediación de las TIC.
- Brahms, L., & Wardrip, P. (2014). The learning practices of making: An evolving framework for design. *Children's Museum of Pittsburgh*.
- Daza Alvarado, J. K., & Sánchez Fernández, F. E. (2022). Uso de un ambiente de programación visual y el aprendizaje basado en retos para desarrollar el pensamiento algorítmico en estudiantes del grado noveno de la institución educativa los comuneros de la ciudad de Popayán, Cauca.
- Enriquez, N. R. P. (2023). Implementación efectiva de las TIC en la educación para mejorar el aprendizaje: una revisión sistemática. *Ciencia Latina Revista Científica Multidisciplinar*, 7(1), 5788-5804.
- Espitia, A. S., Sánchez, M. Y. P., & Cano, S. P. A. (2022). Categorización de las TIC para la mediación pedagógica en el aula de clase. In *Actas del Congreso de Investigación, Desarrollo e Innovación* (pp. 128-146).

Esquea Arrieta, M. (2021). Estrategia de aprendizaje basado en problemas con el uso de scratch para promover el desarrollo de habilidades del pensamiento computacional en los estudiantes de 10^oa de la IE Virginia Gómez del municipio de Ciénaga, Magdalena en el año escolar 2021.

Eteng, I., Akpotuzor, S., Akinola, S. O., & Agbonlahor, I. (2022). A review on effective approach to teaching computer programming to undergraduates in developing countries. *Scientific African*, e01240.

Jara Holliday, O. (2020). Orientaciones teórico prácticas para la sistematización de experiencias.

Kožuh, I., Krajnc, R., Hadjileontiadis, L. J., & Debevc, M. (2018). Assessment of problem solving ability in novice programmers. *PLOS ONE*, 13(9), e0201919.

Krueger, N. (s/f). Cree un Makerspace escolar en 3 sencillos pasos. Iste.org. Recuperado el 10 de abril de 2023, de <https://www.iste.org/es/explore/classroom/create-school-makerspace-3-simple-steps>

López, Á., & Marcos, M. (2018). Una experiencia de clase invertida en la enseñanza de la programación.

Martínez, C. C., & Vinuesa, T. S. (2018). Estudio de caso sobre la enseñanza de la programación en primaria. In *EDUCación con TECnología: un compromiso social. Aproximaciones desde la investigación y la innovación* (pp. 1493-1499). Edicions de la Universitat de Lleida.

Mishra, P., & Koehler, M. J. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers college record*, 108(6), 1017-1054.

- Mondragón Reyes, J. F. (2019). Desarrollo de habilidades de pensamiento algorítmico basado en la gamificación en estudiantes del grado noveno.
- Papert, S. (2005). You can't think about thinking without thinking about thinking about something. *Contemporary Issues in Technology and Teacher Education*, 5(3/4), 366-367.
- Partovi, H. (2020). Las escuelas deben enseñar el currículo del futuro, no del pasado. *Telos: Cuadernos de comunicación e innovación*, (114), 62-69.
- Pedro Jose, A. T., & Brigas, C. (2022). Computational thinking in early childhood education: an analysis through the Computer Science Unplugged. *Revista interuniversitaria de formación del profesorado*.
- Pérez, D. F. P., González, M. R., & Palacio, E. V. G. (2023). El pensamiento algorítmico como estrategia didáctica para el desarrollo de habilidades de resolución de problemas en el contexto de la educación básica secundaria. *Revista de Educación a Distancia (RED)*, 23(73).
- Rayna, T., & Striukova, L. (2021). Fostering skills for the 21st century: The role of Fab labs and makerspaces. *Technological Forecasting and Social Change*, 164, 120391.
- Rivera Morcillo, I. A. (2020). Aprendizaje basado en retos con mediación de las TIC, una oportunidad para desarrollar el pensamiento computacional.
- Rosenheck, L. (2020). Espacios maker para hacer visible el aprendizaje de los alumnos. *Observatorio / Instituto para el Futuro de la Educación; Instituto para el Futuro de la Educación*. <https://observatorio.tec.mx/edu-bits-blog/espacios-maker-aprendizaje-visible/>

- Rueda-Ortiz, R., & Franco-Avellaneda, M. (2018). Políticas educativas de TIC en Colombia: entre la inclusión digital y formas de resistencia-transformación social. *Pedagogía y saberes*, (48), 9-25.
- Simović, V., Veskovic, M., & Purenovic, J. (2022). Micro: bit as a new technology in education in primary schools. In *Proceedings TIE 2022 9th International Scientific Conference Technics and Informatics in Education*. University of Kragujevac, Faculty of Technical Sciences, Čačak.
- Soza, M. G. M. (2020). TPACK para integrar efectivamente las TIC en educación: Un modelo teórico para la formación docente. *Revista electrónica de conocimientos, saberes y prácticas*, 3(1), 133-148.
- Stephens, M., & Kadjevich, D. M. (2020). Computational/Algorithmic Thinking. En S. Lerman (Ed.), *Encyclopedia of Mathematics Education* (pp. 117-123). Springer International Publishing
- Voštinár, P., & Knežník, J. (2020). Experience with teaching with BBC micro: bit. In *2020 IEEE Global Engineering Education Conference (EDUCON)* (pp. 1306-1310). IEEE.