ISBN: 978-628-96613-1-6. ISSN: 2414-6390. Digital Object Identifier: https://dx.doi.org/10.18687/LACCEI2025.1.1.1072

Visual Programming with Scratch To Strengthen Computational Thinking In Engineering Education

Rumbaut Dayron, M. Sc[1], Echeverria Angel, Ph. D.[1,], Maridueña Milton[1, 2],
Tolozano Gabriela, M. Sc[1], Valderrey Michael Ph. D[3]

[1]Universidad Bolivariana de Ecuador, (UBE).
[2] Universidad de Guayaquil (UG).
[3]Universidad Latinoamericana y del Caribe (ULAC).

Abstract- In the context of higher education, computational thinking has been identified as a key skill for addressing complex problems across various disciplines. For this reason, the present research aimed to evaluate the effectiveness of implementing a visual programming training program using Scratch to strengthen computational thinking in mechanical and electrical engineering students. The program was conducted at two higher education institutions: the Instituto Tecnológico Bolivariano de Tecnología (ITB) and the Universidad Bolivariana del Ecuador (UBE), involving 240 students from the mechanical and electrical engineering programs. The training program was developed following the ADDIE instructional model. Evaluation was carried out through detailed rubrics and satisfaction surveys, utilizing platforms such as Moodle for implementation. Among the main results, 85% of the students developed functional projects that integrated advanced computational thinking concepts, while 92% completed all proposed activities. The interdisciplinary projects, covering areas such as economics and physics, highlighted the applicability of programming in real-world contexts, fostering knowledge transfer. Scratch proved to be an effective tool for teaching programming and computational thinking at advanced levels, promoting both technical and creative skills. The implemented pedagogical strategies increased student motivation and engagement. The results suggest the course's replicability in other educational institutions.

Keywords: Computational Thinking, Higher Education, Visual Programming, Scratch

Programación visual con scratch para fortalecer el pensamiento computacional en la formación ingenieril

Rumbaut Dayron, M. Sc[1], Echeverria Angel, Ph. D.[1,], Maridueña Milton[1, 2],
Tolozano Gabriela, M. Sc[1], Valderrey Michael Ph. D[3]

[1]Universidad Bolivariana de Ecuador, (UBE).
[2] Universidad de Guayaquil (UG).
[3]Universidad Latinoamericana y del Caribe (ULAC).

Resumen- En el contexto de la educación superior, el pensamiento computacional se ha identificado como una habilidad clave para abordar problemas complejos en diversas disciplinas. Por tal motivo, la presente investigación tuvo como propósito evaluar la efectividad de la implementación de un programa de formación en programación visual con Scrath para fortalecer el pensamiento computacional en los estudiantes de ingeniería mecánica y eléctrica. El mismo se llevó a cabo en dos instituciones de educación superior a saber, el Instituto Tecnológico Bolivariano de Tecnología (ITB) y la Universidad Bolivariana del Ecuador (UBE), y contó con la participación de 240 estudiantes miembros de la carrera de mecánica y electricidad. El programa de formación se desarrolló siguiendo el modelo instruccional ADDIE. La evaluación se realizó mediante rúbricas detalladas y encuestas de satisfacción, utilizando plataformas como Moodle para la implementación. Entre los principales resultados se obtuvo que el 85% de los estudiantes desarrolló proyectos funcionales que integraron conceptos avanzados de pensamiento computacional, mientras que el 92% completó todas las actividades propuestas. Los proyectos interdisciplinarios, que abarcaron áreas como economía y física, destacaron la aplicabilidad de la programación en contextos reales, fomentando la transferencia del conocimiento. Scratch se demostró como una herramienta efectiva para enseñar programación y pensamiento computacional en niveles avanzados, promoviendo habilidades técnicas y creativas. Las estrategias pedagógicas implementadas incrementaron la motivación y el compromiso de los estudiantes. Los resultados sugieren la replicabilidad del curso en otras instituciones educativas.

Palabras clave: Pensamiento computacional, Educación superior, Programación visual, Scratch.

I. Introducción

En un contexto donde las competencias digitales son cruciales para la mayoría de las disciplinas, el pensamiento computacional emerge como una habilidad clave para los ciudadanos del siglo XXI. Hoy en día, es difícil encontrar una ocupación o pasatiempo en el que los trabajadores y la tecnología no interactúen [1]. Por ello, en la formación del nuevo ciudadano, y de manera específica en el sistema educativo superior, la manera como se está llevando el proceso formativo, ha transitado de una metodología tradicional y física a una metodología digital, donde ha surgido un nuevo concepto: pensamiento computacional.

Históricamente, el concepto "pensamiento computacional" ha estado en constante evolución, después de una revisión sistémica, podría sintetizarse su definición teniendo en cuenta su origen y evolución, desde la siguiente perspectiva:

- Como la habilidad que implica resolver problemas, diseñar sistemas, y entender el comportamiento humano a partir de los conceptos fundamentales de la informática [2]
- Como un enfoque para resolver un problema concreto que ayuda a la inclusión de tecnologías digitales con ideas humanas [1].
- Como el proceso de reconocimiento de los aspectos computables en el mundo que nos rodea [3]
- Un tipo de pensamiento basado en procesos ejecutados por una persona o una máquina utilizando métodos y modelos que permiten resolver problemas, así como diseñar sistemas que por sí solos no podrían hacerlo [4]
- El principal potencial es la posibilidad de expresar y crear ideas mientras programamos, argumentando que la programación, al igual que la escritura, es una forma de expresarse [5]

Todas estas definiciones han permitido a la educación superior entender que, el proceso de enseñanza aprendizaje no transita de manera mecanizada, sino que la misma debe llevar al formando adquirir competencias en la resolución de problemas, por eso se han implementado metodologías activas con dicho enfoque, donde el proceso se entiende como "la capacidad de formular problemas y sus soluciones de tal manera que un agente de procesamiento de información pueda ejecutarlas eficazmente" [2]. Por ello, fomentar estas competencias permite a los estudiantes enfrentar los retos de la era digital de forma más efectiva, dotándolos de herramientas para el análisis y solución de problemas complejos en sus respectivas áreas de estudio.

Esto ha hecho posible que, el pensamiento computacional no solo se muestre relevante para la informática, sino que también tiene aplicaciones fundamentales en disciplinas como la biología, la economía y las ciencias sociales [6], permitiendo a los estudiantes una comprensión más profunda de los problemas y una mayor capacidad para resolverlos de manera estructurada. El pensamiento computacional se manifiesta, entonces, como un enfoque para resolver un problema concreto que ayuda a la inclusión de tecnologías digitales con ideas humanas.

Esto llevó a ciertos autores a desarrollar, a inicios del siglo XXI, un software educativo llamado Scratch, donde aprender a programar y compartir sus propias creaciones provoca que estudiantes se desarrollen como pensadores computacionales, aprendiendo conceptos básicos a la vez que son capaces de desarrollar estrategias de resolución de problemas, diseño y

formas de colaboración [7], por ello la formación universitaria en carrera de ingenierías debe permitir que los estudiantes forjen estrategias y ejecuten tareas de tal forma que les permita afrontar problemas con eficacia y posibilidades de éxitos.

Por tanto, como un lenguaje de programación visual, Scratch creado por el Instituto de Tecnología de Massachusetts (siglas en MIT) ha sido ampliamente utilizado en niveles educativos debido a su enfoque intuitivo y accesible. El mismo permite a los estudiantes "pensar creativamente, razonar sistemáticamente y trabajar de manera colaborativa" [8], habilidades esenciales en el siglo XXI. Aunque su uso se ha centrado principalmente en la educación primaria y secundaria, el potencial de Scratch para la educación superior ha sido menos explorado.

Esta realidad es la evidenciada después de un proceso de diagnóstico desarrollado en dos instituciones de educación superior ecuatoriano, a saber, el Instituto Tecnológico Bolivariano de Tecnología (ITB) y la Universidad Bolivariana del Ecuador (UBE), por lo que, la presente investigación tiene como propósito evaluar la efectividad de la implementación de un programa de formación en programación visual con Scratch como una vía para fortalecer el pensamiento computacional en los estudiantes de ingeniería mecánica y eléctrica. Teniendo como fundamento lo que expresan algunos autores ante el beneficio de fortalecer el pensamiento computacional con Scratch, pues su "permite a los estudiantes pasar de ser meros consumidores de tecnología a convertirse en creadores activos, fomentando un aprendizaje significativo y motivador" [9].

II. METODOLOGÍA

Para dar cumplimiento al objetivo planteado, los investigadores emplearon el modelo instruccional ADDIE (Analysis, Design, Development, Implementation, Evaluation) como ruta metodológica, ampliamente reconocido por su enfoque sistemático para desarrollar programas educativos efectivos [10]. Cada fase se adaptó para satisfacer las demandas específicas de los estudiantes de educación superior, con un énfasis en la aplicación de conocimientos en situaciones del mundo real.

a- Fase de análisis

La fase de análisis permitió identificar las necesidades particulares de los estudiantes que se forman en el Instituto Tecnológico Bolivariano de Tecnología (ITB) y la Universidad Bolivariana del Ecuador (UBE), las cuales son instituciones de educación superior que se encuentran ubicadas en la provincia del Guayas, Ecuador, sus sedes se ubican en los cantones Guayaquil y Durán. La oferta académica de ambas instituciones es de tercer y cuarto nivel: tecnologías, ingenierías y masterados.

Debido al creciente uso de la tecnología y los videojuegos en los estudiantes, las instituciones buscan estrategias para aprovechar dicha situación en post del desarrollo de competencias en torno al uso de las Tecnologías de la Información y la Comunicación (TIC) así como habilidades transversales de otras materias.

Sin embargo, después de llevar a cabo la recolección de información y su triangulación, por medio de entrevistas a los docentes y aplicación de encuesta a los estudiantes, se pudo evidenciar que:

- Los estudiantes de la carrera de tecnología en mecánica e ingeniería eléctrica en su programa propedéutico, no reciben clases de programación, siendo ésta una habilidad necesaria a corto plazo en el mundo actual pues aprender a programar desarrolla el pensamiento computacional, permite mejorar capacidades como el cálculo numérico, la creatividad, el pensamiento lógico o las aptitudes verbales.
- Los estudiantes ingresan a las instituciones de educación superior sin conocimiento previo (que se haya desarrollado en su formación de bachillerato) sobre programación. Por tanto, al no aprender a programar en la etapa pre-universitaria, los estudiantes no entienden muchas veces cómo funciona el mundo digital, lo que les permitirá no aprovecharlo al máximo en sus estudios superiores.
- Se ha detectado que los estudiantes no desarrollan el proceso creativo por carecer de pensamiento computacional lo que les dificulta combinar la parte racional con la parte intuitiva.
- Y por último, se evidenció que los estudiantes carecen de habilidades para resolver problemas complejos, así como la toma de decisiones estructurada. Escasa adaptabilidad ante tareas tecnológicas o automatizada. Deficiencia para diseñar o seguir procesos estructurados. Débil pensamiento crítico y analítico. En fin, presentan debilidad en la creatividad para soluciones tecnológicas

Por lo que, desde el diagnóstico es necesario recalcar que aprender programación se ha convertido en una competencia fundamental para las nuevas profesiones del futuro es considerada una habilidad valiosa en la era digital. Pues, el mundo a raíz de la pandemia COVID- 19 provocada por el virus SARS-CoV-2 tuvo muchos cambios de paradigmas y la educación no quedo exenta de esta situación. Luego de la suspensión de las clases presenciales; en las instituciones se planificaron e implementaron estrategias educativas desde la virtualidad. Debido a esto lo que en su momento fue un reto el día de hoy es una fortaleza, ya que tanto alumnos, docentes y familiares están comprometidos con este tipo de enseñanza. El 100% de los estudiantes tienen acceso a internet y a un computador o celular inteligente.

En conclusión, esta fase de análisis permitió identificar las necesidades particulares de los estudiantes, lo que permitió que el diseño de la propuesta se centrara en la creación de actividades prácticas alineadas con los objetivos de aprendizaje [10]. La fase de desarrollo incluyó la creación de materiales didácticos interactivos, aprovechando la naturaleza visual de Scratch para facilitar el aprendizaje [8]. La implementación se realizó en un entorno virtual utilizando Moodle, y la evaluación se llevó a cabo mediante rúbricas detalladas para medir competencias clave [11].

b- Fase de diseño y desarrollo

En cuanto al diseño del programa de formación, el mismo se estructuró en tres módulos y contó con una duración de 80 horas académicas en total. Las actividades estuvieron distribuidas en sincrónicas, asincrónicas y autónomas, lo cual permitió la flexibilidad en el aprendizaje. Esto fue fundamental para acomodar los diferentes horarios y responsabilidades de

los estudiantes universitarios, promoviendo un aprendizaje autodirigido y autónomo.

TABLA 1.
Diseño y Desarrollo Del Programa De Formación en programación visual con Scratch

Módulo I: Introducción a Scratch.				
Objetivo: Manejar y comprender el entorno de trabajo del software Scratch por medio de juegos y ejemplos de la comunidad.				
Contenido	**Actividades**	**Objetivo terminal**	**Recursos**	**Responsables**
Unidad 1: Entorno de trabajo de Scratch. Registro e inicio de sesión. Interfaz de Scratch Creación de un proyecto nuevo. Proyectos de ejemplos de la comunidad. **Unidad 2:** Animaciones. Movimientos. Repetición de secuencias Movimientos aleatorios y colisiones. Comentarios. **Unidad 3:** Contando historias. Personajes. Conversaciones Escenas. Sonidos.	Taller de socialización que permita comprender los fundamentos teóricos de la de los Entorno de trabajo de Scratch.	Al finalizar este módulo, los estudiantes serán capaces de: • crear proyectos interactivos en Scratch, integrando conceptos básicos de animación, narrativa digital y programación visual. • Desarrollarán competencias en lógica computacional, creatividad y diseño, aplicando secuencias, bucles, eventos y otras herramientas de Scratch para resolver problemas y expresar ideas.	• Sala de reuniones. • Sala de computación • Equipo de proyección.	• Especialista en Scratch. • Autoridades institucionales. • Participantes
Módulo II: Estructuras de control.				
Objetivo: Diseñar y utilizar estructuras de control para generar secuencias lógicas lo cual le permite solucionar problemas computacionales.				
Contenido	**Actividades**	**Objetivo terminal**	**Recursos**	**Responsables**
Unidad 4: Orientando objetos. Control de rotación de objetos. Objetos gráficos (sprites). Sincronización de movimientos **Unidad 5:** Tomando decisiones. Condiciones. Bloques de selectivas simples. Bloques de selectivas dobles Bloque de selectivas anidadas **Unidad 6:** Bucles. Repitiendo una y otra vez. Bloque "repetir…" Bloque "por siempre". Bloque "repetir hasta que..." Bloque "esperar hasta que…"	Socializar el contenido respectivo. Tomando en cuenta actividades como trabajo en grupo, lluvias de ideas, entre otros.	Al término de este módulo, los estudiantes serán capaces de: • diseñar proyectos interactivos en Scratch que incluyan movimientos sincronizados, toma de decisiones dinámicas y estructuras repetitivas, utilizando objetos gráficos personalizados y lógica computacional avanzada. • Los proyectos permitirán resolver problemas, crear juegos y simular escenarios interactivos.	• Sala de reuniones. • Sala de computación • Equipo de proyección.	• Especialista en Scratch. • Especialista en pensamiento computacional • Participantes
Módulo III: Datos y eventos.				
Objetivo: Diseñar y crear un videojuego aplicando el razonamiento lógico y el pensamiento computacional para evaluar la capacidad de recopilación y análisis de información, descomposición de problemas, realización de algoritmos y procedimientos.				
Contenido	**Actividades**	**Objetivo terminal**	**Recursos**	**Responsables**
Unidad 7: Variables. Como almacenar datos. Concepto de variables. Listas. Control de score. **Unidad 8:** Sensores y eventos. Tipos de sensores. Bloques de sensores. Tipos de eventos. Bloques de eventos. **Unidad 9:** Proyecto Final. Programando un Juego. Diseño de juegos. "Juego tipo Pong" para uno y dos jugadores.	Socializar el contenido respectivo. Tomando en cuenta actividades como trabajo en grupo, lluvias de ideas, entre otros.	Al finalizar el módulo, los participantes serán capaces de: • Utilizar variables, sensores y eventos para desarrollar proyectos interactivos avanzados en Scratch, incluyendo el diseño y programación de un juego completo como "Pong". Fortaleciendo su capacidad para diseñar aplicaciones lúdicas y educativas	• Sala de reuniones. • Sala de computación • Equipo de proyección.	• Especialista en Scratch. • Especialista en pensamiento computacional • Participantes

Fuente: Los autores (2024)

c- Fase de implementación

Para la implementación del programa de formación, el mismo se llevó a cabo por medio de la plataforma Moodle, reconocida por sus capacidades para gestionar y organizar contenidos educativos y facilitar el seguimiento del progreso de los estudiantes [12]. Moodle fue complementada con herramientas como Prezi, Canva, Quizizz y Scratch para enriquecer la experiencia de aprendizaje, proporcionando una variedad de recursos que respondieran a diferentes estilos de aprendizaje [13].

Las estrategias de enseñanza usadas, fueron las siguientes:

- Se empleó el Aprendizaje Basado en Proyectos ABP para fomentar el diseño y desarrollo de proyectos aplicados, lo cual está alineado con la teoría constructivista que sugiere que el aprendizaje es más efectivo cuando los estudiantes participan activamente en la creación de conocimiento [14]. Esta estrategia permitió a los estudiantes contextualizar el aprendizaje y desarrollar habilidades para resolver problemas reales [15]

- Se utilizó la gamificación para incrementar la motivación y el compromiso de los estudiantes, siguiendo el enfoque de Deterding y otros [16] quienes sostienen que la incorporación de elementos de juego en contextos no lúdicos puede mejorar el rendimiento y la participación de los alumnos.

- Las actividades colaborativas fueron diseñadas para mejorar habilidades de comunicación y trabajo en equipo, fundamentales en el ámbito universitario [17]. El aprendizaje colaborativo se ha asociado con mejoras en el rendimiento académico y en la satisfacción de los estudiantes, especialmente en entornos de aprendizaje activo [18]

En cuanto a la evaluación, la misma se realizó mediante el uso de rúbricas para medir competencias en pensamiento computacional y creatividad, además de una encuesta de satisfacción para recoger la retroalimentación de los estudiantes [11]. Las rúbricas se diseñaron para proporcionar criterios claros y específicos que ayudaran a los estudiantes a comprender mejor las expectativas y a reflexionar sobre su propio desempeño. Además, la evaluación formativa se utilizó para proporcionar retroalimentación continua, lo cual es esencial para el desarrollo de habilidades complejas como el pensamiento computacional.

III. RESULTADOS

Este apartado se fundamenta con los resultados obtenidos una vez implementado el programa de formación visual con Scratch y que resultaron significativos por varios aspectos, a saber:

1. El análisis de los resultados relacionados con las Competencias Adquiridas se muestra en la Figura 1 en donde se visualiza que el 85% de los estudiantes logró desarrollar proyectos funcionales que integraron conceptos avanzados de pensamiento computacional. Estos hallazgos refuerzan la eficacia de Scratch como herramienta de aprendizaje, no solo en contextos educativos básicos, sino también en niveles académicos más avanzados, como lo sugieren investigaciones previas [9]

La implementación de Scratch permitió a los estudiantes abordar problemas complejos, facilitando el desarrollo de habilidades críticas como la resolución de problemas y el pensamiento lógico, competencias esenciales para la educación superior. Esto subraya el potencial de la programación visual para fomentar un aprendizaje significativo en entornos académicos diversos.

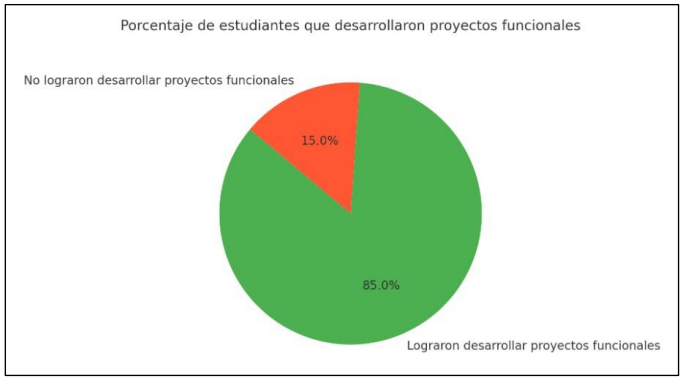

Fig. 1: Estudiantes Que Desarrollan Proyectos Funcionales
Fuente: Los autores (2024)

2. En cuanto a la métrica de Participación Activa, consolidada en la Figura 2, el 92% de los estudiantes completó las actividades propuestas, lo cual sugiere una alta motivación y compromiso con el programa de formación visual con Scratch. Esto puede estar relacionado con el uso de estrategias de gamificación, que incrementaron el interés por las actividades [16]. Además, la inclusión de elementos competitivos y recompensas simbólicas, como puntos y medallas virtuales, contribuyó a mantener el nivel de participación y compromiso [19]

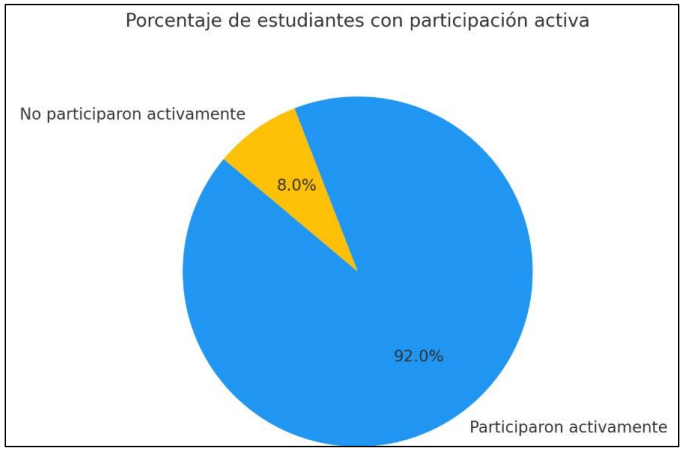

Fig. 2: Participación Activa De Los Estudiantes
Fuente: Los autores (2024)

3. Los proyectos finales incluyeron simulaciones complejas relacionadas con problemas específicos de las

disciplinas de los estudiantes, como la economía y la física en la Figura 3. Esto demuestra la capacidad del curso para conectar los conceptos de programación con aplicaciones del mundo real, favoreciendo la transferencia del conocimiento a contextos prácticos [6]. La interdisciplinariedad de los proyectos permitió a los estudiantes aplicar el pensamiento computacional en el análisis de problemas complejos y en la simulación de escenarios reales, lo cual es un aspecto destacado por algunos autores [2] como una de las principales ventajas del pensamiento computacional.

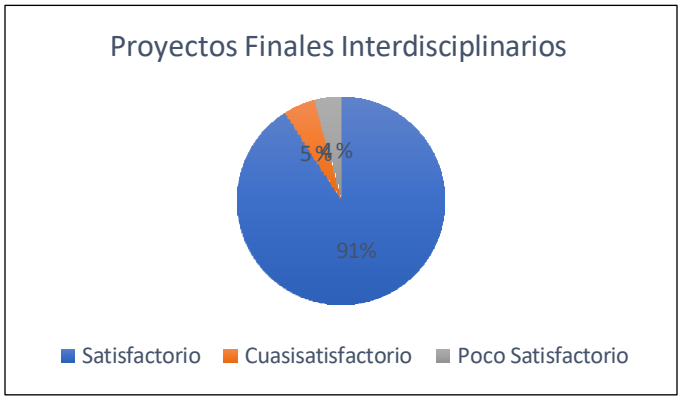

Fig. 3. Interdisciplinariedad En Proyectos Finales
Fuente: Los autores (2024)

4. Más del 90% de los estudiantes valoró positivamente la estructura del curso y los recursos utilizados, destacando la claridad y relevancia de los contenidos, así como la utilidad de las herramientas tecnológicas [11]. La encuesta de satisfacción en la Figura 4, reveló que los estudiantes apreciaron especialmente la naturaleza interactiva y visual del curso, lo cual les permitió aprender de manera más efectiva en comparación con métodos tradicionales. Este resultado es consistente con estudios previos que han demostrado que los enfoques interactivos y visuales mejoran la comprensión y retención del conocimiento [20].

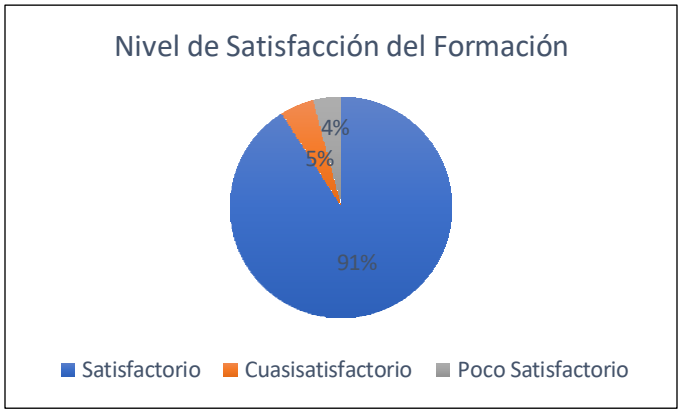

Fig. 4: Satisfacción Del Programa De Formación.
Fuente: Los autores (2024)

IV. Discusión y Análisis

El éxito del programa "programación visual con Scratch como una vía para fortalecer el pensamiento computacional" en la formación de competencias computacionales y creativas muestra el potencial de Scratch para la educación superior, de manera específica en los estudiantes de ingeniería mecánica y eléctrica. A pesar de ser una herramienta desarrollada principalmente para niños, Scratch puede ser adaptado para proporcionar una introducción efectiva a la programación, permitiendo a los estudiantes universitarios abordar problemas complejos de una manera accesible y motivadora [8]. La naturaleza visual y práctica de Scratch permite que conceptos abstractos se traduzcan en experiencias concretas, facilitando un aprendizaje más profundo [21]

El uso del aprendizaje basado en proyectos (ABP) fue clave para mantener el interés de los estudiantes y ayudarlos a ver la relevancia de la programación en sus propias áreas de estudio. Según Thomas [14] el ABP fomenta una mayor retención del conocimiento y habilidades transferibles, lo cual es evidente en la capacidad de los estudiantes para desarrollar proyectos interdisciplinarios. Además, algunos autores [15] sostienen que el ABP promueve un aprendizaje activo y motivador, lo cual fue corroborado por el entusiasmo y compromiso de los estudiantes al trabajar en sus proyectos finales.

La gamificación también tuvo un impacto significativo en la motivación de los estudiantes. La incorporación de elementos de juego en entornos educativos puede transformar la experiencia de aprendizaje [16], haciendo que los estudiantes se sientan más involucrados y comprometidos. En este programa de formación, la gamificación ayudó a superar el reto de mantener la participación activa, particularmente en un entorno virtual donde la desconexión y falta de motivación pueden ser comunes [19].

Otro aspecto importante fue el trabajo colaborativo, que permitió a los estudiantes desarrollar habilidades de comunicación y trabajo en equipo. Algunas posturas argumentan que el aprendizaje colaborativo mejora el rendimiento académico y fomenta el desarrollo de habilidades interpersonales esenciales para el ámbito profesional. En el programa implementado, las actividades colaborativas facilitaron que los estudiantes compartieran conocimientos y se apoyaran mutuamente, lo cual contribuyó a un aprendizaje más significativo y profundo [18].

Sin embargo, durante el desarrollo de la investigación también se identificaron retos. Uno de los principales fue la disponibilidad de tiempo de los estudiantes para participar en las actividades sincrónicas, debido a sus responsabilidades académicas y personales. Este reto podría abordarse mediante un incremento en la cantidad de actividades asincrónicas y el desarrollo de recursos de autoaprendizaje más flexibles [17]. Además, algunas posturas sugieren que el aprendizaje combinado, que integra actividades sincrónicas y asincrónicas, puede ofrecer una solución efectiva para equilibrar las demandas de tiempo de los estudiantes universitarios [22].

En general, la metodología y las herramientas utilizadas facilitaron un aprendizaje significativo, reforzando el valor de integrar metodologías activas y recursos tecnológicos en la educación superior. El aprendizaje multimedia mejora la retención y comprensión cuando se utilizan estrategias que combinan texto, imágenes y actividades interactivas, lo cual fue claramente observado en la retroalimentación positiva de los estudiantes respecto al curso.

V. Conclusiones

1. Scratch demostró ser una herramienta adecuada para la enseñanza de programación y pensamiento computacional en niveles avanzados, proporcionando un enfoque accesible y visual para conceptos complejos.
2. Las estrategias como el aprendizaje basado en proyectos y la gamificación resultaron fundamentales para mantener la motivación y promover un aprendizaje significativo, lo cual coincide con estudios previos sobre la efectividad de estas metodologías.
3. Los estudiantes lograron aplicar los conceptos aprendidos en proyectos interdisciplinarios, lo que sugiere que el curso no solo les proporcionó habilidades técnicas, sino también una mayor comprensión de cómo aplicar estas habilidades en sus propias áreas de estudio.
4. El éxito de este programa de formación sugiere que puede ser replicado en otras instituciones de educación superior, contribuyendo a la formación de competencias esenciales para el futuro profesional de los estudiantes.
5. El programa de formación representa un ejemplo de cómo la programación visual puede ser una herramienta poderosa para el desarrollo de habilidades computacionales y creativas, preparando a los estudiantes para enfrentar los desafíos tecnológicos del mundo moderno.

Por lo anteriormente expuesto, se hace necesario ofrecer algunas recomendaciones que permita abordar la situación identificada en el proceso investigativo y que pueda orientar su solución. Entre las siguiente se mencionan:

1. Realizar estudios que evalúen el impacto de Scratch en otros niveles de educación, como secundaria o educación técnica, y en contextos no académicos, como programas de formación profesional. Esto permitiría explorar cómo la herramienta puede adaptarse para abordar necesidades específicas de diferentes audiencias.
2. Se recomienda diseñar cursos que profundicen en la interdisciplinariedad, integrando desafíos específicos que combinen programación con áreas como las ciencias sociales, las artes o la ingeniería. Estos cursos podrían evaluar cómo la programación visual fomenta el aprendizaje holístico y la resolución de problemas reales en diversos campos.
3. Es importante realizar un seguimiento a largo plazo de los estudiantes que participaron en este curso para analizar cómo las habilidades adquiridas influyen en su desempeño profesional y en su capacidad para adaptarse a entornos laborales tecnológicos. Esto aportaría evidencia sobre el valor práctico de la programación visual en la formación profesional.

Referencias

[1] CSTA (2011). K–12 Computer Science Standards (Level 2) [Documento en línea]. Recuperado de http://csta.acm.org/Curriculum/sub/CurrFiles/CSTA_K12_CSS.pdf

[2] Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35. https://doi.org/10.1145/1118178.1118215.

[3] Royal Society (Great Britain). (2012). Shut down or restart?: The way forward for computing in UK schools. Royal Society.

[4] Kafai, Y. B., & Burke, Q. (2014). Connected code: why children need to learn programming. MIT Press.

[5] Bers, M. U. (2017). Coding as a playground: Programming and computational thinking in the early childhood classroom. Routledge.

[6] Grover, S., & Pea, R. (2013). Computational Thinking in K–12: A Review of the State of the Field. Educational Researcher, 42(1), 38-43. https://doi.org/10.3102/0013189X12463051

[7] ScratchEd Team [Portal Web] (2015). Computational Thinking webinars. Recuperado 2 de Junio de 2015, de http://scratched.gse.harvard.edu/content/1488

[8] Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for all. Communications of the ACM, 52(11), 60-67. https://doi.org/10.1145/1592761.1592779

[9] Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. En Proceedings of the 2012 Annual Meeting of the American Educational Research Association, Vancouver, Canadá.

[10] Molenda, M. (2003). In search of the elusive ADDIE model. Performance Improvement, 42(5), 34-37. https://doi.org/10.1002/pfi.4930420508

[11] Brookhart, S. M. (2013). How to create and use rubrics for formative assessment and grading. ASCD.

[12] Dougiamas, M., & Taylor, P. C. (2003). Moodle: Using learning communities to create an open source course management system. En EdMedia+ Innovate Learning (pp. 171-178). Association for the Advancement of Computing in Education (AACE).

[13] Felder, R. M., & Silverman, L. K. (1988). Learning and Teaching Styles in Engineering Education. Engineering Education, 78(7), 674-681.

[14] Thomas, J. W. (2000). A review of research on project-based learning. Autodesk Foundation.

[15] Blumenfeld, P. C., Soloway, E., Marx, R. W., Krajcik, J. S., Guzdial, M., & Palincsar, A. (1991). Motivating Project-Based Learning: Sustaining the Doing, Supporting the Learning. Educational Psychologist, 26(3-4), 369-398. https://doi.org/10.1080/00461520.1991.9653139

[16] Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From game design elements to gamefulness: Defining "gamification". En Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments (pp. 9-15). https://doi.org/10.1145/2181037.2181040

[17] Johnson, D. W., & Johnson, R. T. (1994). Learning together and alone: Cooperative, competitive, and individualistic learning. Allyn and Bacon.

[18] Slavin, R. E. (1995). Cooperative learning: Theory, research, and practice. Allyn & Bacon.

[19] Zainuddin, Z., Shujahat, M., Haruna, H., & Chu, S. K. W. (2020). The impact of gamification on learning and instruction: A systematic review of empirical evidence. Educational Research Review, 30, 100326. https://doi.org/10.1016/j.edurev.2020.100326

[20] Mayer, R. E. (2009). Multimedia Learning. Cambridge University Press.

[21] Papert, S. (1980). Mindstorms: Children, Computers, and Powerful Ideas. Basic Books.

[22] Garrison, D. R., & Kanuka, H. (2004). Blended learning: Uncovering its transformative potential in higher education. The Internet and Higher Education, 7(2), 95-105. https://doi.org/10.1016/j.iheduc.2004.02.001