

Proyecto para fortalecer el pensamiento computacional mediante una estrategia para mejorar las habilidades de programación en estudiantes de ingenierías en TICs

Abelardo Gómez Andrade, abelardo.gandrade@academicos.udg.mx
María Elena Romero Gastelú, elena.romero@academicos.udg.mx
Janette Araceli Castellanos Barajas, janette.castellanos@academicos.udg.mx
Griselda Pérez Torres, griselda.perez7169@academicos.udg.mx

Universidad de Guadalajara, Centro Universitario de Ciencias Exactas e Ingenierías, Departamento de Ciencias Computacionales, Blvd. Marcelino García Barragán #1421, esq. Calzada Olímpica, C.P. 44430, Guadalajara, Jalisco, México.

Resumen

El siguiente trabajo presenta los resultados obtenidos a partir de un estudio para el fortalecer el pensamiento computacional realizado en los estudiantes de ingenierías orientadas a las Tecnologías de la Información y la Comunicación (TICs) y que requieren como competencia fundamental el dominio de la programación, siendo alumnos del Centro Universitario de Ciencias Exactas e Ingenierías (CUCEI) de la Universidad de Guadalajara (UdeG). Para lo anterior se diseñó una estrategia, consistente en un curso autogestivo que incluye la opción de autoevaluar el aprovechamiento obtenido. Las habilidades que se reforzaron fueron las relacionadas con secuencias, ciclos, eventos, paralelismo, condicionales, operadores y datos. Se empleó como herramienta para diseñar las actividades del curso, el software conocido como Scratch y se implementó en la plataforma Classroom. Se contó con la participación de 247 alumnos, que cursan la materia de Programación o afines, en el periodo de clases comprendido entre agosto de 2021 hasta diciembre del mismo año (2021B). Los resultados de la autoevaluación se analizaron con el software de STATGRAPHICS. En los hallazgos de la presente investigación se confirma la necesidad de fortalecer el pensamiento computacional en los estudiantes mediante el aprendizaje de la programación, ya que la mayoría de los participantes opinaron que el curso sí los apoyó a consolidar los conceptos fundamentales.

Palabras clave

Autogestivo, Estudiantes universitarios, Pensamiento computacional, Programación, Scratch.

Abstract

The following work presents the results obtained from a study to strengthen the computational thinking carried out in engineering students oriented to Information and Communication Technologies (ICTs) and that require as a fundamental competence the programming, being students of the University Center of Exact Sciences and Engineering (CUCEI) of the University of Guadalajara (UdeG). For the above, a strategy was designed, consisting of a self-management course that includes the option of self-assessing the achievement obtained. The skills that were reinforced were those related to sequences, cycles, events, parallelism, conditionals, operators and data. The software known as Scratch was used as a tool to design the course activities and was implemented on the Classroom platform. There was the participation of 247 students, who are taking the subject of Programming or related, in the class period between August 2021 and December of the same year (2021B). The results of the self-assessment were analyzed with the STATGRAPHICS software. The findings of this research confirm the need to strengthen computational thinking in students through learning programming, since most of the participants believed that the course did support them in consolidating the fundamental concepts.

Keywords

Computational thinking, Programming, Scratch, Self-management, University students.

Introducción

“En este mundo de la era digital, en el que vivimos, una de las competencias fundamentales que los estudiantes deben adquirir es la del Pensamiento Computacional. Aunque no existe un consenso general sobre una definición, existe una comprensión general de la misma como un conjunto de habilidades y actitudes necesarias para la resolución, con o sin computadora, de los problemas que puedan surgir en cualquier ámbito de la vida. Medir y evaluar cuál de las competencias adquiridas por los alumnos es fundamental y, para ello, instrumentos de medida previamente validados deben ser usados” (Varela, et. al, 2019). Por lo anterior fue necesario conocer el grado de pensamiento computacional en los alumnos de las carreras que cursan la materia de Programación en el CUCEI de la UdeG. Profesores involucrados en este proceso de enseñanza – aprendizaje y que imparten materias en este centro universitario, formularon las siguientes interrogantes: ¿Cómo es el nivel de pensamiento computacional en los alumnos que cursan la materia de Programación? ¿Cómo se puede medir este valor? ¿Se pueden reforzar las habilidades de programación? ¿Un curso autogestivo diseñado con una herramienta, como Scratch, permite reforzarlas?

El presente trabajo es la conclusión de un proyecto más extenso cuyo objetivo principal es el de fortalecer la competencia del pensamiento computacional de los universitarios que cursan las carreras de Ingeniería en Computación, Ingeniería Informática, Ingeniería Biomédica, Ingeniería en Comunicaciones y Electrónica e Ingeniería Civil. Es un proyecto desarrollado por el cuerpo académico CA-998 titulado Investigación Educativa en Tecnologías de la Información y el CA-539 llamado Análisis e implementación de sistemas, los cuales están adscritos a UdeG y son reconocidos por la Secretaría de Educación Pública de México.

El proyecto consta de dos etapas, una aplicada antes de la impartición del curso de Programación, y otra posterior, al finalizar el periodo de clases 2021B (figura 1).

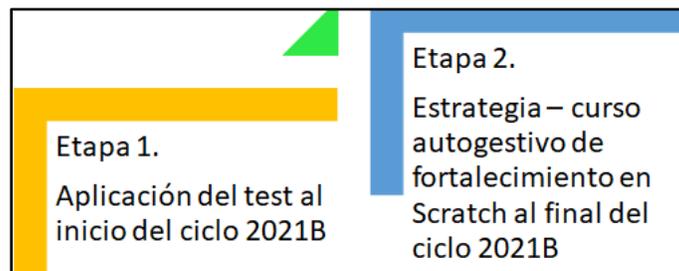


Figura 1. Etapas del proyecto para fortalecer el pensamiento computacional.

La metodología con la que se implementó esta etapa y los hallazgos, se detallan en los siguientes apartados.

Justificación

Como parte de su labor docente un par de integrantes del cuerpo académico CA998 antes referido, imparten cursos de programación, los cuales son fundamentales como parte de la conformación del perfil profesional de los ingenieros en TICs que estudian en CUCEI, reforzando las habilidades del pensamiento computacional.

Existen organismos cuyo objetivo fundamental son las TICs aplicadas a la educación y que reconocen la importancia de contar con habilidades y competencias suficientes de programación y de pensamiento computacional, como ACM (ACM. s.f), ANIEI (ANIEI, s.f) y CONAIC (CONAIC, s.f).

ACM “es una organización científica y educativa mundial dedicada al avance del arte, la ciencia, la ingeniería y la aplicación de la informática, al servicio de los intereses profesionales y públicos mediante el fomento del intercambio abierto de información y la promoción de los más altos estándares profesionales y éticos” (ACM, s.f). ANIEI es “una asociación mexicana, con 40 años de experiencia, que nació en Guadalajara, Jalisco, el 8 de Octubre de 1982; su esencia y su espíritu están dados por el objetivo de

contribuir a la formación de profesionales en TICs sólidamente preparados, y de impulsar la difusión y la asimilación de una cultura computacional en la sociedad” (ANIEI, s.f); entre otros trabajos, destaca el diseño de los modelos curriculares y de competencias, que rigen los procesos de acreditación del Consejo Nacional de Acreditación en Informática y Computación (CONAIC).

Es importante ofrecer a los estudiantes, herramientas distintas que hagan más eficiente el proceso de enseñanza-aprendizaje por lo que se decidió implementar este proyecto para conocer más sobre el comportamiento del curso curricular y ofrecer a los estudiantes otra alternativa.

Objetivos

Objetivo general

Implementar una estrategia para mejorar las habilidades de programación por medio del fortalecimiento del pensamiento computacional en estudiantes de carreras de ingenierías relacionadas con las TIC's.

Objetivos específicos

- Crear un curso autogestivo de programación con actividades diseñadas con bloques de Scratch, con la opción de autoevaluar el aprovechamiento obtenido.
- Interpretar los resultados de la autoevaluación.
- Proponer nuevas estrategias enfocadas a mejorar el proceso de enseñanza-aprendizaje de los estudiantes de nuestro centro universitario.

Marco teórico

“Es sabido que el Pensamiento Computacional carece de un marco de referencia común que determine cuál es su definición y los componentes que lo forman y este hecho provoca un vacío en el desarrollo de metodologías y herramientas de evaluación. En consecuencia, existe la necesidad de llevar a cabo estudios que evidencian el efecto del aprendizaje de la programación informática en el desarrollo del Pensamiento Computacional, ya que estos no abundan entre la literatura especializada.” (Pérez, Arranz, 2017). “El pensamiento computacional ha sido incorporado formalmente en el sistema educativo de varios países, aunque no hay consenso sobre un marco conceptual para incorporar el pensamiento computacional en el currículo” (Pérez, 2021). Los autores de la presente investigación han coincidido con las opiniones anteriores, por lo cual se ha decidido realizar la misma, diseñando un nuevo curso, puesto que en la currícula de las carreras que llevan la materia de programación no incluyen en ella el pensamiento computacional.

“El Pensamiento Computacional ha recibido considerable atención en los últimos años, aunque no existe acuerdo sobre lo que debe incluir una definición de este (Allan et al., 2010; Barr & Stephenson, 2011; National Academies of Science, 2010). Cuny, Snyder, y Wing (2010) definen el Pensamiento Computacional como los “procesos de pensamiento involucrados en formular problemas y encontrar sus soluciones de manera que las soluciones estén representadas de forma tal que puedan llevarse a cabo efectivamente por un agente que procesa información”; descripción ésta que atinadamente (aunque de forma concisa) encuadra el trabajo de los creadores computacionales” (Brennan, Resnick, 2012).

“La frase Pensamiento Computacional nos ayuda a pensar sobre el aprendizaje con Scratch; existe la creencia de que programar con Scratch ofrece un contexto y un conjunto de oportunidades para contribuir en la conversación activa sobre Pensamiento Computacional. Existe el interés sobre la forma en que las actividades de aprendizaje basadas en diseño, particularmente la programación de medios interactivos apoya el desarrollo del Pensamiento Computacional en los jóvenes. Ese interés lo estimula, en parte, la creciente disponibilidad de herramientas que permiten a los jóvenes diseñar sus propios medios interactivos. Pero, más importante aún, ese interés hunde sus raíces en el compromiso de aprender mediante el diseño de actividades, enfoque constructivista del aprendizaje que resalta la importancia de que los jóvenes se comprometan o involucren con el desarrollo de artefactos externos (Kafai & Resnick, 1996).” (Brennan, Resnick, 2012).

Durante los últimos años, estudiando la actividad de la comunidad en línea de Scratch y de los talleres sobre Scratch, se ha desarrollado una definición de Pensamiento Computacional que incluye tres dimensiones clave: conceptos computacionales (los que emplean los diseñadores a medida que programan); prácticas computacionales (las que desarrollan los diseñadores a medida que programan) y perspectivas computacionales (las que los diseñadores construyen sobre el mundo a su alrededor y sobre ellos mismos) (Brennan, Resnick, 2012).

Con Scratch, que es un software gratuito, se puede programar historias interactivas propias, juegos, animaciones y compartir las creaciones con otros en la comunidad online. Scratch ayuda a los jóvenes a aprender a pensar de forma creativa, a razonar sistemáticamente, y a trabajar de forma colaborativa con habilidades esenciales para la vida en el siglo XXI. Scratch está diseñado, desarrollado y moderado por Scratch Foundation, una organización sin fines de lucro cuyo sitio oficial es <https://scratch.mit.edu/> (Scratch, 2021).

Existen otros softwares similares a Scratch, que a través de bloques que representan distintos conceptos de programación, los estudiantes pueden crear personajes animados, videojuegos o hasta videos musicales (Educación 3.0, 2022). Cuentan, además, con plantillas que facilitan la tarea ya que con ellas se puede seleccionar y personalizar un escenario, un sonido o un personaje siendo el propio alumnado el que interactúa con ellos mediante el uso de comandos lógicos. Un ejemplo de estos programas es Bebras, una iniciativa internacional que tiene como objetivo promover la informática (Ciencias de la Computación o Computación) y el pensamiento computacional entre los estudiantes de escuelas de todas las edades. Los participantes suelen estar supervisados por profesores que pueden integrar el desafío Bebras en sus actividades de enseñanza. El desafío se realiza en las escuelas utilizando computadoras o dispositivos móviles (Bebras, 2022).

La programación estructurada es un paradigma de programación orientado a mejorar la claridad, calidad y tiempo de desarrollo de un programa de computadora, utilizando únicamente subrutinas y tres estructuras: secuencia, selección (if y switch) e iteración (bucles, for y while), considerando innecesario y contraproducente el uso de la instrucción de transferencia incondicional (GOTO), que podría conducir a "código espagueti", que es mucho más difícil de seguir y de mantener y era la causa de muchos errores de programación (EcuRed, 2022).

Metodología

Se diseñó una estrategia para cada etapa del proyecto.

Etapa 1. Aplicación de test al inicio del ciclo escolar

La primera etapa se enfocó a conocer la percepción del alumnado de las carreras de Ingenierías en sus habilidades de pensamiento computacional al inicio del ciclo escolar por medio de un cuestionario. Los resultados fueron presentados en el marco del XXXIV Congreso Nacional y XX Congreso Internacional de Informática y Computación ANIEI (Castellanos, et. al, 2021).

El test está compuesto por tres secciones:

- Conceptos básicos de programación.
- Conceptos de Scratch relativos a la programación.
- Bondades de Scratch.

El test se diseñó por medio de la herramienta de recogida de información desde la perspectiva de los alumnos (Cearreta, 2015 en Arranz, 2016). Se implementó con un formulario de Google, participaron 365 estudiantes de las carreras anteriormente citadas, donde 99 fueron alumnos de Informática y 77 de Computación. El 65% de los alumnos no habían llevado cursos de programación previos. En agosto, cuando se aplicó el test, eran 215 estudiantes los que cursarían por primera vez la asignatura de Programación y 19 estudiantes ya tenían 4 años de experiencia programando. De los participantes 92% no habían utilizado Scratch. Sólo 83 alumnos reciben apoyo de un amigo o familiar para aprender a programar.

Para evaluar los conceptos, se le presentaron a los estudiantes soluciones diseñadas con bloques de Scratch.

Etapa 2. Estrategia de fortalecimiento de las habilidades de programación

Consistió en el diseño e implementación por medio de Scratch de un curso autogestivo para el fortalecimiento del pensamiento computacional, con el objetivo de reforzar dicha habilidad en alumnos que cursan Programación o materias afines en el ciclo escolar 2021B. También se invitó a estudiantes que en ese momento no estudiaban las asignaturas referidas anteriormente para poner a prueba el trabajo realizado y contar con una opinión de alumnos con nivel más avanzado de programación, los cuales también requieren contar con la habilidad de tener buenas prácticas de pensamiento computacional enfocado a la programación.

El curso se diseñó con una duración de 20 horas, el cual incluyó los siguientes temas:

1. ¿Qué es el Pensamiento computacional?
2. Conceptos de programación con Scratch:
 - Secuencias
 - Ciclos
 - Eventos
 - Paralelismo
 - Condicionales
 - Operadores
 - Datos

Para el curso se empleó como herramienta el software conocido como Scratch y se implementó en Classroom y participaron estudiantes que cursan las carreras de Ingeniería en Computación, Ingeniería Informática, Ingeniería Biomédica, Ingeniería en Comunicaciones y Electrónica e Ingeniería Civil.

Para cada actividad en el curso, se indicó el objetivo, las instrucciones, y además se anexaron los materiales de apoyo (ver figuras 2, 3 y 4).



Figura 2. Presentación del curso, actividades y materiales.

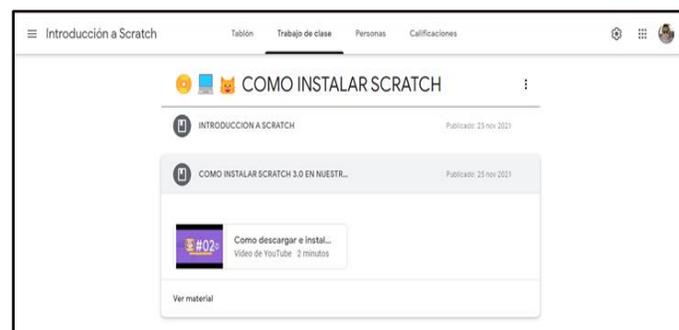


Figura 3. Instalación de Scratch.



Figura 4. Actividad de Datos.

Como última actividad en el curso, los estudiantes realizaron la autoevaluación de su aprendizaje, de acuerdo a la vivencia del curso. Esta autoevaluación se dividió en distintas secciones.

Con los datos de la autoevaluación se realizó una interpretación utilizando el método cuantitativo con análisis descriptivo empleando la herramienta STATGRAPHICS en los datos más relevantes. El paradigma positivista (cuantitativo) busca descubrir el conocimiento a partir de relaciones causa-efecto con las que pretende controlar, explicar y predecir hechos. El investigador busca la neutralidad y hace que prevalezca la objetividad. Este instrumento se centra en aspectos observables que sean posibles de cuantificar y sean libres de valores. (Barberá, 2008).

Los hallazgos se detallan en la siguiente sección del presente documento.

Hallazgos

Con base en los resultados de la sección 1 de la autoevaluación realizada por 247 estudiantes, se destaca la participación del 43.3% de alumnos de la carrera de Ingeniería en Computación, seguida por el 37.2% de Ingeniería Informática. El 26.7% de los participantes corresponden a alumnos de carreras que cursan materias de Programación, el resto concierne a estudiantes de ciclos más avanzados (figura 5).

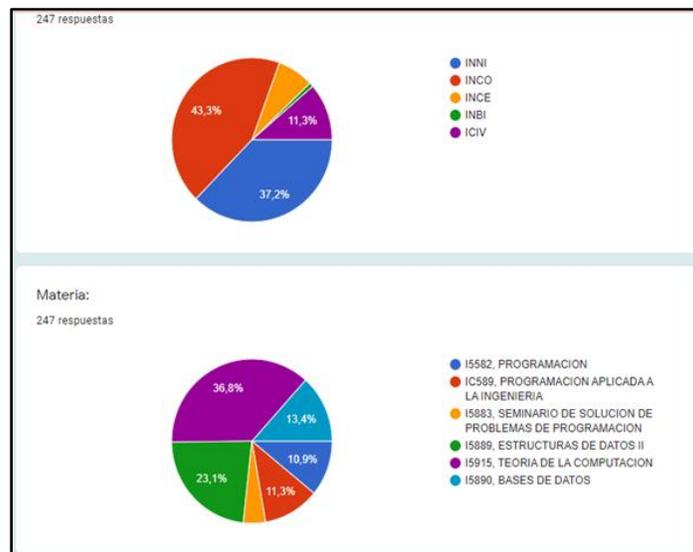


Figura 5. Gráficas de alumnos participantes.

En la sección 2 de la autoevaluación, se utilizó la escala de Likert el 1 representa “Totalmente en desacuerdo” y 5 “Totalmente de acuerdo”, resalta que el mayor porcentaje de respuestas se concentra en el valor 5 de la escala. Por ejemplo, en la pregunta ¿Fortaleciste o reforzaste el entendimiento del concepto

de SECUENCIAS?, sobresale el valor 5 de la escala, lo cual representa que el 51% de los estudiantes está “Totalmente de acuerdo” (ver ilustración 1 y figura 6).

Recuento	247
Mediana	5.0
Moda	5.0
Desviación Estándar	0.958351

Ilustración 1. Resumen estadístico concepto SECUENCIAS.

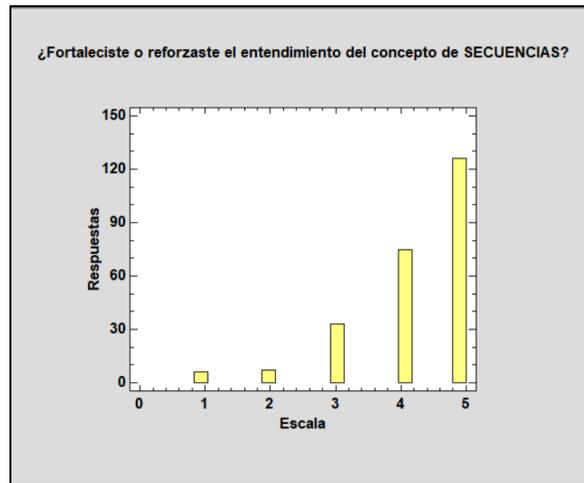


Figura 6. Concepto SECUENCIAS.

De los 247 alumnos, 66 de ellos pertenecen a cursos básicos, como Programación, Seminario de Solución de Problemas de Programación y Programación Aplicada a la Ingeniería; y 181 pertenecen a cursos avanzados, como Bases de Datos, Teoría de la Computación y Estructuras de Datos II.

De estos 181 estudiantes: 91 de ellos afirman estar totalmente de acuerdo en haber fortalecido el concepto de secuencias, esto se corresponde al 50.27% de los participantes. De los 66 alumnos de cursos básicos sólo 35 de ellos están totalmente de acuerdo en haber fortalecido las secuencias, esto se corresponde al 53.03%. La mitad de los alumnos de nivel básico y avanzados consideran que fortalecieron sus habilidades en el tema referido.

A continuación, se detalla la pregunta ¿Fortaleciste o reforzaste el entendimiento del concepto de OPERADORES?, resalta que el porcentaje mayor de respuestas se concentra en el valor 5 de la escala, lo cual representa que el 59.1 % de los estudiantes está “Totalmente de acuerdo” (ver ilustración 2 y figura 7).

Recuento	247
Mediana	5.0
Moda	5.0
Desviación Estándar	1.08563

Ilustración 2. Resumen estadístico concepto OPERADORES.

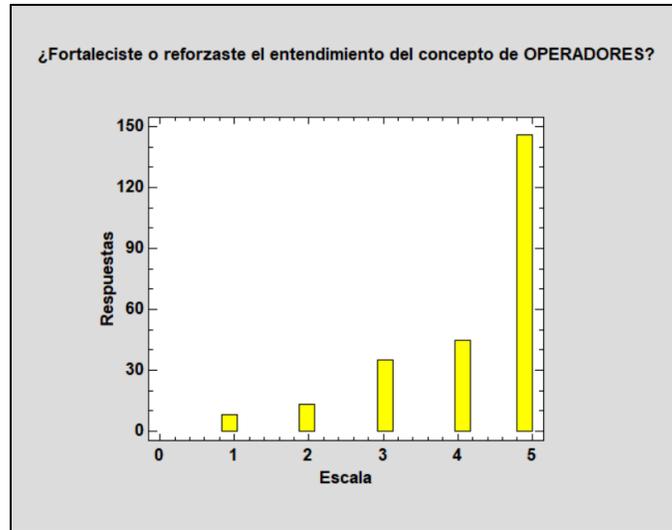


Figura 7. Concepto OPERADORES.

Esto se corrobora con los trabajos sobre la actividad integradora, que es la de Datos. Fueron particularmente sobresalientes los mismos, ya que los alumnos demostraron creatividad y originalidad al realizarla, aplicando los conceptos básicos de programación e integrándolos en una propuesta que superó las instrucciones de la actividad misma. A continuación, se muestran imágenes representativas de esta actividad (figuras 8 y 9).

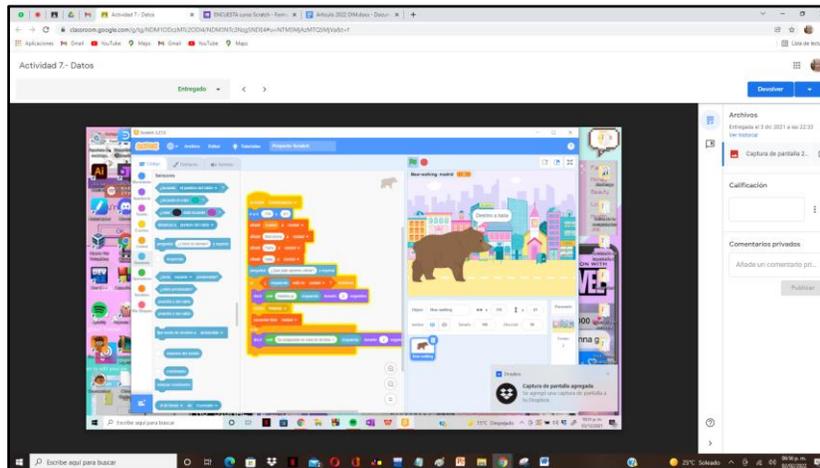


Figura 8. Ejemplo 1 de actividad de "DATOS".

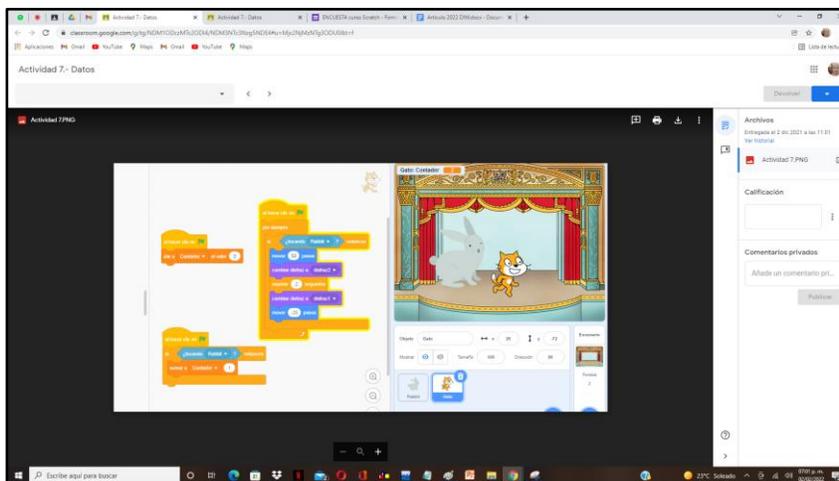


Figura 9. Ejemplo 2 de actividad de “DATOS”.

De las anteriores la que se corresponde a los materiales de la actividad de los ciclos, y que dice: ¿Consideras que los materiales propuestos para CICLOS son adecuados para la resolución de la actividad y la comprensión del concepto?, sobresale de con el valor 5 de la escala, lo cual representa que el 66% de los estudiantes que están “Totalmente de acuerdo” (ver ilustración 3 y figura 10).

Resumen Estadístico para Materiales CICLOS	
Recuento	247
Mediana	5.0
Moda	5.0
Desviación Estándar	0.712544

Ilustración 3. Resumen estadístico de los materiales de la actividad de CICLOS. Fuente diseño propio, creado con STATGRAPHICS.

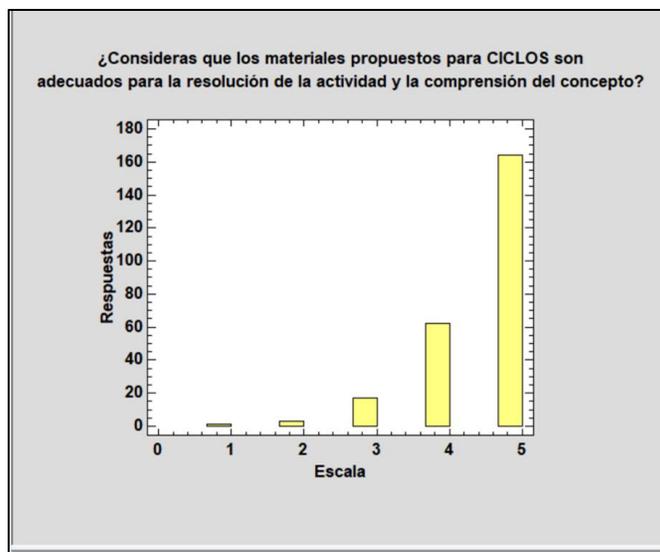


Figura 10. Materiales de la actividad de CICLOS.

Así mismo, en otra de las preguntas del tercer apartado de la encuesta donde dice ¿Consideras que los materiales propuestos para DATOS son adecuados para la resolución de la actividad y la comprensión del concepto?, un 6% de los estudiantes está “Totalmente en desacuerdo”, un 8% “En desacuerdo”, el 13% “Ni de acuerdo ni en desacuerdo”, un 27% “De acuerdo” y el 46% “Totalmente de acuerdo” (ver ilustración 4 y figura 11).

Resumen Estadístico para Materiales DATOS	
Recuento	247
Mediana	4.0
Moda	5.0
Desviación Estándar	1.21123

Ilustración 4. Resumen estadístico de los materiales de la actividad de DATOS. Fuente diseño propio, creado con STATGRAPHICS.

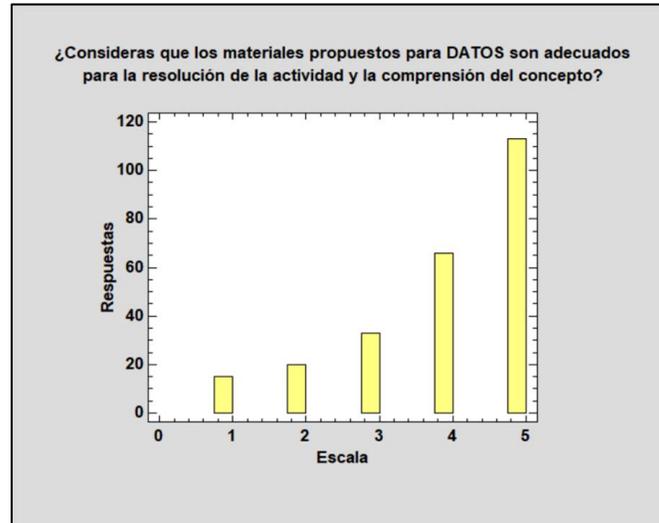


Figura 11. Materiales de la actividad de DATOS.

Conclusiones y trabajos futuros

La experiencia obtenida en el ciclo escolar 2021B, resultado de la conclusión de este proyecto de investigación, ha permitido conocer mejor el asunto del entendimiento de los jóvenes respecto a sus habilidades de programación y de pensamiento computacional al interpretar elementos por bloques de Scratch y realizar los suyos propios, aplicando los conceptos básicos de programación, como son secuencias, ciclos, eventos, paralelismo, condicionales, operadores y datos.

En los resultados de la presente investigación se confirma la necesidad de fortalecer el pensamiento computacional en los estudiantes que cursan programación o alguna otra materia afín, ya que la mayoría de los participantes opinaron que el curso sí los apoyó para comprender mejor los conceptos fundamentales para esta asignatura.

En Pérez (2021) se refiere que no existe un consenso sobre el marco conceptual para incorporar el pensamiento computacional en la currícula y se concluye que un curso adicional de Scratch, como el que se implementó en este proyecto, es idóneo para estudiar algunos aspectos del mismo y acercar a los estudiantes a nuevos enfoques y metodologías que fortalezcan esta habilidad.

En Pérez y Arranz (2017) se menciona que existe la necesidad de llevar a cabo estudios que den evidencia del efecto del aprendizaje de la programación en el desarrollo del pensamiento computacional. Esta investigación aporta información sobre este efecto entre los estudiantes participantes.

De acuerdo a los trabajos de Brennan y Resnick (2012) proponen estudiar el fortalecimiento del pensamiento computacional utilizando el software Scratch y basarse en tres dimensiones: conceptos computacionales, prácticas computacionales y perspectivas computacionales. Con base a lo anterior el presente estudio abarcó la primera dimensión, de las tres referidas anteriormente.

De esta dimensión, la actividad del concepto de datos fue la más representativa de esta investigación, ya que para esta implementación los estudiantes tuvieron que integrar todos los conceptos básicos de la programación, como son secuencias, ciclos, eventos, paralelismo, condicionales y operadores que se

integran en la actividad de datos. Se reflejó el entendimiento de los alumnos en el uso de bloques y por las soluciones que propusieron, al haber superado las expectativas para realizar la misma.

Por lo tanto, los resultados individuales del resto de la autoevaluación, y habiendo analizado cada uno de ellos, se concluye que la estrategia propuesta cumplió el objetivo para el que fue establecida.

Como propuestas de algunas estrategias con base a los hallazgos de esta investigación y con el fin de fortalecer el pensamiento computacional en ciclos escolares futuros entre los estudiantes de nuestro centro universitario, se proponen las siguientes:

- a) Durante el año 2022 impartir un curso Introdutorio a Scratch, pero con una duración de 40 horas. Incluir nuevas actividades con mayor grado de dificultad y con nuevos materiales complementarios. Mantener el curso autogestivo y con autoevaluación. Ampliar el instrumento de evaluación para medir nuevos aspectos del aprovechamiento del curso e interpretar los datos con STATGRAPHICS.
- b) Proponer una alternativa de curso propedéutico a impartirse en la jornada de inducción a alumnos de nuevo ingreso en las licenciaturas que requieran tener habilidades de programación. El mismo sería de carácter voluntario.
- c) Realizar una propuesta dentro de la academia de Programación, para que los profesores que imparten cursos vinculados con la programación implementen algunas actividades curriculares por medio del paradigma del pensamiento computacional estructurado por bloques y empleando la herramienta Scratch.

Estos esfuerzos seguirán estando encaminados a mejorar la eficiencia del proceso de enseñanza - aprendizaje de los estudiantes de ingenierías en tecnologías de la información.

Agradecimientos

Este trabajo fue posible de llevar a buen término gracias a los estudiantes que atendieron a la invitación que recibieron para participar en este trabajo de investigación y que los resultados obtenidos hayan sido confiables, pues aportaron sus opiniones de manera colaborativa y propositiva, con el ánimo de participar en base a su propia experiencia académica de la unidad de aprendizaje de Programación estructurada con lenguaje C.

También fue fundamental el trabajo de académicos adscritos al Departamento de Ciencias Computacionales de la institución ya que apoyaron en la aplicación de la encuesta que se llevó a cabo en el inicio del ciclo escolar 2021B.

Referencias bibliográficas

- ACM. (s.f). Association for Computing Machinery. Consulta: 14 de febrero de 2022. Disponible en: <https://www.acm.org/>
- ANIEI. (s.f). Asociación Nacional de Instituciones de Educación en Tecnologías de la Información. Consulta: 14 de febrero de 2022. Disponible en: <http://www.aniei.org.mx/ANIEI/asociacion/antecedentes/>
- ARRANZ, Héctor. (2016). Herramienta utilizada en el TFM "*Evaluación del Pensamiento Computacional*" propuesta por Cearreta (2015). Consultado el: 06 de abril de 2021. Publicado el: 30 de septiembre de 2016. En: <https://es.slideshare.net/hectorconz/herramienta-recogida-de-informacion-desde-la-perspectiva-de-los-alumnos-cearreta-2015>
- BARBERÁ, E. (2008). Aprender e-learning. Barcelona: Paidós.
- BEBRAS (2022). Consultado en: 23 de febrero de 2022. En: <https://www.bebras.org>
- BRENNAN, K., RESNICK, M. (2012). Nuevas propuestas para evaluar el Pensamiento Computacional. Eduteka. Consultado: el: 23 de febrero de 2022. En: <http://eduteka.icesi.edu.co/articulos/EvaluarPensamientoComputacional>

- CASTELLANOS, J., GÓMEZ, A., ROMERO, M. (2021). *Diseño de instrumento para evaluar el pensamiento computacional en alumnos de ingeniería que cursan la asignatura de Programación*. Transformación Digital: Avances y paradigmas tecnológicos. Editorial: Alfa-Omega Grupo Editor, S.A DE C.V. ISBN: 978-607-538-797-0. Ciudad de México, México. Fecha: diciembre de 2021. Consultado: 18 de febrero de 2022. En: http://www.aniei.org.mx/Archivos/Libros/Libro_transformacion21.pdf
- CONAIC. (s.f.). Consejo Nacional de Acreditación en Informática y Computación. Consultado: 14 de febrero de 2022. Disponible en: <https://www.conaic.net/>
- ECURED. (2022). Programación Estructurada. Consultado: 23 de febrero de 2022. En: https://www.ecured.cu/Programacion_estructurada.
- EDUCACIÓN 3.0 (2022). No sólo Scratch: 15 lenguajes y plataformas para enseñar programación en Primaria y Secundaria. Consultado: el 23 de febrero de 2022. En: <https://www.educaciontrespuntocero.com/recursos/programacion/lenguajes-programacion-informatica-para-primaria-secundaria/>
- PÉREZ, A., ARRANZ, H. (2017). Evaluación del Pensamiento Computacional en Educación. Dialnet. Consultado: 23 de febrero de 2022. En: <https://dialnet.unirioja.es/servlet/articulo?codigo=6484848>
- PÉREZ, J. (2021). Percepción de estudiantes universitarios sobre el pensamiento computacional. Universidad de Los Andes; Venezuela. Consultado: 23 de febrero de 2022. En: <https://polipapers.upv.es/index.php/REDU/article/view/15491>
- SCRATCH. Consultado el: 06 de abril de 2021. En: <https://scratch.mit.edu/>
- VARELA, C., REBOLLAR, C., GARCÍA, O., BRAVO, E., BILBAO, J. (2019). Skills in computational thinking of engineering students of the first school year. Heliyon, Volume 5, Issue 11, ISSN 2405-8440. Consultado el 08 de agosto de 2021. En: <https://doi.org/10.1016/j.heliyon.2019.e02820>. (<https://www.sciencedirect.com/science/article/pii/S2405844019364801>)