

Desarrollo del pensamiento computacional a través del aprendizaje de la programación en estudiantes de ingeniería

Development of computational thinking through learning of programming in engineering students

Augusto Ramiro Brugés Romero ¹, Yadira del Pilar Camperos Villamizar ²

¹ Universidad de Pamplona, Facultad de Ingeniería, Pamplona, Norte de Santander, Colombia.

² Universidad de Pamplona, Facultad de Salud, Pamplona, Norte de Santander, Colombia.

Correspondencia: yadira.camperos@unipamplona.edu.co

Derechos de autor 2022 Revista investigación & praxis en CS Sociales.

Esta obra está bajo una licencia internacional <https://creativecommons.org/licenses/by/4.0/>



Recibido: 02-12-2021
Aceptado: 04-01-2022
Publicado: 01-06-2022

Como Citar: Brugés, A & Camperos, Y. (2022). Desarrollo del pensamiento computacional a través de la programación en estudiantes de Ingeniería. Revista Investigación & Praxis En CS Sociales, 1(1), 1–16.

Resumen: Cuando los estudiantes se enfrentan al estudio de la programación deben acceder de cierta manera a un nivel de deducción y raciocino superior, requiriendo habilidades cognitivas especiales que les permitan realizar un análisis abstracto de lo que puede ser la realidad, sobrepasando los procedimientos cognitivos memorísticos y repetitivos comunes, propios de otras áreas. Por eso la presente investigación de enfoque cualitativo, con diseño fenomenológico, se desarrolló con el fin de saber cómo los estudiantes adquieren habilidades cognitivas asociadas al pensamiento computacional a través del aprendizaje de la programación. Los informantes fueron 10 estudiantes del curso de programación I y II en el periodo 2021-I, y utilizó como técnicas de recolección de información pruebas al iniciar y al finalizar el curso, la observación participante y la entrevista semiestructurada. Los resultados evidenciaron que los estudiantes presentan problemas de Comprensión e interpretación al iniciar los cursos de programación, así como desconocimiento de

conceptos básicos que obstaculizaron el correcto planteamiento de soluciones a los problemas presentados en las pruebas iniciales. El estudio permitió concluir que los estudiantes con poco o ningún conocimiento de programación tienen deficiencias para solucionar problemas, ya que los elementos del pensamiento computacional son aplicados de manera mínima y los aciertos obedecen más a procesos mentales de orden inferior. Los resultados de la prueba final mostraron que la adquisición de conocimientos sobre programación ayudó a mejorar las habilidades de abstracción y análisis, elementos propios del pensamiento computacional; así como el desarrollo de elementos adicionales como el aumento de la creatividad; la inclusión de elementos de diseño, planificación y seguimiento.

Palabras clave: Abstracción, aprendizaje, habilidad cognitiva, lenguaje de programación pensamiento computacional, programación, solución de problemas.

Abstract: When students face the study of programming they must access from certain way to a higher level of deduction and reasoning, requiring special cognitive skills that allow an abstract analysis of what reality can be, surpassing memory and repetitive cognitive procedures, proper of other areas. For this reason, the present research with a qualitative approach, with phenomenological design, was developed in order to know how students acquire cognitive skills associated with computational thinking through the learning of programming. The informants of the research were 10 students of the programming course I and II in the period 2021-I, tests at the beginning and at the end of the course, participant observation and semi-structured interview were used as data collection instruments. The results of the instruments applied showed that students present problems of understanding and interpretation when starting programming courses, as well as lack of knowledge of basic concepts that hindered the correct approach of solutions to the problems presented in the initial tests. The study concluded that students with little or no knowledge of programming have noticeable deficiencies in solving problems, since the elements of computational thinking are minimally and the correct answers obey more to mental processes of inferior order. The results of the final test and the analysis of the other applied instruments shown that the acquisition of programming knowledge helped to improve abstraction and analysis skills, elements of computational thinking; as well as the development of additional elements such as increased creativity; the inclusion of design, planning and monitoring elements.

Keywords: Abstraction, cognitive skill, computational thinking, learning, programming, programming language, problem solving.

1. INTRODUCCIÓN

Actualmente los sistemas de información y, en general, los programas de computador, han sido pieza fundamental en el desarrollo de la sociedad, pues han aportado a la solución de problemas y al manejo de la información en tiempo real, de allí que la programación de sistemas trascienda a los conocimientos relacionados con un área específica y se muestre como una competencia altamente atractiva que le permite al profesional conformar grupos interdisciplinarios para desarrollar herramientas tecnológicas que propendan por el crecimiento de las instituciones y las personas. Una de las competencias que la industria demanda para los futuros ingenieros, independiente de su área de especialidad, es la escritura de programas para computador, a través de los cuales sea capaz de sistematizar sus ideas ya sea para simular sus interpretaciones e hipótesis o para crear soluciones informáticas (Adell Segura, J., Llopis Nebot, M. Á., Esteve Mon, F. M., y Valdeolivas Novella, M. G. 2019).

Es importante saber que aprender a programar no solo es memorizar un conjunto de palabras y símbolos para escribirlos de manera adecuada, la programación requiere de una habilidad especial para la solución de problemas, teniendo en cuenta que cada problema obedece a un planteamiento particular y que cada programador puede abordarlo de manera diferente, dependiendo de su capacidad de análisis y síntesis. Es por ello que se deben buscar alternativas en las estrategias aplicadas para la enseñanza y aprendizaje de la programación, haciendo uso por ejemplo de la gamificación, la programación visual, la robótica educativa, los software diseñados para la enseñanza de la programación, el pensamiento computacional, las aplicaciones móviles, los efectos de la zona de confort, el enfoque constructivista de desarrollo de proyectos, entre otras, de manera que se puedan obtener conclusiones sobre su respectiva eficiencia y efectividad en el aprendizaje de la programación (Santimateo, Núñez, González, 2018).

Insuasty (2016), menciona en su artículo Problemas de Enseñanza y Aprendizaje de los Fundamentos de Programación que: “Ciertas habilidades cognitivas son relevantes al momento del aprendizaje de los fundamentos de programación, tales como la capacidad de abstracción, una buena aptitud lógico-matemática y la facilidad para la resolución de problemas de orden algorítmico” (Burgos, B. M. V., de Cleves, N. R., y Márquez, M. G. C. (2013, p. 235). Con esto se entiende que existen habilidades de pensamiento que, de no haber sido desarrolladas previamente, se constituyen en factores negativos para el aprendizaje. Entre las habilidades de pensamiento se destacan: la capacidad de abstracción, la facilidad de análisis y la destreza para la síntesis, siendo todas ellas habilidades cognitivas fundamentales en la resolución de problemas (Delval, J. 2001).

Durante años de estudios, se han desarrollado múltiples alternativas que propenden por una mejor comprensión de los contenidos de los cursos de programación y se han aplicado muchas de las teorías del aprendizaje con el fin de mejorar su enseñanza y así obtener profesionales bien formados en el área de programación. Desde la psicología, donde inicialmente se dan estos aportes, los trabajos sobre aprendizaje se han desarrollado mayormente sobre los modelos cognitivos y conductistas, sin embargo, dichos modelos han ido en decadencia al encontrar los esquemas constructivistas como alternativa. De allí se han trabajado diferentes influencias que han permitido decantar la forma como los estudiantes pueden adquirir el conocimiento (Fernández, J., Zúñiga, M. E., Rosas, M. V., y Guerrero, R. A. 2018). Una de las teorías utilizadas recientemente en diferentes instituciones para fomentar y mejorar la capacidad de análisis, síntesis y de abstracción propias de la resolución de problemas es el pensamiento computacional (Wing, 2011) y puede ser este enfoque el que permita al estudiante de ingeniería mejorar sus habilidades para apropiarse del conocimiento del curso de programación y en general los métodos para resolver problemas.

2. ANTECEDENTES

Con el fin de conocer investigaciones relacionadas con el tema objeto de estudio, se realizaron diferentes consultas bibliográficas que permitieron revisar el estado del arte en relación con referentes nacionales e internacionales, lo cual se tomó como suministro para el desarrollo de la investigación y permitió contrastar los resultados obtenidos en este trabajo con los ya existentes.

Como primer antecedente se presenta el estudio realizado por Michel Voskoglou y Sherley Buckley, de la escuela de computación de la Universidad de Sudáfrica, presentado en 2012 y resumido en el artículo "Problem Solving and Computers in a Learning Environment" publicado en la revista *Egyptian Computer Science*. En dicha investigación se presentan conceptualizaciones respecto a la solución de problemas, el pensamiento crítico y el pensamiento computacional, la relación entre estos y cómo pueden influenciar en la metodología de solución de problemas. Además, presenta sendos experimentos sobre la aplicación del pensamiento computacional en estudiantes del instituto de educación tecnológica (TEI por sus siglas en inglés) de Patras, Grecia. El artículo menciona que el proceso de solución de problemas es un proceso cognitivo, y que la capacidad de pensamiento es proporcional a la complejidad del problema a solucionar, en el cual se necesita de un pensamiento crítico previo para dar alternativas de solución. Definen en este estudio el pensamiento computacional como un tipo de pensamiento analítico que emplea el pensamiento matemático y de ingeniería para comprender y resolver problemas complejos dentro de las limitaciones del mundo real (Granja, D. O. 2015). Una de las conclusiones más relevantes y que evidencia la importancia que tiene este estudio para la actual investigación tiene relación con el aprendizaje de la programación.

El estudio concluye que, aunque los estudiantes deberían primero desarrollar el pensamiento computacional para luego abordar cursos de programación, la mejor manera de desarrollar el

pensamiento computacional es a través del aprendizaje de la programación (Herrera Clavero, F. 2001).

Otro de los estudios revisados y que aportan argumentación valedera para plantear que la estrategia metodológica utilizada en los cursos de programación es fundamental para la enseñanza y el aprendizaje de los contenidos del curso, se plasma en el artículo “Estrategia metodológica de la enseñanza de la programación para la permanencia de los alumnos de primer año de Ingeniería Electrónica” de los autores Arellano, Fernández, Rosas y Zuñiga de la Universidad de San Luis, en Argentina, artículo publicado en la revista Iberoamericana de Tecnología en Educación y Educación en Tecnología (TEy ET) en junio de 2014. Dicho artículo describe las bondades de las estrategias relacionadas con la metodología de la solución de problemas (incluyendo el pensamiento computacional), como experiencia obtenida en el primer curso de programación que reciben los alumnos de primer año en la carrera de Ingeniería Electrónica de la Facultad de Ciencias Físico-Matemáticas y Naturales, de la Universidad Nacional de San Luis. El estudio menciona que otra metodología sugerida y que tiene su asiento en la era digital es el pensamiento computacional, el cual define el autor como “una habilidad que se encuentra al alcance de todos y que es el resultado de un proceso que permite la solución de problemas caracterizado por organizar los datos de manera lógica para su análisis, representar datos mediante abstracciones, formular soluciones a problemas computacionales y automatizarlas algorítmicamente (Insuasty, J. 2016). El estudio indica que la formación del pensamiento computacional en la resolución de problemas se puede utilizar como estrategia de excelentes resultados con el fin de desarrollar en los alumnos del curso de programación habilidades que estimulen la formación del pensamiento crítico y el desarrollo de la creatividad, resultado que da un indicio fuerte de lo que se puede obtener en la presente investigación y como la estrategia asociada al pensamiento computacional puede contribuir a mejorar los procesos cognitivos de los estudiantes del curso de programación en la Universidad de Pamplona.

El Constructivismo

Definido como una teoría epistemológica, el constructivismo fue desarrollado por Jean Piaget y tomó como influencia pensadores del siglo XVIII como Vico, aunque también se encuentran elementos más recientes tomados de Kant, Marx o Darwin, con el fin de analizar la forma en que adquirimos y transformamos el conocimiento.

En esta teoría, Jean Piaget pretendía mostrar cómo se forma y se desarrolla el conocimiento y como éste cambia a estados de mayor elaboración, además de indicar cuáles son los procesos que tienen lugar en el sujeto para que este adquiera nuevos conocimientos. Para Piaget, todos tenemos algunas capacidades generales innatas, sin embargo, otras se van desarrollando a medida que el sujeto construye su inteligencia y su conocimiento sobre la realidad, lo anterior lo desarrolla actuando sobre su entorno, experimentando con los objetos y las situaciones, transformándolos (Delval, 2001).

Como menciona Granja (2015), “desde el punto de vista constructivista, se puede pensar que el aprendizaje se trata de un proceso de desarrollo de habilidades cognitivas y afectivas, alcanzadas en ciertos niveles de maduración. Este proceso implica la asimilación y acomodación lograda por el sujeto, con respecto a la información que percibe. Se espera que esta información sea lo más significativa posible, para que pueda ser aprendida. Este proceso se realiza en interacción con los demás sujetos participantes, ya sean compañeros y docentes, para alcanzar un cambio que conduzca a una mejor adaptación al medio” (Granja, 2015, p. 99).

Dentro de las teorías cognitivas del aprendizaje elaboradas desde posiciones organicistas, la propuesta por Ausubel es especialmente interesante tras la exposición de la teoría de Vygotsky, ya que está centrada en el aprendizaje producido en un contexto educativo, es decir en el marco de una situación de interiorización o asimilación, a través de la instrucción. Pero además la teoría de Ausubel se ocupa específicamente de los procesos de enseñanza / aprendizaje de los conceptos científicos a partir de los conceptos previamente formados por el individuo en su vida cotidiana. En la terminología de Vygotsky diríamos que Ausubel desarrolla una teoría sobre la interiorización o asimilación, a través de la instrucción, de los conceptos verdaderos, que se construyen a partir de conceptos previamente formados o descubiertos por el individuo en su entorno (Pozo, J. I. 1989). Además, al igual que otras teorías organicistas –o verdaderamente constructivistas – “Ausubel pone el acento de su teoría en la organización del conocimiento de estructuras y en las reestructuraciones que se producen debido a la interacción entre esas estructuras presentes en el sujeto y la nueva información” (Pozo, 1989, p. 201).

Pensamiento Computacional

Los orígenes del pensamiento computacional se remiten al término utilizado por la doctora Jeannette M. Wing, directora del instituto de ciencia de datos de la Universidad de Columbia y consultora informática de la Universidad de Carnegie Mellon, en un artículo publicado en la revista *Communications of the ACM* en marzo de 2006, revista que recopila múltiples recursos en pro de apoyar la investigación y la práctica de la informática. Dicho artículo pretendía dar un acercamiento a las ventajas que tenía que las personas pensarán como informáticos, sin tener necesariamente que ser expertos en el área (Riaño, H. A. V. 2013).

Aunque existen varios intentos por precisar con claridad el término desde diferentes enfoques, ubicamos la definición dada por la doctora Wing (2011), a quien se le reconoce como la precursora del concepto, la cual estableció el pensamiento computacional (PC) como “el proceso del pensamiento que involucramos en la solución de problemas para que las soluciones planteadas puedan ser llevadas a cabo por un agente de procesamiento de información” (Wing, 2011).

Una síntesis del concepto la encontramos en el artículo *El Debate Sobre el Pensamiento Computacional*, en el cual los autores definen el PC como “un conjunto de habilidades y destrezas (herramientas mentales), habituales en los profesionales de las ciencias de la

computación, pero que todos los seres humanos deberían poseer y utilizar para resolver problemas, diseñar sistemas y comprender el comportamiento humano”. Para ellos, el PC debería formar parte de la educación de todo ser humano (Adell, Llopis, Esteve y Valdeolivas, 2019, p. 173).

Componentes y Características del Pensamiento Computacional

Integrantes del comité para el desarrollo del pensamiento computacional (Committee for the Workshops on Computational Thinking) han coincidido en que el PC debería entenderse como una habilidad fundamental comparable con la lectura, la escritura, la expresión oral y la aritmética, que tienen como propósito describir y explicar problemas. Dicho de otra manera, el pensamiento computacional es comparable con las habilidades cognitivas básicas que se espera que tenga una persona promedio en esta era tecnológica (National Research Council, 2010). La mayoría de autores y grupos de trabajo sobre el pensamiento computacional coinciden en incluir tres términos como base fundamental del pensamiento computacional: proceso del pensar, la abstracción y la descomposición (Selby & Woollard, 2013).

Para incluir las habilidades para el desarrollo de problemas soportado en los procesos computacionales dentro del PC, se definen los componentes propios de este como un conjunto de elementos específicos, los cuales se representan en la figura1.

Figura 1

Componentes del pensamiento computacional



Fuente: Elaboración propia.

Habilidades Cognitivas

En palabras de Rigney (1978), citada por Francisco Herrera (2001), las habilidades cognitivas “son entendidas como operaciones y procedimientos que puede usar el estudiante para

adquirir, retener y recuperar diferentes tipos de conocimientos y ejecución... suponen del estudiante capacidades de representación (lectura, imágenes, habla, escritura y dibujo), capacidades de selección (atención e intención) y capacidades de autodirección (autoprogramación y autocontrol)” (Herrera 2001, p. 1).

Las habilidades cognitivas han permitido al hombre desde la elaboración de herramientas primitivas para manipular su entorno, hasta alcanzar las grandes proezas científicas y tecnológicas de la actualidad. Zuñiga, Rosas, Fernández y Guerrero (2014) en su artículo “El Desarrollo del Pensamiento Computacional para la Resolución de Problemas en la Enseñanza Inicial de la Programación” menciona que los estudiantes deben desarrollar una gran variedad de habilidades, más allá de la memorización de sentencias que permitan codificar, pues antes de escribir un código se requiere entender un problema, es decir, abstraer, modelar y analizar, con el fin de plantear soluciones acordes a las necesidades (Zuñiga, Rosas, Fernández y Guerrero, 2014, p. 341). Desde esta perspectiva, el estudiante deberá desarrollar habilidades cognitivas adicionales que contribuyan al mejoramiento de la metodología usada para la resolución de problemas.

3. METODOLOGÍA

Naturaleza de la Investigación

Analizar el desarrollo del pensamiento computacional a través del aprendizaje de la programación y el consecuente desarrollo de habilidades cognitivas de orden superior, necesitó de un enfoque cualitativo y un diseño fenomenológico, puesto que el objetivo principal de dicho enfoque es reconstruir la realidad según la observación que realizan los autores y la interpretación que hacen sobre los comportamientos de los individuos objeto de investigación.

Informantes

Los informantes de la investigación fueron 10 estudiantes y 3 docentes de los cursos de programación estructurada I y II en el periodo 2021-I, de los programas de ingeniería ambiental, civil, eléctrica, telecomunicaciones, mecánica y mecatrónica de la sede principal de la Universidad de Pamplona, seleccionados de manera intencional, teniendo en cuenta los criterios de inclusión: que no estuviesen repitiendo la materia para programación I y que estuviesen cursando los programas de ingeniería ambiental, civil, eléctrica, telecomunicaciones, mecánica y mecatrónica.

Fases de la Investigación

Tal como lo presenta Martínez Miguelez, en su libro Ciencia y Arte en la Metodología Cualitativa, el método fenomenológico consta de tres (3) etapas diferentes: etapa descriptiva,

etapa estructural y etapa de discusión de resultados. La figura 2 representa las diferentes etapas de la investigación.

Figura 2

Fases de la Investigación



Fuente: Elaboración propia.

En la etapa descriptiva se realizó la recolección y revisión bibliográfica, ubicación de los antecedentes investigativos relacionados con el pensamiento computacional, el aprendizaje de la programación y la solución de problemas, que permitieron describir claramente lo que se pretendía estudiar. Posteriormente se ubicó el método a aplicar y las formas de recolección de información, que correspondieron a la aplicación de la prueba diagnóstica, la observación directa, la entrevista semi estructurada y la aplicación de prueba validadora. En la etapa estructural se revisó la información obtenida; se aplicó la prueba inicial; se hizo la observación y se aplicó la entrevista a los informantes estudiantes del curso de programación I y II. Luego se aplicó la prueba final, se hizo la observación y la entrevista final a los estudiantes del curso programación II y la entrevista a los docentes. Después se realizó la categorización de los elementos a evaluar según los resultados obtenidos, se presentaron los resultados según las fases y se obtuvieron resultados generales. En la etapa de resultados y discusión se relacionaron los resultados obtenidos con estudios similares de otros autores, contrastando los logros con el fin de enriquecer el conocimiento en el área. Los resultados permitieron confrontar las posturas de los autores plasmadas en el marco referencial con el fin de corroborar o controvertir las conceptualizaciones dadas por ellos y de esta manera validar la aplicabilidad de los conceptos en el contexto objeto de estudio.

Técnicas de recolección de información: Los instrumentos utilizados para la recolección de la información buscaron establecer el desarrollo del pensamiento computacional y las habilidades asociadas a los niveles cognitivos a medida que el estudiante adquiría conocimientos de programación. Para obtener la información conducente al logro de los objetivos, se seleccionaron como técnicas e instrumentos, los siguientes:

Prueba Diagnóstica y Prueba Validadora: Los instrumentos prueba diagnóstica y prueba validadora constaron de ocho (8) problemas planteados a los cuales el estudiante respondió según su interpretación, haciendo un breve análisis sobre la forma como lo realizó para llegar

a su solución. Este último mecanismo sirvió como elemento de observación por parte del investigador. Las preguntas y problemas allí planteados se tomaron de las iniciativas internacionales “International Bebras Contest” y “Computer Olympiad Talent Search”. Una característica importante de dichas pruebas es que no pretendían evaluar conocimientos previos o habilidades específicas sobre ciencias de la computación, solo se presentaron actividades de resolución de problemas para que dieran solución a una serie de tareas basadas en problemas. La prueba diagnóstica se aplicaron en cinco (5) informantes estudiantes de programación I y cinco (5) informantes estudiantes de programación II al inicio del semestre. El grupo de programación I representó los informantes sin conocimientos de programación y sus resultados se contrastaron con estudiantes que iniciaban el curso de programación II que correspondían a informantes con un conocimiento medio de programación.

La prueba validadora se aplicó a cinco (5) informantes estudiantes de programación II una vez finalizaron el curso. Este grupo representa aquellos informantes que cuentan con un conocimiento superior de programación respecto a los anteriores.

La observación participante: Para la presente investigación, se hizo observación de los estudiantes a través de la descripción que realizaron del procedimiento usado al momento de enfrentar las preguntas de las pruebas diagnóstica y validadora que se aplicaron al iniciar el curso de programación I y programación II y al finalizar el curso de programación II respectivamente. Con este instrumento se obtuvo información en un contacto directo con la realidad, que permitió observar el comportamiento de los individuos objeto de investigación durante el desarrollo de las actividades al momento de plantear soluciones a los problemas a medida que van adquiriendo conocimientos de programación. El análisis que realizó cada estudiante es descrito por él junto con la respuesta de cada ítem en cada prueba, elementos que posteriormente fueron analizados por parte del investigador.

Entrevista Semiestructurada: En la presente investigación la entrevista buscó obtener información sobre las percepciones de los estudiantes sobre el uso, existencia y desarrollo de habilidades cognitivas y metacognitivas al iniciar y finalizar los cursos de programación y de la manera como estas influyen al momento de solucionar los problemas planteados. Su aplicación se realizó en el momento en que cada estudiante completó la prueba diagnóstica y la prueba validadora y estuvo compuesta de un conjunto de preguntas abiertas que tenían relación con los elementos que utilizaron para desarrollar cada pregunta o con la forma en que dichos elementos aportan al desarrollo de los problemas.

4. RESULTADOS

Para determinar si el estudiante utilizó o no cierta habilidad de manera adecuada, cada uno de los problemas planteados se relacionó con uno o más elementos del pensamiento computacional, relación que se encuentra previamente establecida en cada uno de los problemas presentados a las iniciativas “International Bebras Contest” y “Computer Olympiad

Talent Search” y que se adaptaron a las pruebas aplicadas en la presente investigación. Cuando el estudiante dio respuesta a algún problema de la prueba aplicada, el investigador pudo realizar la clasificación de las habilidades utilizadas y al final determinar si desarrolla el pensamiento computacional. De esta manera, la prueba permitió evaluar los siguientes elementos: descomposición (DE), abstracción (AB), reconocimiento de patrones (PR), algoritmos (AL), evaluación (EV), razonamiento Lógico (RL), todos elementos asociados al pensamiento computacional.

Al aplicar la relación de las preguntas acertadas con respecto a la habilidad que se utilizó, se obtuvieron los resultados que se presentan en la tabla 1, la cual permite determinar el uso y efectividad de la habilidad computacional al desarrollar los problemas planteados.

Tabla 1: Resultados por Habilidad – Diagnóstico

Habilidad	Programación I			Programación II		
	Si	No	No Contestó	Si	No	No Contestó
Descomposición (DE)	7	6	2	13	2	0
Abstracción (AB)	6	9	0	10	5	0
Reconocimiento Patrones (PR)	4	3	3	6	3	1
Algoritmos (AL)	9	6	5	15	4	1
Evaluación (EV)	6	7	2	11	4	0
Razonamiento Lógico	8	2	0	8	2	0

Fuente: Elaboración propia.

Los resultados obtenidos muestran la debilidad de los estudiantes que inician el curso de programación en la mayoría de las habilidades computacionales, evidenciando falencias notorias en más de la mitad de ellos en elementos como la abstracción, el reconocimiento de patrones, la algoritmia y la evaluación de los resultados. Estos elementos fueron mejorados en algunos de los estudiantes que contaban con un conocimiento previo de programación, revirtiendo las fallas al solucionar los problemas y obteniendo mejores resultados.

El resultado de la observación realizada sobre la prueba diagnóstica y la forma en que el estudiante sin conocimientos de programación afronta el problema, denotan una alta deficiencia en elementos cognitivos superiores en la mayoría de los estudiantes, sin embargo, estos elementos mejoran de manera gradual en estudiantes que han visto al menos un curso de programación. Por su parte, las habilidades de orden inferior muestran buena presencia y se van fortaleciendo a medida que el estudiante adquiere conocimientos mayores de programación y desarrolla el pensamiento computacional. Como resultado de las entrevistas aplicadas al finalizar las pruebas diagnósticas los estudiantes de los dos grupos seleccionados se inclinan por mencionar habilidades de orden inferior para la solución de los problemas. Elementos como la observación, comparación, la interpretación, la memoria y el uso de

conceptos anteriores aparecen identificados por parte de los informantes. Al igual que lo refleja la prueba diagnóstica, la entrevista deja al descubierto problemas generales de interpretación y comprensión en los estudiantes que inician o han visto un curso de programación, así como deficiencias o desconocimiento de herramientas y técnicas que les permitan desarrollar de una manera más adecuada los problemas planteados. Al indagar sobre qué problemas se les facilitaron, las respuestas dadas permitieron determinar que los problemas con contenidos de algoritmos o descomposición son fáciles de resolver por los estudiantes, mientras que los que requieren habilidades cognitivas más complejas como evaluación, abstracción y síntesis se les dificultan.

Muchas de las dificultades expresadas por los estudiantes con ningún o poco conocimiento de programación se centran en la falta de comprensión y en no encontrar alternativas o métodos para solucionar los problemas. Esperan, de forma general, que el aprendizaje de la programación les aporte estos elementos, de manera que se reflejen en una mayor efectividad al momento de desarrollar los problemas, generalizando soluciones, reduciendo tiempos y evitando errores.

Los resultados del siguiente instrumento, correspondiente a la prueba validadora, se muestran en la tabla 2.

Tabla 2: Resultados por Habilidad Prueba Validadora

Habilidad	Si	No	No Contestó
Descomposición (DE)	14	1	0
Abstracción (AB)	22	3	0
Reconocimiento Patrones (PR)	13	2	0
Algoritmos (AL)	24	6	0
Evaluación (EV)	5	0	0
Razonamiento Lógico	2	3	0

Fuente: Elaboración propia.

Los resultados de la prueba validadora muestran una mejora significativa en el desarrollo de los problemas, lo que indica un mejor desempeño en cuanto a las habilidades asociadas con el pensamiento computacional de los estudiantes que tienen mayor conocimiento sobre programación. Se observa que logran aplicar elementos asociados con habilidades de orden superior, como descomposición y evaluación y elementos de generalización asociados con la

síntesis, que se miden a través de la capacidad de identificación de patrones y de pensamiento algorítmico.

Se observa que las soluciones dadas por los estudiantes se relacionan con una alta capacidad para seguir secuencias, es decir, son capaces de ejecutar un algoritmo, sin embargo, otras habilidades que revisten mayor complejidad no son capaces de ejecutarlas de una manera adecuada o simplemente no se utilizan.

Como elemento particular se debe mencionar que el razonamiento lógico sigue mostrando un buen desempeño, es una habilidad que permanece a lo largo del tiempo en los estudiantes y que no se afecta a causa del desarrollo de habilidades asociadas al pensamiento computacional (Riaño, H. A. V. 2013).

Un análisis entre los resultados obtenidos en la prueba diagnóstica aplicada a estudiantes con ningún o bajo conocimiento de programación y los resultados de la prueba validadora, mostró una alta tendencia de mejora en cuanto a la solución de los problemas para los estudiantes que han obtenido conocimientos de programación en todas las habilidades cognitivas asociadas con el pensamiento computacional, recalcando avances significativos en procesos de descomposición, abstracción, generalización y evaluación.

En esta prueba se pudo evidenciar un alto contenido de elementos asociados al pensamiento computacional, elementos que eran escasos en la primera prueba. Los resultados presentados y las observaciones realizadas también permitieron identificar que muy pocos estudiantes del curso de programación II tuvieron dificultades al desarrollar pruebas con elementos asociados a la descomposición, el pensamiento algorítmico sigue siendo una fortaleza y además se incluyen elementos de evaluación.

La percepción del estudiante respecto a las habilidades que el aprendizaje de la programación y el desarrollo del pensamiento computacional está orientada a la mejora de elementos de análisis y comprensión, así como el desarrollo del razonamiento y la memoria. Adicionalmente el estudiante expresa tener mejoras en la creatividad y la concentración. Los estudiantes expresan la importancia del aprendizaje de la programación y el desarrollo del pensamiento computacional en los elementos que permiten tener alternativas en la solución de problemas.

5. CONCLUSIONES

Aprender a programar requiere de algunas habilidades cognitivas básicas previas que le permitan al estudiante adquirir los elementos centrales de un lenguaje en particular, como su semántica y su sintáctica. Sin embargo, el aprendizaje de la programación en su etapa temprana no requiere destrezas específicas ni mucho menos complejas. Como efecto colateral, el aprendizaje de la programación permite desarrollar el pensamiento computacional, como lo expresaron Voskoglou y Buckley (2012) en su artículo "Problem

Solving and Computers in a Learning Environment”, situación que facilita la adquisición de elementos adicionales como la algoritmia, la descomposición y la generalización, entre otras. Estos elementos ayudan al estudiante a desarrollar nuevas habilidades asociadas con niveles cognitivos superiores que, según Bloom, se relacionan con el análisis, la síntesis y la evaluación. Como aspecto retroalimentador, el desarrollo de estas habilidades cognitivas mejoran el proceso de aprendizaje, no solo de la programación, sino de las demás materias. De esta manera, y en consonancia con lo expresado por algunos autores referenciados en este trabajo investigativo, se puede decir que las habilidades cognitivas inferiores sirven para resolver problemas elementales y son el soporte para aprender a programar con los elementos básicos de un lenguaje de programación; sin embargo, dicho aprendizaje conlleva a desarrollar habilidades cognitivas de orden superior, algunas asociadas al desarrollo del pensamiento computacional, que mejoran la forma en que se da solución a los problemas y contribuye a la obtención de un aprendizaje significativo.

El trabajo investigativo permitió observar que los estudiantes identificaron la comprensión como un elemento fundamental para afrontar problemas y como una habilidad casi innata que todo individuo tiene, habilidad que puede llegar a perfeccionarse con el aprendizaje de la programación y el desarrollo del pensamiento computacional. Otro elemento que rescataron los estudiantes como una habilidad necesaria y preexistente al aprendizaje de la programación, y que de hecho utilizaron para afrontar este tipo de problemas, es el análisis, tomado este como una simple interpretación de lo solicitado, es decir, asociada a la comprensión de lo requerido.

Para finalizar, se debe tener en cuenta que el pensamiento es una experiencia interna e intrasubjetiva, a través de la cual se puede abstraer, discriminar, inferir, inventar, encontrar respuestas, resolver problemas, analizar, reflexionar, lo cual se manifiesta mediante la elaboración de hipótesis, razonamientos y emisión de juicios. Es por ello que se encuentran procesos simples como la observación y la comparación, o procesos más complejos como el análisis, la síntesis y la abstracción y que son aplicados de formas diferentes por cada individuo (Burgos, de Cleves, Márquez, 2013). Para las técnicas de solución de problemas soportadas en lenguajes de programación, el individuo debe ser capaz de desarrollar y perfeccionar aquellas habilidades de nivel superior que le permitan tener las herramientas necesarias para transformar su entorno. Dentro de estas habilidades está el análisis, la síntesis, la argumentación y la abstracción (Burgos, de Cleves, Márquez, 2013).

Bajo esta perspectiva, el estudiante espera que el aprendizaje de la programación y el desarrollo del pensamiento computacional le ayude a potenciar dichos elementos y así lograr tener una mayor efectividad y asertividad al momento de desarrollar problemas, logrando generalizar los procedimientos para reducir los tiempos y evitar posibles errores en los que puedan incurrir. El desarrollo del pensamiento computacional a través del aprendizaje de la programación les permitió a los estudiantes desarrollar destrezas en la forma en que dan solución a los problemas. Si bien dichas destrezas no se enmarcan propiamente dentro de las estructuras cognitivas de los niveles que establece Bloom, son elementos afines que permiten

complementar los procesos de resolución de problemas. Los estudiantes expresaron que al adquirir conocimientos de programación pudieron mejorar la forma como presentan las soluciones, identificando la creatividad, el pensamiento algorítmico, la planificación, el diseño, el seguimiento y la rapidez con que desarrollan ahora los problemas, como elementos que se adquirieron o mejoraron.

Se debe mencionar entonces que el aprendizaje de la programación facilita el desarrollo del pensamiento computacional y este a su vez el desarrollo de habilidades metacognitivas, tal como lo menciona Burgos, de Cleves, y Márquez (2013), y que permite al estudiante adquirir elementos y desarrollar otras destrezas para la solución de problemas de cualquier tipo. Sin embargo, se debe precisar que el estudio no determina la efectividad que tendría el desarrollo del pensamiento computacional en contextos diferentes al educativo y sería un buen punto de partida para realizar nuevas investigaciones al respecto.

REFERENCIAS

Adell Segura, J., Llopis Nebot, M. Á., Esteve Mon, F. M., y Valdeolivas Novella, M. G. (2019). El debate sobre el pensamiento computacional en educación. *Revista Iberoamericana de Educación a Distancia*, 22(1), 171-186.

Burgos, B. M. V., de Cleves, N. R., y Márquez, M. G. C. (2013). Habilidades de pensamiento como estrategia de aprendizaje para los estudiantes universitarios. *Revista de investigaciones UNAD*, 12(2), 23-41.

Delval, J. (2001). Hoy todos son constructivistas. *Educere, La Revista Venezolana de Educación*, 5(15), 353-359.

Fernández, J., Zúñiga, M. E., Rosas, M. V., y Guerrero, R. A. (2018). Experiences in learning problem-solving through computational thinking. *Journal of Computer Science & Technology*, 18(2), 136-142.

Granja, D. O. (2015). El constructivismo como teoría y método de enseñanza. *Sophia*, (19), 93-110.

Herrera Clavero, F. (2001). Habilidades cognitivas. *Notas del departamento de Psicología Evolutiva y de la educación*. (Universidad de Granada. España).

Insuasty, J. (2016). Problemas de enseñanza y aprendizaje de los fundamentos de programación. *Revista educación y desarrollo social*, 10(2), 234-246.

Martínez, M. (2004). *Ciencia y arte en la metodología cualitativa (1aed.)*. Editorial Trillas.

Pozo, J. I. (1989). Teorías cognitivas del aprendizaje (1ª ed.). Ediciones Morata.

Pozo, J. I., Pérez, M. D., Domínguez, J., Gómez, M. A., y Postigo, Y. (1994). La solución de problemas. Vol. 57. Madrid: Santillana.

Riaño, H. A. V. (2013). Características que evidencian el impacto educativo y cultural, a causa del fenómeno del desplazamiento forzado en Cúcuta. *Educación y Humanismo*, 15(25), 73-93.

Santimateo, D., Nuñez, G., y González, E. (2018). Estudio de dificultades en la enseñanza y aprendizaje en los cursos básicos de programación de computadoras en Panamá. *Revista de Investigación en Tecnologías de la Información*, 6(11), 13-18.

Wing, J. (2011). Research notebook: Computational thinking—What and why. *The Link Magazine*, 20-23.

Zúñiga, M. E., Rosas, M. V., Fernández, J., y Guerrero, R. A. (2014). El desarrollo del pensamiento computacional para la resolución de problemas en la enseñanza inicial de la programación. In *XVI Workshop de Investigadores en Ciencias de la Computación*, 340-343.