



Percepción de estudiantes universitarios sobre el pensamiento computacional

Perception of university students about computational thinking

Pérez, J.  **Pérez, J.**

perezj89@gmail.com

perezj89@gmail.com

Universidad de Los Andes
(Venezuela)

Universidad de Los Andes
(Venezuela)

Resumen

El pensamiento computacional ha sido incorporado formalmente en el sistema educativo de varios países, sin embargo, no hay consenso sobre un marco conceptual para incorporar el pensamiento computacional en el currículo. Como etapa previa a la incorporación formal del pensamiento computacional en una asignatura de introducción a la programación a nivel universitario, el objetivo de esta investigación es develar la percepción de los estudiantes sobre el pensamiento computacional para obtener un marco de referencia. Esta

Abstract

Computational thinking has been formally incorporated into the educational system of several countries, however, there is no consensus about a theoretical framework for incorporating computational thinking in the curriculum. As an initial stage to the formal incorporation of computational thinking in an introductory programming course at the university level, the objective of this research is to reveal the students' perception about computational thinking to get a reference. This research was carried out with a qualitative approach of descriptive scope through interviews that

investigación se llevó a cabo con un enfoque cualitativo de alcance descriptivo mediante entrevistas que estuvieron basadas en la escalera metacognitiva y fueron aplicadas a algunos informantes clave que aprobaron una asignatura de introducción a la programación con diferentes rendimientos: bajo, regular, y alto. El análisis de las entrevistas permitió develar la percepción de los estudiantes como un marco conceptual que incluye una definición, tres características, un procedimiento, y dos contextos principales de aplicación. La definición está orientada a la resolución de problemas y la ejecución de actividades. Las características están relacionadas con cuatro conceptos, dos formas de adquisición, y beneficios para la resolución de problemas. El procedimiento tiene seis pasos: comprensión de la situación, identificación de la dificultad, descomposición en las partes constituyentes, reconocimiento de patrones, selección de información relevante, y diseño y ejecución de un algoritmo. Los contextos principales de aplicación son la universidad y la vida cotidiana. Finalmente, la comparación con otros marcos conceptuales permitió concluir que los estudiantes tienen una percepción acertada del pensamiento computacional, pero en los términos básicos.

Palabras clave: pensamiento, enseñanza superior, programación, análisis cualitativo, entrevista, enseñanza de la informática.

were based on the metacognitive ladder and applied to some key informants who approved an introductory programming course with different performances: low, medium, and high. The analysis of the interviews allowed us to reveal the students' perception as a theoretical framework that includes a definition, three characteristics, a procedure, and two main application contexts. The definition is related to the resolution of problems and the execution of activities. The features are associated with four concepts, two ways of acquisition, and benefits to the resolution of problems. The procedure has six steps: understanding the situation, identifying the difficulty, decomposition into the constituent parts, pattern recognition, selecting relevant information, and designing and executing an algorithm. The main contexts of application are university and everyday life. Finally, a comparison with other theoretical frameworks allowed us to conclude that students have an accurate perception of computational thinking, but in the basic terms.

Key words: thinking, higher education, programming, qualitative analysis, interviews, computer science education.

Introducción

Aunque el pensamiento computacional surgió de la necesidad por programar los computadores, ha sido adoptado para la resolución de problemas en otras disciplinas. Para Wing (2006), pionera del pensamiento computacional, representa una actitud y

un conjunto de habilidades universalmente aplicables que todos, no solo científicos informáticos, estarían ansiosos por aprender y usar. De acuerdo con Wing (2008), algunas evidencias del impacto positivo del pensamiento computacional en otras disciplinas pueden encontrarse en la estadística, la biología, y la economía. En la estadística, el aprendizaje automático ha permitido identificar patrones en grandes conjuntos de datos; en la biología, los algoritmos de secuenciación han acelerado la capacidad de secuenciar el genoma humano; en la economía, ha surgido el campo de la microeconomía computacional, con aplicaciones como la personalización de anuncios, subastas en línea o servicios de reputación.

El impacto positivo del pensamiento computacional ha despertado el interés por incorporarlo en todos los niveles educativos. De acuerdo con Compañ et al. (2015), si se entiende como una habilidad para resolver problemas usando conceptos informáticos, es natural asumir que este pensamiento se desarrolla en una asignatura de introducción a la programación. Desde esa perspectiva, en los últimos años se ha venido incorporando la programación como asignatura en la enseñanza obligatoria de varios países, tales como Alemania (Delcker e Ifenthaler, 2017), España (Valverde et al., 2015), Noruega (Bocconi et al., 2018), Reino Unido (Csizmadia et al., 2015), entre otros.

El pensamiento computacional también puede ser desarrollado de manera desconectada, es decir, a través de un conjunto de actividades que favorecen el pensamiento computacional sin la utilización de computadores. Entre las propuestas desconectadas se pueden encontrar: creación de historias (Kordaki y Kakavas, 2017), juegos de mesa (Bayeck, 2018), actividades de una gymkana (Jagušt et al., 2018), juguetes Montessori (Zapata-Ros, 2019), preguntas metacognitivas durante la ejecución de actividades de la vida cotidiana (Pérez, 2019a), entre otras.

La incorporación formal del pensamiento computacional en el currículo requiere un marco conceptual como soporte. Aunque existen varios marcos conceptuales sobre el pensamiento computacional (Adell et al., 2019), no hay consenso sobre alguno en particular. De hecho, según Polanco et al. (2020), ni siquiera hay una definición consensuada del pensamiento computacional. Por ejemplo, Brennan y Resnick (2012) asocian el pensamiento computacional con siete conceptos: secuencias, ciclos, eventos, paralelismo, condicionales, operadores, y datos. Por otra parte, Zapata-Ros (2015) asocia catorce conceptos: análisis descendente, análisis ascendente, heurística, pensamiento divergente, creatividad, resolución de problemas, pensamiento abstracto, recursividad, iteración, métodos por aproximaciones sucesivas, métodos colaborativos, patrones, sinéctica y metacognición. Por su parte, Sondakh et al. (2020) asocian once conceptos: abstracción, pensamiento algorítmico, automatización, descomposición, depuración, evaluación, generalización, resolución de problemas, trabajo en equipo, comunicación, e inteligencia espiritual.

Independientemente de la estrategia (asignaturas introductorias de programación o propuestas desconectadas) y del marco conceptual, la incorporación del pensamiento computacional es un hecho en varios países. Dado que no hay consenso sobre el marco conceptual, y que el interés del autor de esta investigación es incorporar formalmente el pensamiento computacional en una asignatura introductoria de programación a nivel universitario, el objetivo de esta investigación es descubrir la percepción de los estudiantes sobre el pensamiento computacional en aras de obtener un marco de referencia.

Método

La investigación se llevó a cabo con un enfoque cualitativo de alcance descriptivo. En concordancia con Hernández et al. (2010), el enfoque es cualitativo porque se utilizó la recolección de datos sin medición numérica, y el alcance es descriptivo porque se especificaron características y rasgos importantes del fenómeno de estudio. De acuerdo con Flick (2015), la investigación cualitativa parte de la noción de la construcción social de las realidades sometidas a estudio, interesándose en las perspectivas de los participantes, las prácticas cotidianas, y el conocimiento cotidiano que hace referencia a la cuestión estudiada. Por esa razón, para conocer la percepción de los estudiantes sobre el pensamiento computacional se aplicaron entrevistas a tres participantes o informantes clave, quienes según Galeano (2004) se definen como interlocutores competentes social y culturalmente que conocen y participan en la realidad objeto de estudio.

Participantes

Dado el interés de esta investigación por conocer la percepción de los estudiantes actuales para favorecer a los estudiantes futuros, los sujetos de estudio fueron estudiantes que ya aprobaron una asignatura de introducción a la programación. En primera instancia, se asume que los estudiantes aprobados han desarrollado el pensamiento computacional porque son capaces de resolver los problemas clásicos de programación para asignaturas introductorias. La metodología utilizada en la asignatura se puede encontrar en (Pérez y Azuaje, 2019; Pérez y Castro, 2018; Pérez y Pedroza, 2018).

De acuerdo con Galeano (2004), los informantes clave se seleccionan con criterios de representatividad cualitativa, tales como: conocimiento, experiencia, motivación para participar en el estudio, y aceptación de las condiciones de desarrollo de la investigación. Además, según Hernández et al. (2010), se debe garantizar una muestra diversa en aras de localizar diferencias y coincidencias. En ese orden de ideas, se seleccionaron como informantes clave a tres estudiantes con diferente calificación final en la asignatura (ver Tabla 1). El primer estudiante (I1) obtuvo bajo rendimiento con una calificación de 10 puntos (mínima calificación aprobatoria); el segundo estudiante (I2) obtuvo rendimiento regular con una calificación de 15 puntos; y el tercer estudiante (I3) obtuvo alto rendimiento con una calificación de 20 puntos (máxima calificación aprobatoria).

En aras de caracterizar a los informantes clave, se aplicaron dos instrumentos para conocer sus características como estudiantes nativos digitales y sus habilidades de pensamiento computacional. El primer instrumento, que consistió en prestar atención a un video mientras se responden algunas preguntas, tal como se aplica en (Pérez, 2019b), permitió conocer que los tres estudiantes acusan necesidad de acceso diario a Internet, creencia de competencia tecnológica, y preferencia por la información gráfica. Además, permitió conocer que sólo dos informantes prefieren premios por cualquier logro, y que sólo un informante ha desarrollado la capacidad multitarea (ver Tabla 1).

El segundo instrumento consistió en resolver problemas extraídos del *International Bebras Contest*, tal como lo aplican en (Mooney y Lockwood, 2020), y permitió conocer el pensamiento computacional desarrollado por los estudiantes según cinco habilidades: descomposición, generalización, abstracción, algoritmos, y evaluación. Los resultados mostraron que todos dominan la descomposición, la generalización, y los algoritmos, pero sólo dos dominan la evaluación, y sólo uno domina la abstracción (ver Tabla 1).

Tabla 1. Características de los informantes clave.

Instrumento	Característica	I1	I2	I3
Rendimiento Académico	Calificación final	10	15	20
Características de nativos digitales	Necesidad de acceso diario a Internet	Sí	Sí	Sí
	Creencia de competencia tecnológica	Sí	Sí	Sí
	Preferencia por la información gráfica	Sí	Sí	Sí
	Preferencia por premios	Sí	Sí	No
	Capacidad multitarea	No	No	Sí
Habilidades del pensamiento computacional	Descomposición	Sí	Sí	Sí
	Generalización	Sí	Sí	Sí
	Abstracción	No	No	Sí
	Algoritmo	Sí	Sí	Sí
	Evaluación	No	Sí	Sí

Nota: Elaboración propia.

Entrevista

El propósito de la entrevista fue descubrir la percepción de los estudiantes sobre el pensamiento computacional desde sus propias palabras. En ese sentido, se aplicó una entrevista estructurada de cuatro preguntas que están basadas en la escalera metacognitiva (Perkins y Swartz, 1991). Las preguntas fueron: ¿Qué es para usted el pensamiento computacional?, ¿Cómo cree usted que adquirió el pensamiento computacional?, ¿Cómo aplica usted el pensamiento computacional?, y ¿Cuáles conceptos de informática aplica usted en otros ámbitos?; las entrevistas se llevaron a cabo en un salón de clases donde sólo se encontraban el entrevistador y un informante a la vez. El procedimiento aplicado fue el siguiente: primero, el entrevistador le explicó al informante clave el propósito de la entrevista; segundo, para cada pregunta el entrevistador leyó en voz alta y el informante respondió en voz alta; y tercero, el entrevistador agradeció al informante por su participación. Las tres entrevistas fueron grabadas en audio para su posterior transcripción.

La validez y la confiabilidad de las entrevistas están basadas en los criterios de credibilidad y dependencia, respectivamente, en concordancia con las recomendaciones presentadas por Hernández et al. (2010). El criterio de credibilidad se refiere a que el investigador ha captado el significado de las experiencias de los participantes. Para ello, en esta investigación se tomaron en cuenta cinco recomendaciones: primero, se procuró en todo momento realizar una interpretación objetiva, evitando que las creencias y opiniones del investigador influyeran en las interpretaciones; segundo, se consideraron todos los datos con la misma importancia; tercero, se percibieron a todos los informantes clave como iguales; cuarto, durante las entrevistas se procuró no influir a los informantes clave; y quinto, se intentó codificar las categorías emergentes sin pensar en sus aportes para la construcción de la teoría, en aras de buscar evidencia tanto favorecedora como desfavorecedora. Por otro lado, para el criterio de dependencia, se presentaron con claridad los criterios de selección de los informantes clave, las preguntas de las entrevistas, y el papel que desempeñó el investigador como entrevistador.

El análisis de las entrevistas se llevó a cabo mediante la categorización, la cual es un proceso analítico por medio del cual se fragmentan, conceptualizan e integran los datos para formar una teoría (Strauss y Corbin, 2002). La categorización permitió

reunir grupos de conceptos o subcategorías y establecer las respectivas relaciones entre conceptos sobre el mismo fenómeno. En esta investigación se aplicó la categorización en primer y segundo plano. En la categorización de primer plano se codificaron las unidades en categorías iniciales, y en segundo plano, se compararon las categorías iniciales entre sí, buscando posibles vinculaciones para agruparlas en categorías generales. Luego, se procedió a interpretar los resultados.

Resultados

Categorización en primer plano

El Informante 1 (I1), el cual es un estudiante con bajo rendimiento, ha proporcionado dieciséis categorías emergentes iniciales, las cuales son: extrapolación, optimización, resolución de problemas, programación, investigación, herramientas, razonamiento lógico, descomposición, dificultad, algoritmo, repetición, abstracción, satisfacción, vida cotidiana, planificación y asignatura. En la Tabla 2 se presentan los extractos en las propias palabras del Informante 1.

Tabla 2. Categorías emergentes del Informante 1.

Categoría	Extracto de la entrevista
Extrapolación	"...es asociar los, este, conceptos de programación que puedan utilizarse para la vida cotidiana..."
Optimización	"Creo que es una manera de resolver más fácil los problemas o cualquier otra cosa que vayas a hacer..."
Resolución de problemas	"yo pueda resolver por ejemplo la descomposición de un problema"
Programación	"A través de los cursos de programación "
Investigación	"...leyendo un poco sobre el tema para trabajos de investigación "
Herramientas	"...herramientas para evaluar el pensamiento computacional como Scratch..."
Razonamiento lógico	"Más que todo la mayoría utiliza Scratch y problemas de razonamiento lógico."
Descomposición	"...el problema grande partiéndolo en pequeños problemas donde se puedan trabajar mejor..."
Dificultad	" ver cuáles problemas son más fáciles "
Algoritmo	"...resuelvo primero y después me enfoco con los que tienen mayor dificultad..."
Repetición	" Conceptos de informática, los ciclos de repetición, por ejemplo, aunque suene muy sencillo el comer todos días "
Abstracción	"...cuando voy a hacer algo, esta una tarea que tenga que hacer solamente me enfoco en lo necesario para resolverlo..."
Satisfacción	"...se siente bien porque por ejemplo yo puedo trazar varios objetivos, y dependiendo de esos objetivos puedo ir cumpliendo algunas de esas etapas..."
Vida cotidiana	"...los ciclos de repetición, por ejemplo, aunque suene muy sencillo el comer todos días, cepillarse todos los días, bañarse todos los días "
Planificación	" para hacer lo que voy a hacer durante el día lo descompongo en varios objetivos, por ejemplo, estudiar de tal hora a tal hora, e, administrar el tiempo para hacer cada cosa, y eso, descomponiendo el día en varias partes para hacer varias cosas..."
Asignatura	" en los exámenes ver cuáles problemas son más fáciles, los resuelvo primero y después me enfoco con los que tienen mayor dificultad..."

Nota: Elaboración propia.

El Informante 2 (I2), el cual es un estudiante con rendimiento regular, ha proporcionado quince categorías, las cuales son: transdisciplinariedad, resolución de problemas, programación, razonamiento lógico, comprensión, descomposición, patrones, dificultad, abstracción, secuencias, complejidad, optimización, vida cotidiana, asignatura y trámite. En la Tabla 3 se presentan los extractos en las propias palabras del Informante 2.

Tabla 3. Categorías emergentes del Informante 2.

Categoría	Extracto de la entrevista
Transdisciplinariedad	"...enfrentar un problema ya sea de la vida cotidiana o del ámbito informático."
Resolución de problemas	"son un conjunto de conceptos, métodos y técnicas para afrontar un problema ..."
Programación	"A través del desarrollo de programas..."
Razonamiento lógico	"al desarrollar algún tipo de ejercicio que implique alguna lógica..."
Comprensión	"debemos tratar de entender cuál es la principal problemática que vemos en dicho problema..."
Descomposición	"...luego entender las partes que lo componen..."
Patrones	"la búsqueda de patrones porque cuando tendemos a atacar un problema en muchos casos no es la primera vez que nos enfrentamos a él, entonces tratamos de encontrar similitudes para poder resolverlos."
Dificultad	"...y ver si puede ser solucionado de una manera factible."
Abstracción	"la abstracción, debido a que debemos tratar de entender las principales características de los problemas que enfrentamos día a día..."
Secuencias	"...la secuencialidad porque debemos encontrar un orden al realizar actividades..."
Complejidad	"...y el grado de complejidad de los mismos, baja un poco porque no es un, no busco dificultar más las soluciones sino facilitarlas."
Optimización	"...encontrando mejores maneras de solucionar entendiendo mejor la situación en la que me encuentro, de manera que me permite comprender y analizar patrones que permiten mejorar la resolución de dichos problemas."
Vida cotidiana	"...en un problema con un vecino trato de ver cuál es lo que está afectando a la relación entre las personas y ver si puede ser solucionado de una manera factible."
Asignatura	"En la universidad, por lo menos cuando voy a resolver algún problema ya sea de una materia "
Trámite	"o de un proceso administrativo pues trato de entender lo más importante para resolverlo..."

Nota: Elaboración propia.

El Informante 3 (I3), el cual es un estudiante de alto rendimiento, ha proporcionado con sus respuestas catorce categorías, las cuales son: extrapolación, transdisciplinariedad, resolución de problemas, patrones, práctica, secuencias, repeticiones, algoritmos, decisiones, comodidad, inconsciente, optimización, vida cotidiana y asignatura. En la Tabla 4 se presentan los extractos en las propias palabras del Informante 3.

Tabla 4. Categorías emergentes del Informante 3.

Categoría	Extracto de la entrevista
Extrapolación	“ algo relacionado a la computación o a la informática se pueda implementar también en otras cosas ”
Transdisciplinariedad	“ se puede ya de manera automática aplicarlo en otros ámbitos, en otras, si en otras cosas...”
Resolución de problemas	“ se puede tener una resolución de problemas vista desde el lado matemático o de otros ámbitos ”
Patrones	“ a un problema encontrarle solución pero ya a través de algo aprendido, algo que ya él aprendió.”
Práctica	“...y lo adquiriré pues en la práctica de verdad, solamente practicando creo que se obtiene ese pensamiento.”
Secuencias	“...por ejemplo, conseguir un, mantener un orden...”
Repeticiones	“por ejemplo, el repetir algo que es uno, uno lo hace en la vida cotidiana”
Algoritmos	“Aplico la resolución de algoritmos para tomar decisiones, e, seguir estructuras, algo que sea estructurado por ejemplo, hacer algo primero, luego otra cosa, luego otra cosa, y así, pero siguiendo un orden específico ”
Decisiones	“...aplico, e, eso básicamente, si porque lo otro entra también en decisiones.”
Comodidad	“Me siento cómoda al utilizar el pensamiento computacional...”
Inconsciente	“...actualmente ya se pudiera decir que lo hago hasta inconscientemente, para la resolución de problemas...”
Optimización	“ me va ayudando a la resolución de problemas y me sirve mejor.”
Vida cotidiana	“...entonces se aplica prácticamente en todo desde que uno se despierta, lo aplico para comer, lo aplico para vestirme, lo aplico para ir a la facultad, lo aplico para, e, para cosas que uno hace.”
Asignatura	“ lo aplico para la resolución de problemas que no están dentro de la materia programación, sino en otras materias de la carrera ”

Nota: Elaboración propia.

De las categorías emergentes de los tres informantes clave, surgieron veintiséis categorías distintas. Hay cuatro categorías que son comunes para todos, las cuales son: optimización, resolución de problemas, vida cotidiana y asignatura. Las cinco categorías que únicamente tienen en común los Informantes 1 y 2, son: programación, razonamiento lógico, descomposición, dificultad y abstracción. Las tres categorías que tienen en común los Informantes 1 y 3, son: extrapolación, algoritmo y repetición. Las tres categorías comunes entre los Informantes 2 y 3, son: transdisciplinariedad, patrones y secuencias. Por otra parte, con respecto a las categorías únicas emergentes, del Informante 1 se tienen cuatro: investigación, herramientas, satisfacción y planificación. Del Informante 2 se tienen tres: trámite, comprensión y complejidad. Finalmente, del Informante 3 se tienen cuatro categorías únicas: decisiones, práctica, comodidad e inconsciente. En la Tabla 5 se presenta una relación de las categorías mencionadas por los informantes clave con su respectiva descripción.

Tabla 5. Categorización en primer plano.

Categoría	Descripción	I1	I2	I3
Extrapolación	Los conceptos de computación pueden aplicarse en otros ámbitos	X		X
Optimización	El pensamiento computacional facilita la resolución de problemas	X	X	X
Resolución de problemas	El pensamiento computacional se aplica para la resolución de problemas	X	X	X
Programación	El pensamiento computacional se adquiere en los cursos de programación	X	X	
Investigación	El pensamiento computacional se adquiere leyendo	X		
Herramientas	El pensamiento computacional se adquiere con herramientas como Scratch	X		
Razonamiento lógico	El pensamiento computacional se adquiere resolviendo ejercicios de razonamiento lógico	X	X	
Descomposición	Los problemas se descomponen	X	X	
Dificultad	Se resuelven primero los problemas más fáciles	X	X	
Algoritmo	Se usan algoritmos para la resolución de problemas	X		X
Repetición	Se usa el concepto de repetición para la resolución de problemas	X		X
Abstracción	Se usa el concepto de abstracción para la resolución de problemas	X	X	
Satisfacción	Alcanzar objetivos parciales genera satisfacción	X		
Vida cotidiana	Los conceptos de programación se utilizan en la vida cotidiana	X	X	X
Planificación	El pensamiento computacional se utiliza para planificar	X		
Asignatura	Los conceptos de programación se utilizan en otras asignaturas	X	X	X
Trámite	El pensamiento computacional se utiliza para trámites administrativos		X	
Transdisciplinariedad	El pensamiento computacional permite resolver problemas de cualquier ámbito		X	X
Comprensión	Primero hay que comprender el problema		X	
Patrones	Hay que determinar el orden de las actividades		X	X
Secuencias	Hay que identificar similitudes entre los problemas		X	X
Complejidad	El pensamiento computacional permite abordar la complejidad		X	
Decisiones	Se usa el concepto de decisiones para la resolución de problemas			X
Práctica	El pensamiento computacional se adquiere mediante la práctica			X
Comodidad	El uso del pensamiento computacional es cómodo			X
Inconsciente	El pensamiento computacional se aplica de manera inconsciente			X

Nota: Elaboración propia.

Categorización en segundo plano

Las categorías iniciales se agruparon en cinco categorías generales: significado, beneficios, adquisición, aplicación y contexto. La categoría de *significado* está relacionada a las percepciones de los informantes clave sobre la definición del pensamiento computacional y agrupa tres subcategorías: resolución de problemas, extrapolación y transdisciplinariedad. La categoría *beneficios* está asociada a las ventajas que ofrece el pensamiento computacional, y agrupa cinco subcategorías: optimización, satisfacción, complejidad, comodidad e inconsciente. La categoría *adquisición* está relacionada a las distintas formas de desarrollar el pensamiento computacional que conciben los informantes clave y reúne cinco subcategorías: programación, investigación, herramientas, razonamiento lógico, y práctica. La categoría *aplicación* está asociada a los conceptos que se ponen en práctica cuando los informantes clave usan el pensamiento computacional, y agrupa nueve subcategorías: descomposición, dificultad, algoritmo, repetición, abstracción, comprensión, patrones, secuencias y decisiones. Finalmente, la categoría *contexto* está asociada al ámbito de aplicación, y aquí se agrupan cuatro subcategorías: vida cotidiana, planificación, asignatura y trámites. En la Tabla 6 se presenta la agrupación de las categorías.

Tabla 6. Categorización en segundo plano.

Categoría	Descripción	Subcategorías
Significado	Definición de pensamiento computacional	Extrapolación Transdisciplinariedad Resolución de problemas
Beneficios	Ventajas inherentes a la utilización del pensamiento computacional	Satisfacción Optimización Complejidad Comodidad Inconsciente
Adquisición	Distintas formas de desarrollar el pensamiento computacional	Programación Investigación Herramientas Razonamiento lógico Práctica
Aplicación	Procedimientos asociados al pensamiento computacional	Descomposición Dificultad Algoritmo Repetición Abstracción Comprensión Patrones Secuencias Decisiones
Contexto	Ámbitos de aplicación	Vida cotidiana Planificación Asignatura Trámite

Nota: Elaboración propia.

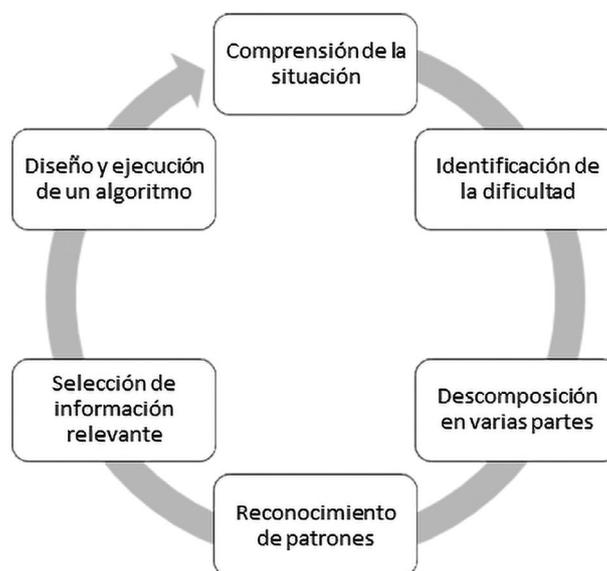
Interpretación

Los resultados obtenidos permitieron develar una definición, tres características (conceptos asociados, formas de adquisición, y beneficios), un procedimiento y dos contextos de aplicación.

- Definición de pensamiento computacional: los informantes clave asociaron el pensamiento computacional con el uso de conceptos que se aprenden en el ámbito de la informática para mejorar la resolución de problemas o la realización de actividades tanto en contextos universitarios como en la vida cotidiana. Para los informantes, el pensamiento computacional implica el desarrollo de una actitud transdisciplinaria y una capacidad de extrapolación para la resolución de problemas en diferentes contextos y la implementación de los conceptos de informática en otras actividades, respectivamente.
- Conceptos asociados al pensamiento computacional: para los informantes clave, los conceptos de la informática que se usan en el pensamiento computacional son la descomposición, el reconocimiento de patrones, la abstracción, y los algoritmos. El concepto de descomposición se refiere al fraccionamiento de un problema o actividad en sus partes constituyentes, en aras de facilitar su comprensión, planificación de estrategia de desarrollo y ejecución. El reconocimiento de patrones consiste en la identificación de similitudes entre diferentes problemas o actividades, e incluso entre las partes que componen a alguno de ellos, con la intención de reutilizar soluciones o pasos previamente conocidos en aras de acelerar la resolución del problema o la ejecución de la actividad. El concepto de abstracción hace referencia a la identificación de la información relevante para descartar aquella que es innecesaria, con el propósito de facilitar la comprensión del problema o actividad. El concepto de algoritmo está asociado con el diseño de una estrategia de resolución o la ejecución de una actividad, especificada paso a paso, mediante la combinación de tres conceptos muy utilizados en la programación de computadoras: las secuencias, que hacen referencia a la ejecución de instrucciones de manera ordenada; las decisiones, que condicionan la ejecución de las instrucciones; y las repeticiones, que permiten especificar, en una o varias instrucciones, más de una ejecución.
- Adquisición del pensamiento computacional: según los informantes clave, éste puede ser adquirido mediante la programación de computadores o con la realización de actividades distintas. La adquisición con la programación de computadores puede ser mediante la programación textual o a través de la programación visual por bloques. Por otro lado, las otras actividades que permiten adquirir el pensamiento computacional son la lectura de información obtenida por cuenta propia o la resolución de problemas de razonamiento lógico. Independientemente del tipo de adquisición, tal como un informante clave resaltó, es muy importante la práctica de los conceptos asociados al pensamiento computacional de manera similar a la práctica de las operaciones aritméticas básicas de las matemáticas.
- Beneficios del pensamiento computacional: la aplicación del pensamiento computacional, según los informantes clave, principalmente proporciona beneficios para la resolución de problemas. El pensamiento computacional

permite optimizar el proceso de resolución de problemas de programación porque al hacer énfasis en los conceptos asociados al pensamiento computacional, los pasos de resolución se perciben con mayor facilidad, lo cual implica que los problemas de programación también pueden solucionarse mediante la aplicación de razonamientos básicos de lógica. De manera análoga sucede con la resolución de problemas en otros ámbitos o la ejecución de actividades cotidianas, ya que permite reducir su complejidad. La utilización del pensamiento computacional también tiende a ser cómoda; por un lado, esto se debe a que la resolución de problemas con este pensamiento disminuye la carga cognitiva de las personas, y, por otro lado, la alta frecuencia de resolución de problemas tanto a nivel universitario como de la vida cotidiana promueve el uso inconsciente o automático, es decir, la aplicación del pensamiento computacional sin pensar en ello. Además, estimula la sensación de satisfacción cuando se resuelve correctamente un problema.

- Procedimiento de aplicación del pensamiento computacional: la categoría *aplicación* ha permitido configurar un procedimiento de aplicación del pensamiento computacional, organizando lógicamente los aspectos procedimentales mencionados por los informantes clave. El procedimiento tiene seis pasos: comprensión de la situación, identificación de la dificultad, descomposición en las partes constituyentes, reconocimiento de patrones, selección de información relevante, y diseño y ejecución de un algoritmo. En la Figura 1 se presentan estos pasos de manera secuencial para indicar el procedimiento general, sin embargo, es importante destacar que, dependiendo de la complejidad del problema o de la actividad, se pueden aplicar todos los pasos o combinaciones entre ellos.



Nota: Elaboración propia.

Figura 1. Procedimiento para aplicar el pensamiento computacional.

- Contextos de aplicación: de acuerdo con los informantes clave, los ámbitos de aplicación del pensamiento computacional para los estudiantes universitarios son principalmente dos: la universidad y la vida cotidiana. En la universidad, aplican este pensamiento en varios contextos: planificación del tiempo, trámites

administrativos y resolución de problemas en las asignaturas. En la planificación del tiempo, que tiene como propósito administrar el estudio, la asistencia a clases y las actividades extras, utilizan la descomposición, y el diseño y ejecución de un algoritmo. En la realización de trámites administrativos, que por lo general son extensos debido a la burocracia universitaria, aplican la comprensión de la situación, la selección de la información relevante, y el diseño y ejecución de un algoritmo. En las asignaturas, utilizan este pensamiento para la resolución de problemas tanto de la asignatura de programación como de otras asignaturas, teniendo como referencia todos los pasos del procedimiento. Finalmente, en la vida cotidiana usan este pensamiento en varios contextos, tales como: planificar una visita al banco, comer, cepillarse los dientes, resolver problemas con los vecinos, vestirse, ir a la facultad, entre otros.

Discusión

El significado que los informantes clave asociaron al pensamiento computacional es consistente con las definiciones de otros autores. Wing (2006) menciona que el pensamiento computacional involucra resolución de problemas, diseño de sistemas, y comprensión del comportamiento humano, basado en conceptos fundamentales de las ciencias de la computación. En ese sentido, los estudiantes señalaron la resolución de problemas utilizando conceptos de la computación, pero no mencionaron el diseño de sistemas o la comprensión del comportamiento humano. No obstante, la percepción de los informantes es totalmente consistente con autores que enfatizan el pensamiento computacional para la resolución de problemas. Por ejemplo, Aho (2012) se refiere al pensamiento computacional como un proceso de pensamiento involucrado en la formulación de problemas, de tal manera que sus soluciones puedan ser representadas como pasos computacionales y algoritmos. Adicionalmente, los informantes mencionaron que este pensamiento mejora la realización de actividades en contextos universitarios o en la vida cotidiana, lo cual es consistente con Basogain et al. (2015), quienes se refieren al pensamiento computacional como una metodología basada en la implementación de los conceptos básicos de las ciencias de la computación para resolver problemas cotidianos, diseñar sistemas domésticos y realizar tareas rutinarias. Otras definiciones similares se pueden encontrar en la literatura. Por ejemplo, Polanco et al. (2020) encontraron treinta definiciones del pensamiento computacional.

Los informantes asociaron cuatro conceptos de la computación con el pensamiento computacional, sin embargo, otros autores presentan listas de conceptos que son más amplias. Por ejemplo, Brennan y Resnick (2012) presentan siete conceptos: secuencias, ciclos, eventos, paralelismo, condicionales, operadores, y datos. En las entrevistas, los informantes sólo mencionaron las secuencias y los condicionales. Por otro lado, Zapata-Ros (2015) presenta una lista de conceptos compuesta por catorce elementos: análisis descendente, análisis ascendente, heurística, pensamiento divergente, creatividad, resolución de problemas, pensamiento abstracto, recursividad, iteración, métodos por aproximaciones sucesivas, métodos colaborativos, patrones, sinéctica y metacognición. Los informantes coincidieron con dos elementos de esta lista: abstracción, y patrones. Aunque los informantes mencionaron la resolución de problemas, no la percibieron como un concepto asociado sino como un propósito. Por su parte, Sondakh, et al. (2020) presentan otra lista de conceptos: abstracción, pensamiento algorítmico, automatización, descomposición, depuración, evaluación, generalización, resolución de problemas,

trabajo en equipo, comunicación, e inteligencia espiritual. En este caso, los informantes coincidieron con cuatro elementos de esta lista: abstracción, patrones, descomposición, y algoritmos. Adell et al. (2019) encontraron otros marcos conceptuales que también tienen elementos comunes con las respuestas de los informantes clave.

Los informantes indicaron que el pensamiento computacional puede ser adquirido mediante la programación (textual o visual) de computadores. Este aspecto es consistente con Compañ et al. (2015) quién asume que este pensamiento se desarrolla naturalmente en una asignatura de introducción a la programación. Los estudiantes también indicaron que los conceptos se pueden adquirir con la resolución de problemas de razonamiento lógico o la lectura de información obtenida por cuenta propia. Para Zapata-Ros (2015), la codificación es parte del razonamiento lógico, lo cual es consistente con la idea de los estudiantes. Sin embargo, parece raro que los estudiantes señalaran la posibilidad de adquirir el pensamiento computacional a través de la lectura porque según Espinosa (2017) los estudiantes nativos digitales no tienen hábitos de lectura para la adquisición de conocimientos.

Los informantes señalaron que el pensamiento computacional permite percibir los pasos de resolución de problemas con mayor facilidad y comodidad. Al respecto, Park et al. (2015) indican que mediante la técnica de electroencefalograma se puede apreciar cómo la carga cognitiva de los estudiantes que emplean un pensamiento computacional es menor que la carga cognitiva de los estudiantes que no emplean este pensamiento. En este mismo orden de ideas, Zapata-Ros (2015) indica que permite a los estudiantes organizar su mundo de relaciones, en un contexto de comunicación más racional y eficiente.

Los informantes mencionaron los siguientes aspectos procedimentales: comprensión de la situación, identificación de la dificultad, descomposición en las partes constituyentes, reconocimiento de patrones, selección de información relevante, y diseño y ejecución de un algoritmo. En este sentido, Zapata-Ros (2015) indica que el pensamiento computacional tiene asociado siete elementos que son utilizados en la informática a nivel procedimental: detección y delimitación del problema y de su naturaleza; delimitación de métodos y disciplinas en la resolución del problema; organización de la resolución, retroalimentación e investigación formativa; diseño de la resolución; algoritmo o diagrama de flujo; elaboración del código o programa; y prueba o validación. No obstante, para otros autores como Valverde et al. (2015) existen sólo cuatro (4) fases a nivel procedimental, las cuales son: descomposición, reconocimiento de patrones, generalización de patrones y abstracción, y diseño algorítmico. Otros autores como Csizmadia et al. (2015) trabajan con cinco fases: descomposición, generalización, abstracción, pensamiento algorítmico y evaluación.

Los ámbitos de aplicación del pensamiento computacional para los estudiantes universitarios son principalmente la universidad y la vida cotidiana. Al respecto, Zapata-Ros (2015) indica que el pensamiento computacional permite a los individuos organizar su entorno y sus estrategias de desenvolvimiento. El pensamiento computacional es útil, no sólo directamente en las ciencias de la computación, sino que también aporta beneficios al resto de asignaturas. En general, los resultados son consistentes con Wing (2006) cuando menciona que este pensamiento representa un conjunto de habilidades y actitudes, aplicables universalmente.

Conclusiones

El análisis de las entrevistas permitió extraer veintiséis categorías iniciales que fueron agrupadas posteriormente en cinco categorías generales. La interpretación de esas categorías permitió conocer que los estudiantes tienen una percepción acertada sobre el pensamiento computacional, pero en los términos básicos. Los informantes clave definieron el pensamiento computacional como el uso de conceptos que se aprenden en el ámbito de la informática para mejorar la resolución de problemas o la realización de actividades en contextos universitarios y en la vida cotidiana.

De manera general, los informantes clave asociaron el pensamiento computacional con cuatro conceptos (descomposición, reconocimiento de patrones, abstracción y algoritmos); indicaron que estos conceptos pueden ser adquiridos mediante la programación (textual o visual) o con la realización de actividades distintas (lectura o resolución de problemas de razonamiento lógico); expresaron que proporciona beneficios (facilidad y comodidad) para la resolución de problemas; evidenciaron seis pasos (comprensión del problema, identificación de la dificultad, descomposición en las partes que lo componen, reconocimiento de patrones, abstracción, y algoritmos) para la resolución de problemas; y manifestaron dos ámbitos (la universidad y la vida cotidiana) de aplicación.

La percepción de los estudiantes como marco conceptual podría ser útil para investigadores, profesores, y estudiantes. Los investigadores podrían utilizar este marco conceptual como referencia de logros alcanzados por estudiantes en asignaturas introductorias de programación; los profesores podrían dar soporte a la reformulación curricular de sus asignaturas o los diseños de nuevas estrategias de aprendizaje; y los estudiantes podrían obtener una referencia de las implicaciones que puede tener el pensamiento computacional para su desenvolvimiento en el contexto universitario o cotidiano.

El trabajo futuro de esta investigación consiste en incorporar formalmente el pensamiento computacional en una asignatura introductoria de programación a nivel universitario. En este sentido, el marco conceptual permite confirmar que la programación por sí misma promueve que los estudiantes construyan una percepción acertada de los fundamentos básicos del pensamiento computacional, por lo tanto, una alternativa para incorporar formalmente el pensamiento computacional consiste en modificar los contenidos de la asignatura para que los estudiantes puedan construir una percepción más amplia del pensamiento computacional según marcos conceptuales más completos, por ejemplo, el marco conceptual presentado por Zapata-Ros (2015).

Referencias

- Adell, J., Llopis, M., Esteve, M. y Valdeolivas, N. (2019). El debate sobre el pensamiento computacional en educación. *RIED. Revista Iberoamericana de Educación a Distancia*, 22(1), 171-186. <https://doi.org/10.5944/ried.22.1.22303>
- Aho, A. (2012). Computation and computational thinking. *The Computer Journal*, 55(7), 832-835. <https://doi.org/10.1093/comjnl/bxs074>

- Basogain, X., Olabe, M. y Olabe, J. (2015). Pensamiento Computacional a través de la Programación: Paradigma de Aprendizaje. *Revista de Educación a Distancia*, 46(6), 1-33. <https://doi.org/10.6018/red/46/6>
- Bayeck, R. (2018). A review of five African board games: is there any educational potential? *Cambridge Journal of Education*, 48(5), 533-552. <https://doi.org/10.1080/0305764X.2017.1371671>
- Bocconi, S., Chiocciariello, A. y Earp, J. (2018). The Nordic approach to introducing Computational Thinking and programming in compulsory education. *Report prepared for the Nordic@ BETT2018 Steering Group*, 397-400.
- Brennan, K. y 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*, 1-25.
- Compañ, P., Satorre, R., Llorens, F. y Molina, R. (2015). Enseñando a programar: un camino directo para desarrollar el pensamiento computacional. *Revista de Educación a Distancia*, 46(11), 1-15.
- Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C. y Woollard, J. (2015). *Computational thinking - A guide for teachers*. United Kingdom, Computing at School.
- Delcker, J. e Ifenthaler D. (2017). Computational Thinking as an Interdisciplinary Approach to Computer Science School Curricula: A German Perspective. En Rich P. y Hodges C. (eds) *Emerging Research, Practice, and Policy on Computational Thinking. Educational Communications and Technology: Issues and Innovations*. Springer, Cham. https://doi.org/10.1007/978-3-319-52691-1_4
- Espinosa, A. (2017). Profesores “migrantes digitales” enseñando a estudiantes “nativos digitales”. *MediSur*, 15(4), 463-473.
- Flick, U. (2015). *El diseño de la investigación cualitativa*. Madrid, España: Ediciones Morata.
- Galeano, M. (2004). *Diseño de proyectos en la investigación cualitativa*. Medellín, Colombia: Universidad EAFIT.
- Hernández, R., Fernández, C. y Baptista, P. (2010). *Metodología de la investigación*. México: McGraw-Hill.
- Jagušt, T., Krzic, A., Gledec, G. y Grgi, M. (2018). Exploring Different Unplugged Game-like Activities for Teaching Computational Thinking. En *IEEE Frontiers in Education Conference (FIE)*. <https://doi.org/10.1109/FIE.2018.8659077>
- Kordaki, M. y Kakavas, P. (2017). Digital Storytelling as an Effective Framework for the Development of Computational Thinking Skills. En *9th International Conference on Education and New Learning Technologies*. <https://doi.org/10.21125/edulearn.2017.2435>
- Mooney, A., y Lockwood, J. (2020). The Analysis of a Novel Computational Thinking Test in First Year Undergraduate Computer Science Course. *All Ireland Journal of Higher Education*, 12(1), 1-26.

- Park, S., Song, K. y Kim, S. (2015). EEG Analysis for Computational Thinking based Education Effect on the Learners' Cognitive Load. En *Proceedings of the Applied Computer and Applied Computational Science*, Kuala Lumpur, Malaysia.
- Pérez, J. (2019a). El pensamiento computacional en la vida cotidiana. *Revista Scientific*, 4(13), 293-306. <https://doi.org/10.29394/Scientific.issn.2542-2987.2019.4.13.15.293-306>
- Pérez, J. (2019b). DINADI: una estrategia para el diagnóstico de nativos digitales en el ámbito universitario. *Revista Paradigma*, 40(1), 56-75.
- Pérez, J. y Azuaje, M. (2019). LE1: una estrategia amistosa para un curso introductorio de programación. *Revista Educación En Ingeniería*, 14(28), 45-53. <https://doi.org/10.26507/rei.v14n28.998>
- Pérez, J. y Castro, J. (2018). LRS1: un robot social de bajo costo para la asignatura "Programación 1". *Revista Colombiana de Tecnologías de Avanzada*, 2(32), 68-77. <https://doi.org/10.24054/16927257.v32.n32.2018.3028>
- Pérez, J. y Pedroza, O. (2018). LM1: una metodología de estudio para la asignatura "Programación 1". *Revista Educere*, 22(73), 635-648.
- Perkins, D. y Swartz, R. (1991). The nine basics of teaching thinking. En Costa, A., Bellanca, J. y Fogarty R. (eds). *If minds matter: A forward to the future*, 2, 53-69. Palatine: Skylight Publications.
- Polanco, N., Ferrer, S. y Fernández, M. (2021). Aproximación a una definición de pensamiento computacional. *RIED. Revista Iberoamericana de Educación a Distancia*, 24(1), 55-76. <https://doi.org/10.5944/ried.24.1.27419>
- Sondakh, D., Osman, K. y Zainudin, S. (2020). A Proposal for Holistic Assessment of Computational Thinking for Undergraduate: Content Validity. *European Journal of Educational Research*, 9(1), 33-50. <https://doi.org/10.12973/eu-jer.9.1.33>
- Strauss, A. y Corbin, J. (2002). *Bases de la investigación cualitativa. Técnicas y procedimientos para desarrollar la teoría fundamentada*. Colombia: Universidad de Antioquia.
- Valverde, J., Fernández, M. y Garrido, M. (2015). El pensamiento computacional y las nuevas ecologías del aprendizaje. *Revista de Educación a Distancia*, 46(3), 1-18.
- Wing, J. (2006). Computational Thinking. It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use. *Communications of the ACM*, 49(3), 33-35. <https://doi.org/10.1145/1118178.1118215>
- Wing, J. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717-3725. <https://doi.org/10.1098/rsta.2008.0118>
- Zapata-Ros, M. (2015). Pensamiento computacional: Una nueva alfabetización digital. *Revista de Educación a Distancia*, 46(4), 1-47. <https://doi.org/10.6018/red/46/4>
- Zapata-Ros, M. (2019). Computational Thinking Unplugged. *Education in the Knowledge Society*, 20(1), 1-29. https://doi.org/10.14201/eks2019_20_a18