Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

Cifuentes Osorio, Y. E., Torres Pardo, I. D., & González Gil, M. (2023, mayo-agosto). Modelos mentales y algoritmos de programación en estudiantes de media técnica en informática. *Revista Virtual Universidad Católica del Norte*, (69), 98-134.
https://www.doi.org/10.35575/rvucn.n69a5

Modelos mentales y algoritmos de programación en estudiantes de media técnica en informática

Mental models and programming algorithms in computer science technical high school students

Yazmín Eliana Cifuentes Osorio
Candidata a doctora en Educación
Facultad de Educación y Humanidades, Escuela de Posgrados, Universidad Católica Luis Amigó
Medellín, Colombia
yazmin.cifuentesos@amigo.edu.co
Orcid: https://orcid.org/0000-0001-7659-4798
CvLAC:
https://scienti.minciencias.gov.co/cvlac/visualizador/generarCurriculoCv.do?cod_rh=0001469886

Ingrid Durley Torres Pardo
Doctora en Ingeniería
Facultad de Ingenierías y Arquitectura, Universidad Católica Luis Amigó
Medellín, Colombia
ingrid.torrespa@amigo.edu.co
Orcid: https://orcid.org/0000-0003-4503-7512
CvLAC:
https://scienti.minciencias.gov.co/cvlac/visualizador/generarCurriculoCv.do?cod_rh=0001404075

Marta González Gil
Doctora en Educación
Facultad de Educación y Humanidades, Universidad Católica Luis Amigó
Medellín, Colombia
marta.gonzalezgi@amigo.edu.co
Orcid: https://orcid.org/0000-0002-7038-1027
CvLAC:
https://scienti.minciencias.gov.co/cvlac/visualizador/generarCurriculoCv.do?cod_rh=0001410807

Recibido: 21 de julio de 2022
Evaluado: 16 de enero de 2023
Aprobado: 25 de abril de 2023

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

Tipo de artículo: Investigación.

Resumen

El presente artículo analiza la asociación entre la tipología de estructuras algorítmicas de programación y los modelos mentales construidos para su resolución. El estudio se inscribe en el paradigma cuantitativo con un diseño cuasi-experimental, y contó con la participación de 95 estudiantes de grado undécimo de media técnica en informática. La metodología implicó evaluar los procesos desarrollados por los estudiantes, a través de los modelos obtenidos en una prueba de resolución de problemas algorítmicos; los resultados fueron sometidos a un análisis descriptivo, con el fin de establecer diferencias estadísticamente significativas entre los niveles de consistencia de los modelos asociados al proceso de resolución y a los tipos de estructuras. Lo anterior, analizado en razón a la verosimilitud Chi-cuadrado, complementado con la magnitud del efecto V de Cramér; también, se aplicó la prueba de *Dwass-Steel-Critchlow-Fligner* (DSCF), para realizar comparaciones entre las estructuras. Los resultados indicaron modelos más consistentes en la estructura secuencial, en contraste con los modelos inconsistentes y ambiguos predominantes en las estructuras condicionales, cíclicas y anidadas. Las principales conclusiones exponen una relación significativa entre los modelos mentales y la tipología de las estructuras algorítmicas de programación, evidenciando la necesidad de enfocar las estrategias didácticas en la resolución de problemas.

Palabras clave: Algoritmo; Estudiantes; Modelos mentales; Programación informática; Resolución de problemas; Visualización.

Abstract

This paper analyzes the association between the typology of algorithmic programming structures and the mental models constructed for resolution. The study is part of the quantitative paradigm with a quasi-experimental design and included the participation of ninety-five students of eleventh grade of computer science technical high school. The methodology involved evaluating the processes developed by the students, through the models obtained in an algorithmic problem-solving test; the results obtained, were submitted to a descriptive analysis, in order to establish

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

statistically significant differences between the levels of consistency of the models associated with the resolution process and the types of structures. The above, was analyzed in terms of Chi-square likelihood, supplemented by the magnitude of the Cramer V effect; the Dwass-Steel-Critchlow-Fligner (DSCF) test, was also applied to make comparisons between structures. The results indicated more consistent models in the sequential structure, in contrast to inconsistent and ambiguous models prevalent in conditional, cyclical, and nested structures. The main conclusions show a meaningful relationship between mental models and the typology of algorithmic programming structures, demonstrating the need to focus didactic strategies on problem solving.

Key words: Algorithms; Students; Mental models; Computer programming; Problem solving; Visualization.

Introducción

La resolución de algoritmos de programación es un proceso fundamental que hace parte del pensamiento lógico requerido para acceder a los conocimientos de programación, sea en la educación secundaria o en la formación profesional. Las habilidades necesarias para resolver este tipo de problemas implican, en su mayoría, la operación de conceptos abstractos y, aunque se asume que el estudiante que transita en la media técnica se encuentra en el estadio de las operaciones formales, los procesos asociados a esta etapa pueden no estar acordes al desarrollo cognitivo (Inhelder & Piaget, 1973). De este modo, autores como Goldson et al. (1993), Barland et al. (2009), Bubica y Boljat (2015), Insuasti (2016), Vrachnos y Jimoyiannis (2017), Alzahrani et al. (2019), Izu et al. (2019), y Scapin y Mirolo (2020) coinciden en afirmar que el aprendizaje de la programación es un proceso muy complejo para los estudiantes.

La situación expuesta ha sido tema de interés no solamente en el ámbito académico, sino también en la agenda política de varios países (Kirçali & Özdener, 2022; Saxena et al., 2020). De manera particular, en Colombia se ha dado respuesta desde las políticas educativas que buscan fortalecer las competencias en lenguaje y matemáticas; estas orientaciones se materializan en los lineamientos curriculares, los estándares de competencias y los derechos básicos del aprendizaje.

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

Estas orientaciones legislativas están contempladas desde la organización del sistema educativo colombiano que ha sido establecido por los niveles preescolar, básica y media. Este último nivel ha sido fortalecido con la formación en media técnica, como una etapa educativa que prepara a los estudiantes en unas competencias específicas para el mundo laboral inmediato, en paralelo con el acceso y permanencia en la educación superior (Ley 115 de 1994, Artículo 32). Igualmente, con la Ley 749 de 2002, este ciclo educativo puede ser articulado con la formación técnica, tecnológica y profesional a través de ciclos propedéuticos en diversas áreas. De manera particular, los estudiantes que inician la media técnica en informática desarrollan los dos primeros semestres propedéuticos, mientras cursan los grados décimo y undécimo y, posteriormente, pueden continuar hasta el nivel de formación profesional, correspondiente a la Ingeniería en Sistemas.

En el ciclo de bachiller técnico los estudiantes abordan varias asignaturas, entre las que se encuentra Lógica de Programación, orientada al desarrollo del pensamiento lógico, a través de la resolución de algoritmos en un pseudolenguaje denominado pseudocódigo, cuya sintaxis está estructurada por la combinación de instrucciones en lenguaje de programación y lenguaje natural. Se busca que, en este primer proceso, los estudiantes utilicen lápiz y papel para representar la solución en pseudocódigo, como un proceso previo a la traducción a un lenguaje de programación particular que se implementa directamente en el computador.

Sin embargo, en la práctica se observa que este primer acercamiento a la programación se realiza generalmente mediante la utilización de intérpretes de código o de lenguajes de programación, sin tener en cuenta la comprensión lógica de una situación expresada en lenguaje natural, y cuya solución debe traducirse a la sintaxis de un lenguaje formal o pseudolenguaje. En este sentido, autores como Enes da Silveira y Gomes (2019) y Sukirman et al. (2022) sostienen que diseñar el algoritmo solución para un problema representa mayor dificultad que el aprendizaje de un lenguaje de programación. No obstante, es común que los cursos de programación descuiden el proceso de comprensión del programa y se centren en la escritura de código (Lonati & Morpurgo, 2021), puesto que en los profesores predomina más la enseñanza de la sintaxis del lenguaje que las habilidades de resolución (Sambe et al., 2021).

Como resultado del abordaje, que generalmente se realiza en la enseñanza de la asignatura Lógica de Programación, se han observado en los estudiantes dificultades en la comprensión de los enunciados, debido a la tendencia que presentan para operar los datos a partir de reemplazos

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

directos y repetitivos, sin detallar aspectos del significado y de las relaciones que se establecen en el enunciado del problema. Adicionalmente, se evidencian niveles bajos en la abstracción, contrastación, secuenciación y categorización; procesos que son necesarios para realizar la representación de la solución de un problema algorítmico. Al respecto, Desoete et al. (2003) sostienen que es fundamental comprender la situación problema para elaborar un modelo de ella y para poder resolverla correctamente. Igualmente, Izu et al. (2019) argumentan que la enseñanza de la escritura de código es tan importante como enseñar a leer e interpretar código, dado que la relación causal entre los enunciados es importante para comprender y describir cómo funciona el programa.

En relación con las dificultades planteadas, involucrar a los principiantes en la enseñanza de la programación representa un reto (Ma et al., 2023; Twigg et al., 2019); incluso algunos errores presentados por los estudiantes en los cursos introductorios son tan complejos que pueden generarles frustración, disminuir su voluntad para aprender y llevarlos a la deserción (Alzahrani et al., 2019; Chang et al., 2022). Igualmente, Cheah (2020) afirma que la programación representa un problema en el plan de estudios de informática en el ámbito mundial, como consecuencia de los altos índices de abandono y de fracaso escolar que ocurren desde los primeros semestres. En línea con este planteamiento, Díaz et al. (2022) sostienen que uno de los factores internos que influye en la permanencia académica de los estudiantes está asociado con el rendimiento académico. De ahí que se hace necesario implementar estrategias de seguimiento y apoyo académico, las cuales ayuden a mitigar los factores de riesgo que obstaculizan la permanencia escolar (Chalpartar Nasner et al., 2022).

A partir del contexto esbozado, se plantea un acercamiento a la resolución de problemas algorítmicos, desde la perspectiva de los modelos mentales. En la línea de las investigaciones que han abordado la relación entre modelos mentales y resolución de problemas se encuentran estudios realizados por Khemlani y Johnson-Laird (2017), Teichert et al. (2017), Muñoz-Campos et al. (2018), Supriyadi et al. (2018), Slapničar et al. (2018), Derman et al. (2019), Álvarez et al. (2020), Praisri y Faikhamta (2020), Prayekti et al. (2020), Burkholder et al. (2020), y Khemlani y Johnson-Laird (2022); investigadores que se han acercado a la relación objeto del presente estudio, en campos de conocimiento como la química, la física y las matemáticas. En este horizonte conceptual predominan dos líneas investigativas: una de ellas centrada en el análisis de los modelos mentales

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

que construyen los estudiantes durante la comprensión de conceptos, y la otra, enfocada en la abstracción de los modelos mentales y su aplicación en procesos de transferencia a nuevas situaciones problema.

Los hallazgos en el área específica de la programación dan cuenta de una preocupación generalizada por las habilidades para el desempeño en un lenguaje de programación específico, como lo evidencian estudios publicados por Goltermann y Höppner (2017), Spangsberg et al. (2018) y Andrzejewska y Skawińska (2020), en los que se desarrollaron estrategias para mejorar el rendimiento académico de estudiantes universitarios en la implementación de programas directamente en un lenguaje de programación. En cuanto a la educación primaria, Aggarwal et al. (2018) realizaron un estudio de pensamiento en voz alta de estudiantes de cuarto y quinto grado, durante el aprendizaje del lenguaje de programación visual *Kodu Game Lab*, que está basado en el desarrollo de juegos en 3D. Igualmente, los investigadores argumentan que, durante la última década, han implementado diferentes estrategias para promover el aprendizaje de las ciencias de la computación a nivel de la educación secundaria.

El interés de la investigación surgió a partir del escaso desarrollo de habilidades en la resolución de algoritmos de programación que presentaban los estudiantes de grado undécimo de media técnica en informática, a pesar de formarse bajo los preceptos de un currículo estructurado desde las áreas básicas del conocimiento, fortalecidas con la media técnica. No obstante, esta situación no se inscribe solamente en el plano contextual y específico, pues trasciende el horizonte teórico, dado que se hizo un rastreo exhaustivo y se encontraron pocas investigaciones que se ocuparan de este fenómeno. Por consiguiente, la contribución teórica garantiza la novedad de la investigación por tratarse de un campo de conocimiento que ha sido escasamente explorado, en relación con los problemas algorítmicos en pseudocódigo. Al respecto, Bubica y Boljat (2015) sostienen que se han realizado muy pocas investigaciones dedicadas a analizar los modelos mentales de programadores novatos, en relación con la gran cantidad de estudios que han abordado los modelos mentales humanos de varios fenómenos.

En Colombia, Zúñiga et al. (2016) estudiaron la relación entre el desarrollo de habilidades de pensamiento, los modelos mentales compartidos y los mecanismos de abstracción en una población de niños con edades entre 8 y 12 años; los investigadores evaluaron los mecanismos de abstracción de los estudiantes en el entorno de programación por bloques *Scratch*. Entre los

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

hallazgos del estudio, los autores mencionan la necesidad de conocer el uso de las herramientas del programa, así como su interacción para que fluya el proceso de abstracción de los estudiantes.

En relación con la enseñanza de los algoritmos en pseudocódigo, Kwon (2017) analizó los modelos mentales asociados a los conceptos de programación en general, en problemas que podían resolverse utilizando declaraciones condicionales; la población estuvo conformada por un grupo de 12 maestros de pregrado, quienes realizaron la solución de los algoritmos, utilizando lápiz y papel. Como resultado de la investigación, se identificaron dificultades en los estudiantes para diseñar planes de solución que pudieran ser ejecutados por computadoras.

De manera similar, una investigación realizada por Izu et al. (2019) buscó fortalecer las habilidades para la comprensión de los programas en estudiantes de K-12 y de los primeros cursos de programación a nivel universitario; los investigadores realizaron una revisión bibliográfica, y entre los resultados obtenidos recopilaron más de 60 actividades de aprendizaje, categorizadas en una matriz de acuerdo a cuatro niveles de comprensión del programa: atómica, por bloques, relacional y de estructura macro; correspondientes a tres categorías de complejidad: textual, de ejecución y de intención. Igualmente, la investigación generó trayectorias teóricas para el aprendizaje de actividades complejas, a través de una guía que orienta a los profesores sobre los aspectos específicos que se están fomentando.

Las investigaciones anteriores trazaron el camino para analizar la relación existente entre la tipología de estructuras algorítmicas de programación y los modelos mentales que los estudiantes elaboran durante el proceso de resolución, con el propósito de plantear alternativas pedagógicas que permitan potenciar las habilidades de los estudiantes para estructurar su pensamiento formal.

Marco teórico

Entre los referentes teóricos que estructuran el marco conceptual de la investigación, se abordó el concepto de modelos mentales, principalmente desde las propuestas realizadas por Johnson-Laird como autor fundamental en el campo; estos planteamientos se complementaron con los aportes de investigadores como Rapp (2005), Khemlani et al. (2013) e Izu et al. (2019). Asimismo, la línea teórica asociada a los algoritmos de programación se retomó, en su mayoría, a

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

partir de autores con trayectoria conceptual importante en este dominio específico de conocimiento, como lo son Joyanes Aguilar (2008) y Oviedo (2005).

Modelos mentales

Cuando un estudiante se enfrenta a la resolución de un algoritmo, el primer paso que debe realizar consiste en representar su solución, a través de la sintaxis del pseudocódigo. Este proceso de representación exige tanto la comprensión de la situación problema, que se describe en el enunciado, como la traducción del problema en un conjunto de instrucciones, expresadas en las reglas sintácticas y lógicas bajo las que opera el pseudolenguaje. En este sentido, Bayman y Mayer (1983) plantean que los estudiantes principiantes en programación pueden desarrollar modelos mentales para el lenguaje, entendidos como la concepción del usuario sobre el procesamiento de información que ocurre internamente en el computador entre la entrada y la salida.

La relación entre la percepción del mundo y el discurso sobre el mundo a partir de modelos fue propuesta por Johnson-Laird (1983); este autor sostiene que cuando una persona asimila una descripción del mundo, puede representarla; es decir, puede construir un modelo mental del mundo, el cual se basa en el significado de la descripción y en su conocimiento. La teoría del modelo de Johnson-Laird parte de tres supuestos: el primero, se basa en que cada modelo mental representa lo que es común a un conjunto distinto de posibilidades; en el segundo, plantea que los modelos mentales son icónicos en la medida de lo posible; es decir, la estructura de una representación corresponde a la estructura de lo que representa; y en el tercero, expone que los modelos mentales de descripciones representan lo que es verdadero, a expensas de lo falso.

A partir de este tercer supuesto, el autor sustenta que el principio de verdad bajo el que operan las representaciones reduce la carga que los modelos colocan en la memoria de trabajo del sujeto. Por tanto, al razonar a través de modelos no se concluye algo que solamente repita las premisas, o que sea una conjunción de ellas, o que agregue una alternativa a las posibilidades a las que se refieren, sino que se busca una relación o propiedad que no fue afirmada explícitamente en las premisas; es decir, dependiendo de si se cumple en todos, la mayoría o algunos de los modelos, se obtiene una conclusión de su necesidad, probabilidad o posibilidad. En esta línea, Johnson-Laird et al. (1992) plantean el razonamiento como un proceso semántico basado en modelos mentales,

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

donde la información semántica se mantiene, mientras se simplifica y se llega a una nueva conclusión.

Asimismo, Johnson-Laird (2010) argumenta que cuando las personas razonan parecen reunir sus estrategias a medida que exploran problemas, utilizando sus tácticas inferenciales existentes, como la capacidad de agregar información a un modelo de posibilidad. Una vez que han desarrollado una estrategia para un tipo particular de problema, tienden a controlar su razonamiento y a formular explicación a través de la habilidad inductiva, porque usan el conocimiento para ir más allá del contenido estricto de las premisas, en este caso en el enunciado.

Frente a estas consideraciones, Rapp (2005) afirma que los modelos mentales son estructuras de memoria que se pueden emplear para profundizar en la comprensión de la información que se presenta y no quedarse en el nivel de interpretación superficial. De esta forma, se plantea que, desde la asimilación de los modelos mentales como representación de un estado de cosas con sus supuestos principales, se pretende determinar la asociación entre las estructuras algorítmicas de programación (secuenciales, condicionales, cíclicas y anidadas) y los modelos mentales que construyen los estudiantes para comprender y explicar cada tipo de estructura, a medida que transitan por el esquema de resolución, hasta la fase de simulación y deducción.

En esta línea, Khemlani et al. (2013) argumentan que la mente posee una gran habilidad para hacer simulaciones mentales; el sujeto construye modelos mentales que representan distintas posibilidades, o que se despliegan en el tiempo en una secuencia cinemática, y basan sus conclusiones en ellos; de allí que la propia teoría del modelo se basa en otro tipo de representación para capturar el significado de una afirmación, que luego se utiliza para construir modelos. Es así como el modelo mental representa una posibilidad de un pequeño número de alternativas que los humanos pueden elaborar en cualquier situación (Khemlani & Johnson-Laird, 2022).

En consecuencia, la enseñanza de la programación implica la manipulación de entidades abstractas; este proceso requiere que el estudiante no solo plantee relaciones lógicas que conjugan el dominio sintáctico y semántico sobre una situación problema, sino el uso de estrategias que le permitan diseñar e implementar soluciones correctas, que sean aplicables a problemas generalizados o del mismo tipo. En este sentido, Izu et al. (2019) sostienen que la comprensión del programa es un proceso que le permite al estudiante construir su modelo mental. Según los autores, el proceso de comprensión requiere la capacidad de leer, interpretar y explicar el código, debido a

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

que el modelo mental incluye los elementos y la estructura del programa, a partir de la codificación, ejecución y el propósito de los bloques de programación.

Algoritmos de programación

Las personas en la vida cotidiana constantemente están interactuando con algoritmos para realizar determinadas acciones, como bañarse, vestirse, hacer una llamada telefónica e incluso iniciar una conversación, a través de WhatsApp; estas actividades pueden ser descritas como un conjunto de acciones paso a paso. Un algoritmo se define como una secuencia de pasos, procedimientos o instrucciones para alcanzar un resultado o resolver un problema (Cairó, 2005; Hernández, 2010; Oviedo, 2005). De acuerdo con Joyanes Aguilar (2008), la secuencia de pasos debe ser precisa, definida y finita; es decir, debe indicar el orden en que se realiza cada paso y si se sigue varias veces se debe obtener el mismo resultado. El diseño del algoritmo es un proceso independiente a la implementación posterior en un lenguaje de programación. El orden en que se ejecutan las sentencias de un algoritmo determina los tipos de estructuras de control de flujo básicas, como lo son las estructuras secuenciales, repetitivas y selectivas.

A continuación, se describen cada uno de los tipos de estructuras, teniendo en cuenta los planteamientos de Joyanes Aguilar (2008). En la estructura secuencial, las sentencias se ejecutan una después de otra, en el orden en que se sitúan dentro del programa, de forma que la salida de una es la entrada de la siguiente y así sucesivamente hasta finalizar el proceso. La estructura condicional se emplea para tomar decisiones lógicas respecto a la evaluación de una condición; de acuerdo con el resultado de esa evaluación, se realiza una acción u otra. La estructura cíclica, por su parte, permite realizar una secuencia de instrucciones un número determinado de veces, con base en el valor de verdad que arroje la evaluación de una expresión lógica, lo cual le permite al algoritmo tomar la decisión de repetir o dejar de ejecutar el grupo de instrucciones.

Aunque en la literatura no se asume el anidamiento como una estructura algorítmica en sí, en la investigación se le dio ese tratamiento, dado que un anidamiento modifica el control de flujo en la ejecución de las secuencias de un programa. Los anidamientos implican que se inserte una estructura dentro de otra y no son exclusivos de un tipo específico. En el caso de los condicionales, los anidamientos implican que, después de haber tomado una decisión, es necesario tomar otra

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

(Chaves Torres, 2017), y en cuanto a los ciclos, según Joyanes Aguilar (2008), el orden de ejecución está conformado por uno o más ciclos internos y uno externo que determina las iteraciones de los primeros. De esta forma, todas las estructuras mencionadas se pueden anidar, siempre y cuando no se presenten solapamientos entre la interna y la externa.

La solución de un algoritmo de programación está enmarcada en un conjunto de fases que inicia con la definición del problema equivalente al enunciado, en el cual, a través de un texto, se especifica la información relacionada con la situación que se espera solucionar, así como los datos indispensables para plantear la solución. Seguidamente, se desarrolla la fase de análisis que implica el planteamiento matemático y lógico de la solución del problema, a partir de los datos que se especifican en el enunciado y que son necesarios para su resolución (Joyanes Aguilar, 2008). La etapa de diseño se realiza a partir de la descripción paso a paso de las instrucciones lógicas que dan solución al problema, con la finalidad de obtener una solución coherente con los requerimientos. Finalmente, se realiza la verificación, con el propósito de comparar los datos de la salida obtenida con los resultados esperados, y establecer si el algoritmo funciona o si es necesario realizar ajustes (Oviedo, 2005). En relación con la representación de la solución de un algoritmo, debe ser independiente del lenguaje de programación en el que se codificará posteriormente. Entre los métodos de representación más usuales se tienen el pseudocódigo, el diagrama de flujo y los diagramas de N-S (Nassi-Schneiderman).

Adicionalmente, para el propósito de la investigación que dio origen a este texto, se utilizó el concepto de pseudocódigo, en línea con los planteamientos de Joyanes Aguilar (2008). El pseudocódigo es un lenguaje de especificaciones de algoritmos que permite escribir la secuencia de instrucciones en palabras similares al lenguaje natural, con el fin de facilitar la escritura y la lectura del programa. Este tipo de representación permite que el programador se centre en el análisis lógico y en las estructuras de control, sin tener que utilizar la sintaxis propia de un lenguaje de programación, debido a que se emplean términos que corresponden a traducciones libres de palabras reservadas en estos lenguajes.

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

Marco metodológico

El estudio se desarrolló en tres fases: conceptual, contextual y de análisis. En la fase conceptual se realizó la revisión de literatura, se perfiló el objeto de estudio y se definieron los objetivos de la investigación; en la fase contextual, se concretó el paradigma y el método de investigación, que dieron origen al trabajo de campo; y en la fase de análisis, se dio tratamiento estadístico a los datos recabados. La investigación fuente de este artículo, se inscribió en el horizonte epistemológico empírico analítico y estuvo enfocada en explicar en qué circunstancias se manifiesta la asociación entre las estructuras que subyacen a los algoritmos de programación y los modelos mentales que elaboran los estudiantes durante su resolución. Asimismo, se planteó un alcance explicativo en la medida del nivel de comprensión que se esperaba del fenómeno descrito y de las causas que lo determinaban. Se optó por un diseño cuasi-experimental, porque los participantes estaban previamente distribuidos en grupos y no fue necesaria una selección aleatoria.

La población estuvo conformada por los estudiantes matriculados en las 72 instituciones educativas públicas de Medellín, articuladas a la formación media técnica en informática, en el grado undécimo durante el 2021. Este grado fue elegido a partir de criterios conceptuales propios de la asignatura lógica de programación y por el nivel de desarrollo de las habilidades en resolución de problemas algorítmicos, debido a que en el grado décimo los estudiantes abordan por primera vez la asignatura y en el grado undécimo la profundizan. De esta manera se garantizó, desde el plan de estudios, que los estudiantes tuvieran un dominio mínimo de la resolución de las estructuras de programación.

La muestra se determinó a través de un muestreo no probabilístico intencional, que obedeció a la disponibilidad de las instituciones educativas (IE), las condiciones logísticas, de infraestructura y la disposición de los directivos y docentes encargados de la formación media técnica en cada institución para aceptar la participación en la convocatoria. De otro lado, como criterios de exclusión, se consideraron los estudiantes que no respondieron completamente el instrumento y otros factores que influyeron en la pérdida de datos, como la inasistencia o incapacidad médica prolongada. Finalmente, se contó con la participación de 4 instituciones educativas, para un total 95 estudiantes, de los cuales el 53.7 % correspondió a mujeres, con una

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

mediana de la edad de 16 años (*RI* = 1) y el 46.3 % a los hombres, con una mediana de la edad de 17 años (*RI* = 2).

Una vez seleccionados los estudiantes, se realizó una reunión para socializar los aspectos básicos de la investigación y poder contar con su participación. Asimismo, se hizo entrega del consentimiento informado, previamente avalado por el Comité de Ética de la Universidad Católica Luis Amigó, el cual fue aprobado y firmado por sus representantes legales. En este documento se notificaron los posibles riesgos, la confidencialidad de la información suministrada, así como el derecho a retirarse de la investigación en el momento que consideraran oportuno. A continuación, se describe la estructura del instrumento que se elaboró para develar los modelos mentales que construyen los estudiantes durante el proceso de resolución de problemas algorítmicos de tipo secuencial, condicional, cíclico y anidado. Igualmente, se detallan los modelos asociados a cada fase de resolución y se explican los niveles de consistencia en que pueden ubicarse.

Instrumento

Con el propósito de identificar los modelos metales construidos por los estudiantes durante la resolución de los algoritmos de programación, se utilizó la técnica de recolección de información primaria basada en cuestionarios, posteriormente analizados estadísticamente. El cuestionario denominado Prueba de Identificación de Modelos Mentales contó con una validación externa, realizada a través del juicio de expertos con formación académica y trayectoria investigativa en el área de la media técnica en Desarrollo de Software. A cada evaluador se le suministró una lista de categorización para evaluar cada pregunta, de acuerdo con una escala con tres valores: 0 = El reactivo no está bien formulado, 1 = El reactivo debe corregirse, 2 = El reactivo está bien formulado. Se procedió a realizar una revisión con base en algunas correcciones sugeridas por los expertos y se generó una segunda versión de la prueba. Principalmente, el cuestionario se conformó por cuatro ejercicios, cada uno asociado a una estructura algorítmica (secuencial, condicional, cíclica y anidamientos). Para cada ejercicio de la prueba de identificación de modelos mentales se plantearon interrogantes que daban cuenta de la metodología para la resolución de problemas algorítmicos. Asimismo, se articularon los cuatro pasos planteados por Polya (2004), en su metodología para resolver problemas en el campo de las matemáticas, debido a que esta área

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

comparte procesos mentales asociados al pensamiento lógico requerido en el ámbito de la programación. En este sentido, los ocho reactivos para cada tipo de ejercicio se formularon con el propósito de develar los modelos mentales que elaboraban los estudiantes durante las fases de definición, análisis del problema, diseño y verificación de la solución. Los 6 primeros consistieron en preguntas de selección múltiple, y los dos últimos consistieron en preguntas abiertas para poder observar los procesos asociados al planteamiento de la solución y a su verificación.

De esta forma, las preguntas de selección múltiple indagaban por la comprensión del enunciado, a través del reconocimiento de los datos de proceso (MMDP), los datos de entrada explícitos (MMDEE) e implícitos (MMDEI), los datos de proceso secuenciales (MMDPS), condicionales (MMDPC), cíclicos (MMDPR) y los datos de salida (MMDS). En relación con las preguntas de tipo abierto, correspondían al diseño de la solución (MMDS) y a la verificación (MMV) (ver Tabla 1).

Tabla 1

Modelos mentales en la resolución de los algoritmos

Identificador	Identificador estudiante
MMDP	Modelo mental datos de proceso
MMDEE	Modelo mental datos de entrada explícitos
MMDEI	Modelo mental datos de entrada usuario
MMDPS	Modelo mental datos de proceso secuenciales
MMDPC	Modelo mental datos de proceso condicionales
MMDPR	Modelo mental datos de proceso repetitivos
MMD	Modelo mental diseño de la solución
MMV	Modelo mental verificación del pseudocódigo

Las 6 primeras categorías (MMDP, MMDEE, MMDEE, MMDEI, MMDPS, MMDPC, MMDPR) se ubicaron en tres niveles de consistencia. De esta forma, si el modelo mental del estudiante develaba un proceso coherente con los requerimientos del enunciado, presentaba un nivel *consistente*; por su lado, cuando evidenciaba el uso de planteamientos incoherentes y de diversa naturaleza, se ubicaba como *ambiguo*; y en el caso en que no respondiera el reactivo o su elección se asociara a formulaciones alejadas completamente de lo requerido, se ubicaba al estudiante en un nivel *inconsistente*.

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

Así las cosas, en el caso de los modelos diseño de la solución (MMDS) y verificación del pseudocódigo (MMV), se agregaron dos categorías adicionales a los niveles de consistencia. De esta manera, se estableció una categoría que corresponde a la no realización del proceso; la categoría de no consistencia se discriminó en inconsistencia aritmética y lógica para el MMDS, y en el caso del MMV se tuvo en cuenta la inconsistencia en datos de entrada y datos de salida.

Para verificar la confiabilidad del instrumento, se aplicó una prueba piloto a 15 estudiantes de una institución educativa que cumplía con las características semejantes a la población objeto de estudio en cuanto a articulación con la media técnica en desarrollo de software y el grado escolar. Durante la aplicación se registraron las dificultades presentadas por los estudiantes para su resolución. Finalmente, se obtuvo una versión final del instrumento.

Respecto a la validez interna, que acompañó la ya enunciada validación externa, se determinó, en un primer momento, a través del alfa de Cronbach (α), y posteriormente se calculó el omega de McDonald (ω), con la finalidad de obtener un mejor ajuste. Los valores de referencia para analizar la fiabilidad del instrumento se tomaron de George y Mallery (2003), quienes establecen que valores de $\alpha > 0.7$ son aceptables, $\alpha > 0.6$ cuestionables, $\alpha > 0.5$ pobres y $\alpha < 0.5$ inaceptable. Debido a que los reactivos correspondientes a los modelos de cada estructura tenían categorías de respuesta diferentes, fueron analizados en dos grupos. El primero conformado por los modelos MMDP, MMDEE, MMDEI, MMDPS, MMDPC y MMDPR; y el segundo, por los modelos MMDS y MMDS.

Para el primer grupo, los valores obtenidos en los coeficientes α y ω (ver Tabla 2) son muy próximos al valor de confiabilidad aceptable (0.7); incluso, autores como Solomon et al. (2006) y Katz (2006) sostienen que valores de alfa superiores a 0.65 indican buena confiabilidad de la escala

Tabla 2

Confiabilidad grupo 1

	Cronbach's α	**McDonald's ω**
Prueba grupo 1	0.689	0.682

Asimismo, el análisis de confiabilidad, realizado para todos los ítems del segundo grupo de modelos, arrojó nivel de confiabilidad alta para los modelos MMDS y MMV (ver Tabla 3).

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

Tabla 3

Confiabilidad grupo 2

	Cronbach's α	McDonald's ω
Prueba grupo 2	0.881	0.883

El instrumento ilustrado en los párrafos anteriores dio origen a un análisis riguroso que incluyó una serie de pruebas propias de la estadística descriptiva. que derivaron en los resultados de la investigación y que se explican en detalle en el análisis de los datos.

Análisis de datos

Para realizar el análisis descriptivo de los modelos se utilizaron frecuencias absolutas, relativas e indicadores de resumen, como la mediana y el rango intercuartílico. En un primer análisis (ver Tabla 4) se establecieron diferencias estadísticas entre los niveles de consistencia y las estructuras algorítmicas, a través de la aplicación de la prueba Chi cuadrado de razón de verosimilitud. Esta prueba se complementó con la medida de magnitud de efecto V de Cramér, con la finalidad de determinar, con mayor precisión, cuál nivel de consistencia predominó entre las diferentes estructuras. En esta línea, los valores de referencia seguidos se retomaron de Dominguez-Lara (2017, 2018), quien destaca la importancia de la medida de magnitud efecto para el análisis de los datos en estudios empíricos.

En un segundo momento del análisis, se realizó la distribución de los valores de los modelos según la estructura mental (ver Tabla 5). Se estableció la relación de los niveles de consistencia con las cuatro estructuras algorítmicas en cada uno de los modelos mentales, a través del método no paramétrico *Kruskal-Wallis,* complementado con el tamaño del efecto épsilon al cuadrado (ε^2), para determinar la magnitud de la relación, a partir de los valores establecidos por Tomczak y Tomczak (2014) y la Universidad de Cambridge (2021). Asimismo, para realizar el análisis de las estructuras para cada modelo, se utilizó la prueba de *Dwass-Steel-Critchlow-Fligner* (DSCF), con el propósito de realizar comparaciones dos a dos entre grupos de estructuras; estos valores se ajustaron con el coeficiente de correlación biserial, para determinar la magnitud del

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

efecto. Los coeficientes ε^2 y rbis, se clasificaron de acuerdo con los valores de referencia sugeridos por Dominguez-Lara (2018).

El análisis observado en la Tabla 5 solamente se realizó para los modelos MMDP, MMDEE, MMDEI, MMDPS, MMDPC y MMDPR, debido a que presentaban una condición más definida, de acuerdo con las categorías ordinales que se plantearon para cada nivel de consistencia (Consistente 3, Ambiguo 2 e Inconsistente 1). En el caso de los modelos MMDS y MMV, no fue posible establecer escala ordinal por la cantidad de categorías (INC ARI, INC LOG, NO R, A, C), en las que se clasificaron las respuestas a las preguntas abiertas que indagaban por los modelos de diseño (MMDS) y verificación (MMV) de la solución.

A modo de cierre, como puede apreciarse en los apartados anteriores, la aplicación del instrumento permitió el desarrollo de los objetivos de la investigación y el análisis realizado a los datos recabados. De esta manera emergieron los resultados de la investigación, tal como se exponen en el apartado siguiente.

Resultados

Los principales resultados se presentan en dos tablas que muestran los resúmenes del análisis estadístico; en la tabla 4 se visualizan los resultados de la distribución de las categorías de los niveles de consistencia de los modelos mentales, de acuerdo con la estructura algorítmica. Asimismo, la Tabla 5, permite apreciar la distribución de los niveles de consistencia de los modelos, según las estructuras algorítmicas.

Tabla 4

Distribución de las categorías de los modelos según estructura algorítmica

		Estructura				Valor p*; [T.E]**
		Secuencial	***Condicional***	***Cíclica***	***Anidada***	
MMDP	***Inconsistente***	4 (4.2 %)	8 (8.4 %)	53 (55.8 %)	47 (49.5 %)	< 0.001; [0.410]
	Ambiguo	6 (6.3 %)	10 (10.5 %)	19 (20 %)	12 (12.6 %)	
	Consistente	85 (89.5 %)	77 (81.1 %)	23 (24.2 %)	36 (37.9 %)	
MMDEE	***Inconsistente***	12 (12.6 %)	78 (82.1 %)	36 (37.9 %)	25 (26.3 %)	< 0.001; [0.419]
	Ambiguo	24 (25.3 %)	7 (7.4 %)	1 (1.1 %)	8 (8.4 %)	

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

	Consistente	59 (62.1 %)	10 (10.5 %)	58 (61.1 %)	62 (65.3 %)	
MMDEI	***Inconsistente***	21 (22.1 %)	9 (9.5 %)	34 (35.8 %)	22 (23.2 %)	< 0.001; [0.189]
	Ambiguo	15 (15.8 %)	13 (13.7 %)	21 (22.1 %)	20 (21.1 %)	
	Consistente	59 (62.1 %)	73 (76.8 %)	40 (42.1 %)	53 (55.8 %)	
MMDPS	***Inconsistente***	12 (12.6 %)	43 (45.3 %)	21 (22.1 %)	57 (60 %)	< 0.001; [0.359]
	Ambiguo	9 (9.5 %)	27 (28.4 %)	7 (7.4 %)	9 (9.5 %)	
	Consistente	74 (77.9 %)	25 (26.3 %)	67 (70.5 %)	29 (30.5 %)	
MMDPC	***Inconsistente***	28 (29.5 %)	17 (17.9 %)	10 (10.5 %)	33 (34.7 %)	< 0.001; [0.246]
	Ambiguo	24 (25.3 %)	51 (53.7 %)	31 (32.6 %)	16 (16.8 %)	
	Consistente	43 (45.3 %)	27 (28.4 %)	54 (56.8 %)	46 (48.4 %)	
MMDPR	***Inconsistente***	30 (31.6 %)	32 (33.7 %)	16 (16.8 %)	54 (56.8 %)	< 0.001; [0.296]
	Ambiguo	44 (46.3 %)	21 (22.1 %)	20 (21.1 %)	14 (14.7 %)	
	Consistente	21 (22.1 %)	42 (44.2 %)	59 (62.1 %)	27 (28.4 %)	
MMDS	***A***	41 (43.2 %)	43 (45.3 %)	34 (35.8 %)	36 (37.9 %)	< 0.001; [0.253]
	C	30 (31.6 %)	6 (6.3 %)	9 (9.5 %)	4 (4.2 %)	
	INC ARI	5 (5.3 %)	2 (2.1 %)	2 (2.1 %)	3 (3.2 %)	
	INC LOG	9 (9.5 %)	30 (31.6 %)	33 (34.7 %)	18 (18.9 %)	
	NO R	10 (10.5 %)	14 (14.7 %)	17 (17.9 %)	34 (35.8 %	
MMV	***A***	46 (48.4 %)	52 (54.7 %)	48 (50.5 %)	37 (38.9 %)	< 0.001; [0.246]
	C	20 (21.1 %)	1 (1.1 %)	1 (1.1 %)	0 (0 %)	
	INC DE	1 (1.1 %)	11 (11.6 %)	10 (10.5 %)	12 (12.6 %)	
	INC DS	1 (1.1 %)	2 (2.1 %)	2 (2.1 %)	1 (1.1 %)	
	NO R	27 (28.4 %)	29 (30.5 %)	34 (35.8 %)	45 (47.4 %)	

*Prueba de razón de verosimilitud - Chi cuadrado.

**Tamaño del efecto: V de Cramér.

En relación con la distribución de las categorías, para el modelo mental datos de proceso (MMDP), se observaron diferencias estadísticamente significativas ($p < 0.001$), con una magnitud de efecto V de Cramér grande [0.410]. Al realizar el análisis interno del MMDP, entre las diferentes estructuras, se encontró que el mayor nivel de consistencia se presentó en la secuencial y la condicional. Respecto al nivel ambiguo, aunque los puntajes fueron relativamente bajos, predominó levemente en la estructura cíclica.

De manera similar que el MMDP, para el modelo mental datos de entrada explícitos (MMDEE) se tuvo un valor $p < 0.001$ y un índice V de Cramér [0.419], lo que sugiere diferencias estadísticas significativas con un tamaño de efecto grande. Al observar la Tabla 5, sobre los niveles de consistencia para cada estructura, los rangos intercuartílicos indican MMDEE consistentes en todas las estructuras, a excepción de la estructura condicional, inconsistentes en cíclica y anidada,

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

y ambiguos en secuencial. Para el caso del modelo mental datos de entrada implícitos (MMDEI), los valores de $p < 0.001$ y del efecto V de Cramér [0.189] permitieron concluir diferencias significativas entre los modelos, con una magnitud de efecto mediano.

El análisis de los resultados, asociados a los datos de proceso, se discriminó en tres tipos de modelo: secuencial (MMDPS), condicional (MMDPC) y cíclico (MMDPR); el valor de $p <$ 0,001 para todos los modelos de proceso mostró diferencias estadísticamente significativas. Sin embargo, el valor del tamaño del efecto reportó magnitud grande para el MMDPS [0.359] y MMDPR [0.296], y mediana para el MMDPC [0.246], de acuerdo con los valores del índice V de Cramér encontrados para estos procesos.

Respecto a la distribución de las categorías, para el modelo mental diseño de la solución (MMDS) se observaron diferencias estadísticamente significativas ($p < 0.001$), con una magnitud de efecto V de Cramér grande [0.253]. Al observar los valores internos de las diferentes estructuras, se concluyó que la mayor consistencia se presentó en la secuencial, con una calificación de 31.6 %. Sin embargo, los niveles más altos de todas las estructuras se presentaron en la ambigüedad, incluyendo la secuencial.

En cuanto al modelo mental de verificación de la solución (MMV), los valores de $p < 0.001$ y V de Cramér [0.246] denotaron diferencias estadísticamente significativas, con un tamaño de efecto mediano. En todas las estructuras se presentaron los niveles más altos en la ambigüedad. En segundo lugar, se tienen los puntajes más altos en la no realización de este proceso, predominando en la estructura anidada. De otro lado, el nivel de consistencia mayor se presentó en la estructura secuencial.

Tabla 5

Distribución de los niveles de consistencia de los modelos según estructura algorítmica

n=95	Estructura*				Valor p (ε^2)**	Estructura post-hoc: valor p; (rbis)***
	Secuencial	***Condicional***	***Cíclica***	***Anidada***		
MMSDP	3 [0]	3 [0]	1 [1]	2 [2]	< 0.001 (0.3311)	Secuencial vs Condicional: 0.354; (0.085) Secuencial vs Cíclica: < 0.001; (0.679) Secuencial vs Anidada: < 0.001; (0.542) Condicional vs Cíclica: < 0.001; (0.610) Condicional vs Anidada: < 0.001; (0.473)

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

						Cíclica vs Anidada: 0.484; (0.108)
MMDEE	3 [1]	1 [0]	3 [2]	3 [2]	< 0.001 (0.2566)	Secuencial vs Condicional: < 0.001; (0.714) Secuencial vs Cíclica: 0.472; (0.105) Secuencial vs Anidada: 0.987; (0.024) Condicional vs Cíclica: < 0.001; (0.486) Condicional vs Anidada: < 0.001; (0.597) Cíclica vs Anidada: 0.748; (0.071)
MMDEI	3 [1]	3 [0]	2 [2]	3 [1]	< 0.001 (0.0698)	Secuencial vs Condicional: 0.079; (0.163) Secuencial vs Cíclica: 0.033; (0.208) Secuencial vs Anidada: 0.890; (0.053) Condicional vs Cíclica: < 0.001; (0.375) Condicional vs Anidada: 0.008; (0.222) Cíclica vs Anidada: 0.159; (0.161)
MMDPS	3 [0]	2 [2]	3 [1]	1 [2]	< 0.001 (0.2065)	Secuencial vs Condicional: < 0.001; (0.523) Secuencial vs Cíclica: 0.546; (0.085) Secuencial vs Anidada: < 0.001; (0.519) Condicional vs Cíclica: < 0.001; (0.413) Condicional vs Anidada: 0.676; (0.086) Cíclica vs Anidada: < 0.001; (0.423)
MMDPC	2 [2]	2 [1]	3 [1]	2 [2]	0.006 (0.0330)	Secuencial vs Condicional: 0.895; (0.055) Secuencial vs Cíclica: 0.074; (0.185) Secuencial vs Anidada: 0.999; (0.007) Condicional vs Cíclica: 0.001; (0.286) Condicional vs Anidada: 0.946; (0.044) Cíclica vs Anidada: < 0.086; (0.180)
MMDPR	2 [1]	2 [2]	3 [1]	1 [2]	< 0.001 (0.1011)	Secuencial vs Condicional: 0.322; (0.135) Secuencial vs Cíclica: < 0.001; (0.388) Secuencial vs Anidada: 0.203; (0.154) Condicional vs Cíclica: 0.027; (0.213) Condicional vs Anidada: 0.014; (0.234) Cíclica vs Anidada: < 0.001; (0.432)

*Los datos se reportan en Mediana [RI]

** Prueba de Kruskal-Wallis (Tamaño del efecto: épsilon al cuadrado)

***Prueba DSCF (Tamaño del efecto: coeficiente de correlación biserial)

A partir de los valores $p < 0.001$ y $\varepsilon^2 = 0.3311$, obtenidos para el MMDP, fue posible establecer que existen diferencias estadísticamente significativas entre los niveles de consistencia de las cuatro estructuras algorítmicas, con un tamaño de efecto grande. El nivel consistente más alto (3 [0]) se presentó en las estructuras secuencial y condicional; la medida de variabilidad que indica que no se encuentran mayores cambios en esa consistencia. Los modelos más ambiguos se

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

ubicaron en la estructura anidada; la medida de variabilidad señaló que algunos casos se pueden bajar al inconsistente y otros ubicarse en el consistente 2 [2]. En la estructura cíclica, se observa que la gran mayoría es inconsistente con la posibilidad de moverse en el ambiguo 1 [1].

En relación con la comparación entre estructuras, se obtuvo un *p* valor muy pequeño para los modelos MMDP secuenciales y condicionales ($p < 0.001$), respecto a los MMDP cíclicos y anidados. Al observar la magnitud del efecto (rbis = 0.679), se determinó que los mayores cambios se presentaron en la estructura secuencial vs cíclica, seguido de la comparación entre los MMDP condicionales y cíclicos, con un rbis = 0.610. De otro lado, no se encontraron diferencias estadísticamente significativas entre la estructura secuencial y condicional, y entre la cíclica con la anidada.

Al observar los resultados del análisis correspondiente al modelo mental datos entrada explícitos (MMDEE), los valores $p < 0.001$ y $\varepsilon^2 = 0.2566$ indicaron diferencias estadísticamente significativas entre las categorías de los niveles de consistencia y las estructuras algorítmicas, con una magnitud de efecto grande. Aunque se encontraron niveles de consistencia altos para todos los procesos, a excepción del condicional, la medida de variabilidad indicó que, en la estructura secuencial, algunos casos se pueden ubicar en el nivel ambiguo 3 [1]; para las cíclicas y anidadas, posiblemente algunos modelos puedan bajar a inconsistentes y otros a ambiguos 3 [2]. Ninguna estructura se ubicó en el nivel de ambigüedad, y en la estructura condicional se observó que la gran mayoría es inconsistente, sin cambios significativos en ese nivel, según el rango intercuartílico obtenido 1 [0].

En cuanto al análisis de las cuatro estructuras para el MMDEE, los valores *p* resultantes, al realizar las comparaciones entre la secuencial y la condicional, y la condicional con la cíclica y con la anidada ($p < 0.001$), permitieron determinar diferencias significativas entre los modelos anteriores. Los mayores cambios se apreciaron entre la estructura secuencial y condicional, dado el tamaño del efecto (rbis = 0.714).

En el modelo mental datos de entrada del usuario (MMDEI), se indicaron diferencias estadísticamente significativas entre las categorías de los modelos para todas las estructuras, con un valor $p < 0.001$ y con un tamaño de efecto moderado $\varepsilon = 0.0693$. Igualmente, se obtuvo un nivel consistente en las estructuras secuencial, condicional y anidada, con mayores cambios en las dos primeras 3 [1], con factibilidad de moverse a ambigua y la condicional representó más estabilidad

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

3 [0]. La estructura cíclica presentó nivel ambiguo, con posibilidad de ubicarse en consistente y en inconsistente 2 [2], de acuerdo con las medidas de variabilidad resultantes. Respecto a los resultados de las comparaciones del MMDEI, entre todas las estructuras, solamente se presentaron diferencias estadísticamente significativas entre la condicional y la cíclica, con una magnitud de efecto mediana, de acuerdo con los valores $p < 0.001$ y rbis = 0.375.

Respecto al modelo mental datos de proceso secuenciales (MMDPS), el *p* valor obtenido indicó diferencias estadísticamente significativas $p < 0.001$, con un tamaño de efecto pequeño $\varepsilon^2 = 0.04$. El nivel consistente se ubicó principalmente en la estructura secuencial, sin mayores cambios 3 [0], y en la cíclica, con posibilidad de moverse al ambiguo 3 [1]. La ambigüedad más alta se presentó en la estructura condicional con un índice de variabilidad entre inconsistente y consistente 2 [2]. La categoría inconsistente se presentó en los anidamientos, con la opción de desplazarse al ambiguo y al consistente 1 [2]. El análisis de MMDPS entre las estructuras arrojó diferencias estadísticas $p < 0.001$, con magnitud de efecto grande, entre las comparaciones de la secuencial con la condicional rbis = 0.523, y la secuencial con la anidada rbis = 0.519.

Para el caso de datos de proceso condicionales (MMDPC), se obtuvieron diferencias estadísticamente significativas con un tamaño de efecto pequeño $p=0.006$ y $\varepsilon^2=0.0330$. Se observó predominancia del nivel ambiguo en todas las estructuras, excepto en la cíclica. Aunque en la estructura cíclica predominó la consistencia, el valor de variabilidad indicó que se puede mover a la ambigüedad 3 [1]. Asimismo, el análisis entre las estructuras condicional y cíclica fue el único que arrojó diferencias significativas, con una magnitud efecto grande.

Finalmente, el análisis del modelo de datos de proceso repetitivos (MMDPR) arrojó los valores $p < 0.001$ y $\varepsilon^2 = 0.1011$, que determinaron diferencias estadísticamente significativas entre los modelos, con una magnitud de efecto moderada. De esta forma, el nivel consistente predominó en la estructura cíclica, con opción de desplazarse al ambiguo 3 [1]; la ambigüedad se observó en las estructuras secuenciales y cíclicas. En cuanto a los anidamientos, predominó el nivel inconsistente, con medida de variabilidad entre ambiguo e inconsistente 3 [1]. En relación con las comparaciones entre las estructuras, se obtuvieron diferencias estadísticas significativas, al comparar la secuencial con la cíclica y la cíclica con la anidada.

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

Discusión

Los principales hallazgos dan cuenta de un nivel de complejidad mayor en las estructuras condicionales, cíclicas y anidadas, respecto a la estructura secuencial. Para el caso del modelo mental datos de proceso (MMDP), los estudiantes encontraron un nivel de dificultad más alto al enfrentarse a situaciones problema que implican repeticiones, y aunque la estructura anidada las contiene, se observó una tendencia por la identificación de los procesos secuenciales y condicionales que igualmente son parte de esta estructura. Asimismo, el alto nivel de ambigüedad presentado en la estructura anidada da cuenta de una comprensión difusa del estudiante sobre lo que se requiere resolver, debido a que se evidencia un planteamiento incoherente con los requerimientos del problema. Estos resultados, encuentran puntos de convergencia con el estudio realizado por Scapin y Mirolo (2020), quienes encontraron que las repeticiones son más exigentes a nivel cognitivo para el rango de edad de los estudiantes entre 15 y 17 años; además, los autores sostienen que la interpretación de las condiciones que regulan un ciclo y la abstracción requerida en las estructuras anidadas, representan un gran reto para ellos. En la misma línea, autores como Cherenkova et al. (2014) y Liénardy et al. (2021) encontraron que los condicionales y los ciclos son complejos para los estudiantes. Igualmente, Alzahrani et al. (2019) determinaron que entre las mayores dificultades que enfrentan los estudiantes están las estructuras repetitivas y las estructuras repetitivas en anidamientos.

Respecto al modelo mental datos de entrada explícitos (MMDEE), se estableció que, en la estructura secuencial, los estudiantes identificaron de manera más coherente los datos de entrada explícitos en el enunciado secuencial, en relación con las estructuras cíclicas y anidadas. El nivel de inconsistencia predominante en el problema condicional indicó una identificación incoherente de los datos de entrada, en cuanto a que el estudiante no logró diferenciar cuáles datos eran explícitos y cuáles estaban implícitos en el texto del problema. En relación con el modelo mental datos de entrada implícitos (MMDEI), los resultados indican que la identificación de los datos que debía ingresar el usuario fue un proceso con tendencia coherente en los enunciados secuenciales, condicionales y anidados, predominando en la estructura condicional. Para la estructura cíclica, se observó que los estudiantes proponen datos incoherentes y de diversa naturaleza, al mezclar datos de entrada con datos de salida. Estos hallazgos relacionados con los datos de entrada (MMDEE y

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

MMDEI), coinciden con Swidan et al. (2018), quienes determinaron que, entre los tres conceptos erróneos más comunes en programación, se encuentra la dificultad para entender la interactividad de un programa cuando se requiere la entrada del usuario.

Respecto a los modelos mentales datos de proceso, que se determinaron de tipo secuencial (MMDPS), condicional (MMDPC) y cíclico (MMDPR), se estableció que, en el primer caso (MMDPS), los estudiantes identificaron con más coherencia este tipo de datos en las estructuras secuencial y en la cíclica, en contraste con la estructura condicional, en la que plantearon procesos aritméticos incorrectos, tanto a nivel lógico como sintáctico. Finalmente, en los anidamientos, se evidenció la tendencia a operar variables desconocidas o que no se habían calculado previamente.

Al observar el análisis correspondiente al modelo mental datos de proceso condicionales (MMDPC), se encontró un alto nivel de ambigüedad, asociado a la identificación de las operaciones aritméticas y lógicas que implican el planteamiento de condiciones. Si bien, este proceso es propio de la estructura condicional, los ejercicios propuestos para las estructuras cíclicas y anidadas contenían situaciones que requerían el uso de condicionales. En el caso de los ejercicios para evaluar las estructuras condicionales y cíclicas, los estudiantes debían operar porcentajes. Sin embargo, la situación problema de tipo condicional, exponía el cálculo del valor total de un refrigerio en la tienda escolar sujeto a un descuento representado en porcentaje y que dependía del número de compañeros invitados, mientras para los ciclos se planteó determinar si N estudiantes aprobaban o no una asignatura, a partir del cálculo de la nota definitiva, que dependía de los porcentajes que tenían tres notas ingresadas por el usuario. De esta forma, se interpreta que la segunda situación hace parte de un contexto más natural para el estudiante, en relación con los cálculos que debe realizar cada periodo académico para conocer las notas finales y determinar si aprueba o no las diferentes asignaturas del plan de estudios y, aunque comprar el refrigerio es una situación muy cotidiana, no es común el cálculo de porcentajes en esta actividad. Lo anterior, contextualiza el nivel de consistencia hallado en la estructura cíclica, comparado con la ambigüedad que predominó en las demás.

En referencia al modelo para la identificación de datos de proceso repetitivos (MMDPR), presentó naturaleza diversa en las estructuras secuenciales y condicionales, debido a que los estudiantes no lograron reconocer claramente que los problemas asociados a estas estructuras no implicaban el uso de iteraciones. Adicionalmente, en el caso de los anidamientos, las elecciones

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

de los estudiantes, en cuanto a los procedimientos repetitivos, resultaron predominantemente inconsistentes.

En consecuencia, los resultados obtenidos en los modelos mentales de proceso para las tres estructuras (MMDPS, MMDPC y MMDPR) dan cuenta de dificultades para expresar la solución del problema en una secuencia de pasos lógicos y coherentes. A propósito, Kwon (2017) y Swidan et al. (2018) también observaron dificultades en la comprensión de la secuencialidad de las instrucciones como uno de los errores más frecuentes en los programadores novatos.

Los resultados obtenidos para el modelo mental diseño de la solución (MMDS) indicaron un nivel de acercamiento mayor a la codificación en pseudocódigo, en la estructura secuencial y de manera difusa o nula en las demás estructuras. La pregunta que indagaba por el MMDS se formuló de tipo abierto con la finalidad de reconocer los planteamientos de codificación que realizaran los estudiantes. Sin embargo, en su mayoría, se observaron soluciones que combinaban sintaxis del intérprete de código PseInt y de Java; este hábito también fue identificado por Fujiwara et al. (2012), quien lo describió como una obsesión por conocer otros programas, en la que se evidencia la tendencia de los estudiantes por el uso de las características y la sintaxis de lenguajes de programación que ya conocen. Adicionalmente, se identificaron errores en los procesos de asignación y en el uso de variables, pues se observó la tendencia a realizar operaciones aritméticas sin asignación a ninguna variable.

En relación con los condicionales, fue muy frecuente el uso del operador de asignación (=) en lugar de (= =) para comparar igualdad; confusión que, de igual forma, fue ubicada entre las más frecuentes por Altadmri y Brown (2015), quienes la clasificaron como un error de sintaxis y de semántica. Asimismo, algunos estudiantes realizaron la comparación en el “Sino”, y otros omitieron el cierre de la estructura. En la estructura cíclica se destacó el planteamiento mínimo de procesos de tipo secuencial y condicional; la mayoría de los estudiantes no tuvieron en cuenta la cantidad de iteraciones que se debían realizar los procesos. Igualmente, varios estudiantes, que intentaron codificar el problema correspondiente a esta estructura, tuvieron dificultades para escribir en forma correcta los parámetros del control del ciclo y para ubicar la instrucción de cierre de esta estructura, incluso fue común la omisión del cierre del ciclo.

En los anidamientos no fue clara la categoría de anidamiento, debido a que no se declaró la instrucción *Fin si* en muchos casos y la instrucción *Sino* se omitió o se incluyó sin respetar la

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

tabulación lógica a la que correspondía. Lo anterior contrasta los hallazgos de Alzahrani et al. (2019), quienes enumeraron entre los errores comunes que generaron problemas en la programación, los relacionados con el uso de "Sino", en lugar de múltiples "Si".

Para el modelo mental de verificación de la solución (MMV), todas las estructuras presentaron los niveles más altos en la ambigüedad y, en segundo lugar, se tienen los puntajes más altos en la no realización de este proceso, predominando en la estructura anidada. De otro lado, el nivel de consistencia mayor se presentó en la estructura secuencial. Los resultados obtenidos para el MMV son coherentes con el MMDS, debido a que este modelo consiste en la comprobación del pseudocódigo. Lo anterior, evidencia dificultades asociadas a la traducción de enunciados del lenguaje natural al pseudocódigo y a la reversibilidad del pensamiento, implicada en la verificación de la solución. Asimismo, se ratifica la poca comprensión de la secuencia de procesos o instrucciones que se deben plantear para solucionar un problema.

Los hallazgos expuestos sustentan el aporte a una dimensión poco explorada en el contexto de la programación, debido a que el abordaje de la resolución de algoritmos en pseudocódigo ha sido preocupación de escasos estudios de acuerdo con la revisión de literatura; por tanto, este estudio contribuye con la consolidación del campo teórico del fenómeno en cuestión. En el mismo sentido, la investigación representa un punto de partida para nuevos estudios que se interesen por replicar en otros contextos, los hallazgos presentados.

Adicionalmente, dada la complejidad del fenómeno, la investigación que dio origen a este artículo deja expuesto que es necesario desarrollar investigaciones que profundicen de manera específica en cada una de las estructuras de programación, así como fortalecer el método de resolución de problemas algorítmicos desde las fases metodológicas, apoyadas en los planteamientos de Polya (2004). Por otro lado, estos aspectos no se inscriben solamente en un dominio específico de conocimiento, de allí que autores como Weng et al. (2022) plantean que en la actualidad todas las personas deben estar capacitadas para la resolución de problemas y la programación; asimismo, Peel et al. (2022) indican que la informática se ha convertido en un elemento esencial para la resolución de problemas. Por tanto, los aportes de este estudio se pueden extender a otras áreas del conocimiento.

Del mismo modo, el hecho de encontrar una relación entre los modelos mentales y las estructuras algorítmicas genera amplias posibilidades para reorientar las estrategias de enseñanza

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

y aprendizaje en la asignatura Lógica de Programación y reestructurar el ciclo de formación en media técnica en informática, de tal manera que los aprendizajes resulten significativos para los estudiantes y puedan tener aplicabilidad en la vida cotidiana. Estas implicaciones del dominio práctico, además de proporcionar herramientas teóricas y didácticas a los docentes que orientan las áreas específicas, son replicables a nivel universitario, en los primeros semestres de la ingeniería de sistemas, donde se han encontrado problemáticas similares en relación con las dificultades asociadas al aprendizaje de la programación.

Conclusiones

Para el ejercicio investigativo que orientó el desarrollo del estudio fuente de este artículo se construyó una prueba de identificación de modelos mentales que pretendió develar la relación entre las estructuras algorítmicas de programación y los modelos mentales que construyen los estudiantes durante su resolución; la metodología abordada, desde el paradigma empírico analítico, permitió explicar el fenómeno a la luz de tres niveles de consistencia para los modelos implicados en las fases de resolución de problemas en este ámbito, que a su vez determinaron niveles de complejidad asociados a las estructuras de programación.

En este sentido, los resultados expuestos permitieron establecer una relación significativa entre los niveles de consistencia de los modelos mentales y la tipología de las estructuras algorítmicas de programación. Es así como se evidenció una tendencia de modelos mentales consistentes para la estructura secuencial, en contraposición a la inconsistencia y ambigüedad observada en las estructuras condicionales y cíclicas, con predominancia en los anidamientos. Igualmente, se determinó que el diseño y la verificación de la solución, representaron los procesos con mayor dificultad para los estudiantes. En esta misma línea, se pudo establecer que, en el marco de las fases propuestas para la resolución de problemas matemáticos, la mayor complejidad se ubicó en la fase de configuración del plan solución, situación que no permite continuar con el abordaje de la ejecución del plan y la verificación de la solución, y que remite a debilidades en la comprensión del problema.

De lo anterior, se concluye que la enseñanza temprana de un lenguaje de programación o de un intérprete de pseudocódigo, ocasiona que los estudiantes elaboren modelos mentales difusos

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

de diversa naturaleza sin una secuenciación lógica. Asimismo, se evidenció la necesidad de intencionar estrategias didácticas que tengan en cuenta los modelos mentales de los estudiantes, en el marco de una metodología de resolución de problemas; con la finalidad de desarrollar habilidades en la comprensión de los enunciados y en el planteamiento de la solución en pseudocódigo, como momento previo a la programación en computador.

Referencias

Aggarwal, A., Touretsky, D., & Gardner-McCune, C. (2018). *Demonstrating the Ability of Elementary School Students to Reason About Programs* [Demostración de la capacidad de los estudiantes de primaria para razonar sobre los programas] [Sesión de conferencia]. https://www.cs.cmu.edu/~dst/pubs/Aggarwal-SIGCSE-2018.pdf

Altadmri, A., & Brown, N. C. (2015). 37 Million Compilations: Investigating Novice Programming Mistakes in Large-Scale Student Data [37 millones de compilaciones: Investigación de errores de programación de novatos en datos de estudiantes a gran escala]. *SIGCSE '15: The 46th SIGCSE technical symposium on computer science education*. https://doi.org/10.1145/2676723.2677258

Álvarez, V., Torres, T., Gangoso, Z., & Sanjosé, V. (2020). A Cognitive Model to Analyse Physics and Chemistry Problem-Solving Skills: Mental Representations Implied in Solving Actions [Un modelo cognitivo para analizar las habilidades de resolución de problemas de física y química: representaciones mentales implícitas en la resolución de acciones]. *Journal of Baltic Science Education, 19*(5), 730-746. https://doi.org/10.33225/jbse/20.19.730

Alzahrani, N., Vahid, F., & Edgcomb, A. D. (2019). *Manual Analysis of Homework Coding Errors for Improved Teaching and Help* [Análisis del manual de errores de codificación de tareas para mejorar la enseñanza y la ayuda] [Sesión de conferencia]. ASEE Annual Conference & Exposition. Tampa, Florida. https://doi.org/10.18260/1-2--33083

Andrzejewska, M., & Skawińska, A. (2020). Examining Students' Intrinsic Cognitive Load During Program Comprehension – An Eye Tracking Approach [Examen de la carga cognitiva intrínseca de los estudiantes durante la comprensión del programa: un enfoque de seguimiento ocular]. In I. Bittencourt, M. Cukurova, K. Muldner, R. Luckin & E. Millán (Eds.), *Artificial*

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

Intelligence in Education. Lecture Notes in Computer Science (pp. 25-30). Springer. https://doi.org/10.1007/978-3-030-52240-7_5

Barland, I., Felleisen, M., Fisler, K., Kolaitis, P., & Vardi, M. (2009). *Integrating Logic into the Computer Science Curriculum* [Integración de la lógica en el plan de estudios de ciencias de la computación]. http://www.cs.utexas.edu/users/csed/formal-methods/docs/iticse-fislervardi.pdf

Bayman, P., & Mayer, R. (1983). A diagnosis of beginning programmers' misconceptions of BASIC programming statements [Un diagnóstico de los conceptos erróneos de los programadores principiantes sobre las sentencias de programación BASIC]. *Communications of the ACM, 26*(9), 677-679. http://dx.doi.org/10.1145/358172.358408

Bubica, N., & Boljat, I. (2015). *Programming novices'mental models* [Programación de modelos mentales de principiantes] [Sesión de conferencia]. 7th International Conference on Education and New Learning Technologies, Barcelona, Spain. https://doi.org/10.13140/RG.2.1.3773.2960

Burkholder, E., Price, A., Flynn, M., & Wieman, C. (2020). Assessing problem-solving in science and engineering programs [Evaluación de la resolución de problemas en programas de ciencia e ingeniería]. In C. Wolf & Bennett (Eds.), *2019 PERC Proceedings* (pp. 75-80). American Association of Physics Teachers https://doi.org/10.1119/perc.2019.pr.burkholder

Cairó, O. (2005). *Metodología de la Programación*. Alfaomega Grupo Editor, S.A.

Chalpartar Nasner, L. T. M., Fernández Guzmán, A. M., Betancourth Zambrano, S., & Gómez Delgado, Y. A. (2022, mayo-agosto). Deserción en la población estudiantil universitaria durante la pandemia, una mirada cualitativa. *Revista Virtual Universidad Católica Del Norte*, (66), 37-62. https://doi.org/10.35575/rvucn.n66a3

Chang, Y.-H., Yan, Y.-C., & Lu, Y.-T. (2022). Effects of Combining Different Collaborative Learning Strategies with Problem-Based Learning in a Flipped Classroom on Program Language Learning [Efectos de la combinación de diferentes estrategias de aprendizaje colaborativo con el aprendizaje basado en problemas en un aula invertida en el aprendizaje del lenguaje de programación]. *Sustainability*, *14*(9), Article 5282. https://doi.org/10.3390/su14095282

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

Chaves Torres, A. (2017). *Aprenda a diseñar algoritmos*. Universidad Nacional Abierta y a Distancia. https://doi.org/10.22490/9789586516228

Cheah, C. S. (2020). Factors contributing to the difficulties in teaching and learning of computer programming: A literature review [Factores que contribuyen a las dificultades en la enseñanza y el aprendizaje de la programación informática: Una revisión de la literatura]. *Contemporary Educational Technology*, *12*(2), Article ep272. https://doi.org/10.30935/cedtech/8247

Cherenkova, Y., Zingaro, D., & Petersen, A. (2014). Identifying challenging CS1 concepts in a large problem dataset [Identificación de conceptos difíciles de CS1 en un gran conjunto de datos de problemas]. In *Proceedings of the 45thACM technical symposium on computer science education* (pp. 695-700). https://doi.org/10.1145/2538862.2538966

Congreso de la República de Colombia. (1994, 8 de febrero). *Ley 115*, Ley General de Educación. https://www.mineducacion.gov.co/1621/articles-85906_archivo_pdf.pdf

Congreso de la República de Colombia. (2002, 19 de julio). *Ley 749*, por la cual se organiza el servicio público de la educación superior en las modalidades de formación técnica profesional y tecnológica, y se dictan otras disposiciones. https://www.mineducacion.gov.co/1621/articles-86432_Archivo_pdf.pdf

Derman, A., Koçak, N., & Eilks, I. (2019). Insights into Components of Prospective Science. Teachers' Mental Models and Their Preferred Visual Representations of Atoms [Perspectivas sobre los componentes de la ciencia prospectiva. Modelos mentales de los profesores y sus representaciones visuales preferidas de los átomos]. *Education Sciences*, *9*(2), Article 154. https://doi.org/10.3390/educsci9020154

Desoete, A., Roeyers, H., & De Clercq, A. (2003). Can offline metacognition enhance mathematical problem solving? [¿Puede la metacognición fuera de línea mejorar la resolución de problemas matemáticos?]. *Journal of Educational Psychology*, *95*(1), 188-200. https://doi.org/10.1037/0022-0663.95.1.188

Díaz, D., Velásquez Sánchez, M. I., Rincón Barreto, D. M., Blanco Belén, O. A., & Correa López, R. A. (2022, enero-abril). Relación entre rasgos de personalidad, toma de decisiones y la permanencia académica. *Revista Virtual Universidad Católica Del Norte*, (65), 263-283. https://doi.org/10.35575/rvucn.n65a10

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

Dominguez-Lara, S. A. (2017, marzo-abril). Magnitud del efecto en comparaciones entre 2 o más grupos. *Revista de Calidad Asistencial*, *32*(2), 121-122. https://doi.org/10.1016/j.cali.2016.04.002

Dominguez-Lara, S. A. (2018, julio-agosto). Magnitud del efecto, una guía rápida. *Educación Médica*, *19*(4), 251-254. https://doi.org/10.1016/j.edumed.2017.07.002

Enes da Silveira, P., & Gomes, R. (2019). How to achieve better performance in teaching computer programming: Cases of iterative and recursive programming [Cómo lograr un mejor desempeño en la enseñanza de la programación de computadoras: Casos de programación iterativa y recursiva]. *AIP Conference Proceedings*, *2116*(1), Article 410003. https://doi.org/10.1063/1.5114427

Fujiwara, K., Fushida, K., Tamada, H., Igaki, H., & Yoshida, N. (2012). *Why Novice Programmers Fall into a Pitfall?: Coding Pattern Analysis in Programming Exercise* [¿Por qué los programadores novatos caen en una trampa?: Análisis de patrones de codificación en el ejercicio de programación] [Sesión de conferencia]. Proceedings of the 2012 Fourth International Workshop on Empirical Software Engineering in Practice, Massachusetts Ave., NW Washington, DC, United States. https://doi.org/10.1109/IWESEP.2012.13

George, D., & Mallery, P. (2003). *SPSS for Windows step by step: A simple guide and reference. 11.0 update* [SPSS para Windows paso a paso: una guía y referencia sencillas. Actualización 11.0] (4ª Ed.). Allyn & Bacon.

Goldson, D., Reeves, S., & Bornat, R. (1993). A review of several programs for the teaching of logic [Una revisión de varios programas para la enseñanza de la lógica]. The *Computer Journal*, *36*(4), 373–386. https://doi.org/10.1093/comjnl/36.4.373

Goltermann, R., & Höppner, F. (2017). Internalizing a Viable Mental Model of Program Execution in First Year Programming Courses [Internalización de un modelo mental viable de ejecución de programas en cursos de programación de primer año]. *Revista Brasileira de Psiquiatria.* http://ceur-ws.org/Vol-2015/ABP2017_paper_05.pdf

Hernández, M. (2010). *Diseño estructurado de algoritmos, diagramas de flujos y pseudocódigos*. Universidad de Teuxtepe.

Inhelder, B., & Piaget, J. (1973). *De la lógica del niño a la lógica del adolescente*. Paidós.

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

Insuasti, J. (2016). Problemas de enseñanza y aprendizaje de los fundamentos de programación. *Revista Educación y Desarrollo Social*, *10*(2), 234-246. https://doi.org/10/18359/reds.170

Izu, C., Schulte, C., Aggarwal, A., Cutts, Q., Duran, R., Gutica, M., Heinemann, B., Kraemer, E., Lonati, V., Mirolo, C., & Weeda, R. (2019). *Fostering program comprehension in novice programmers - learning activities and learning trajectories* [Fomento de la comprensión del programa en programadores novatos: actividades de aprendizaje y trayectorias de aprendizaje] [Sesión de conferencia]. Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE ’19), Aberdeen, Scotland. https://doi.org/10.1145/3344429.3372501

Johnson-Laird, P. (1983). *Mental models: Towards a cognitive science of language, inference, and consciousness* [Modelos mentales: hacia una ciencia cognitiva del lenguaje, la inferencia y la conciencia]. Harvard University Press.

Johnson-Laird, P. (2010). Mental models and human reasoning [Modelos mentales y razonamiento humano]. *Proceedings of the National Academy of Sciences, 107*(43), 18243-18250. https://doi.org/10.1073/pnas.1012933107

Johnson-Laird, P., Byrne, R. M., & Schaeken, W. (1992). Propositional reasoning by model. *Psychological Review, 99*(3), 418-439. https://doi.org/10.1037/0033-295X.99.3.418

Joyanes Aguilar, L. (2008). *Fundamentos de programación: Algoritmos, estructura de datos y objetos*. McGraw-Hill.

Katz, M. H. (2006). *Multivariable analysis* [Análisis multivariable] (2ª ed.). Cambridge University Press.

Khemlani, S., Mackiewicz, R., Bucciarelli, M., & Johnson-Laird, P. (2013). Kinematic mental simulations in abduction and deduction [Simulaciones mentales cinemáticas en abducción y deducción]. *Proceedings of the National Academy of Sciences, 110*(42), 16766-16771. https://doi.org/10.1073/pnas.1316275110

Khemlani, S., & Johnson-Laird, P. (2017). Illusions in Reasoning [Ilusiones en el razonamiento]. *Minds & Machines, 27*, 11-35. https://doi.org/10.1007/s11023-017-9421

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

Khemlani, S., & Johnson-Laird, P. (2022). Reasoning About Properties: A Computational Theory [Razonamiento sobre propiedades: una teoría computacional]. *Psychological Review*, *129*(2), 289-312. https://doi.org/10.1037/rev0000240

Kirçali, A.Ç., & Özdener, N. (2022). Comparison of Plugged and Unplugged Tools in Teaching Algorithms at the K-12 Level for Computational Thinking Skills [Comparación de herramientas conectadas y desconectadas en la enseñanza de algoritmos en el nivel K-12 para habilidades de pensamiento computacional]. *Technology, Knowledge and Learning*. https://doi.org/10.1007/s10758-021-09585-4

Kwon, K. (2017). Novice programmer´s misconception of programming reflected on problem-solving plans [El concepto erróneo del programador novato sobre la programación reflejado en los planes de resolución de problemas]. *International Journal of Computer Science Education in Schools*, *1*(4), 14-24. https://doi.org/10.21585/ijcses.v1i4.19

Liénardy, S., Donnet, B., & Leduc, L. (2021). Promoting Engagement in a CS1 Course with Assessment for Learning [Promoción de la participación en un curso CS1 con evaluación para el aprendizaje]. *Student Success*, *12*(1), 102-111. https://doi.org/10.5204/ssj.1668

Lonati, V., & Morpurgo, A. (2021). *Fostering Strategic Knowledge and Program Comprehension Skills in Students Struggling with CS1* [Fomentar el conocimiento estratégico y las habilidades de comprensión del programa en estudiantes que luchan con CS1] [Sesión de conferencia]. SIGCSE '21: The 52nd ACM Technical Symposium on Computer Science Education, United States. https://doi.org/10.1145/3408877.3439632

Ma, N., Qian, J., & Gong, K. (2023). Promoting programming education of novice programmers in elementary schools: A contrasting cases approach for learning programming [Fomento de la enseñanza de la programación a programadores novatos en la escuela primaria: Un enfoque de casos contrastados para el aprendizaje de la programación]. *Education and Information Technologies*. https://doi.org/10.1007/s10639-022-11565-9

Muñoz-Campos, V., Franco-Mariscal, A. J., & Blanco-Lopez, A. (2018). Modelos mentales de estudiantes de educación secundaria sobre la transformación de la leche en yogur. *Revista Eureka sobre Enseñanza y Divulgación de las Ciencias*, *15*(2), Artículo 2106. http://dx.doi.org/10.25267/Rev_Eureka_ensen_divulg_cienc.2018.v15.i2.21067

Oviedo, E. (2005). *Lógica de Programación*. Ecoe Ediciones.

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

Peel, A., Sadler, T. D., & Friedrichsen, P. (2022). Algorithmic Explanations: an Unplugged Instructional Approach to Integrate Science and Computational Thinking [Explicaciones algorítmicas: un enfoque didáctico para integrar la ciencia y el pensamiento computacional]. *Journal of Science Education and Technology*, *31*(4), 428-441. https://doi.org/10.1007/s10956-022-09965-0

Polya, G. (2004). *How to solve it: a new aspect of mathematical method* [Cómo resolverlo: un nuevo aspecto del método matemático]. Princeton University Press.

Praisri, A., & Faikhamta, C. (2020). Enhancing Students' Mental Models of Chemical. Equilibrium Through Argumentation within Model-based Learning [Mejorar los modelos mentales de los estudiantes de química. Equilibrio a través de la argumentación dentro Aprendizaje basado en modelos]. *International Journal of Learning, Teaching and Educational Research*, *19*(7), 121-142. https://doi.org/10.26803/ijlter.19.7.7

Prayekti, N., Nusantara, T., Sudirman, Susanto, H., & Rofiki, I. (2020). Students' mental models in mathematics problem-solving [Modelos mentales de los estudiantes en la resolución de problemas matemáticos]. *Journal of Critical Reviews, 7*(12), 468-470. https://dx.doi.org/10.31838/jcr.07.12.83

Rapp, D. N. (2005). Mental Models: Theoretical Issues for Visualizations in Science Education [Modelos mentales: aspectos teóricos de las visualizaciones en la enseñanza de las ciencias]. In J. K. Gilbert (Ed.), *Models and Modeling in Science Education* (Vol. 1 pp. 43-60). Springer. https://doi.org/10.1007/1-4020-3613-2_4

Sambe, G., Dramé, K., & Basse, A. (2021). Towards a Framework to Scaffold Problem-solving Skills in Learning Computer Programming [Hacia un marco de andamiaje para las habilidades de resolución de problemas en el aprendizaje de la programación informática]. *Proceedings of the 13th International Conference on Computer Supported Education, 1*, 323-330. https://doi.org/10.5220/0010446503230330

Saxena, A., Lo, C. K., Hew, K. F., & Wong, G. K. W. (2020). Designing unplugged and plugged activities to cultivate computational thinking: An exploratory study in early childhood education [Diseño de actividades conectadas y desconectadas para cultivar el pensamiento computacional: Un estudio exploratorio en educación infantil]. *Asia-Pacific Education Researcher*, *29*(1), 55-66. https://doi.org/10.1007/s40299-019-00478-w

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

Scapin, E., & Mirolo, C. (2020). *An Exploratory Study of Students' Mastery of Iteration in the High School* [Un estudio exploratorio sobre el dominio de la iteración por parte de los estudiantes en la escuela secundaria] [Sesión de conferencia]. Proceedings of the International Conference on Informatics in School: Situation, Evaluation and Perspectives. Tallinn, Estonia. https://ceur-ws.org/Vol-2755/paper4.pdf

Slapničar, M., Tompa, V., Glažar, S. A., & Devetak, I. (2018). Fourteen-year-old students' misconceptions regarding the sub-micro and symbolic levels of specific chemical concepts [Conceptos erróneos de estudiantes de catorce años sobre los niveles submicro y simbólico de conceptos químicos específicos]. *Journal of Baltic Science Education*, *17*(4), 620-632. https://doi.org/10.33225/JBSE/18.17.620

Solomon, D. H., Finkelstein, J. S., Polinski, J. M., Arnold, M., Licari, A., Cabral, D., Canning, C., Avorn, J., & Katz, J. N. (2006). A randomized controlled trial of mailed osteoporosis education to older adults [Un ensayo controlado aleatorizado de educación sobre osteoporosis enviada por correo a adultos mayores]. *Osteoporosis International*, *17*(5), 760-767. https://doi.org/10.1007/s00198-005-0049-y

Spangsberg, T., Fincher, S., & Dziallas, S. (2018). *Non-Traditional Novices' Perceptions of Learning to Program: A Framework of Developing Mental Models* [Percepciones de los novatos no tradicionales sobre el aprendizaje de la programación: un marco para el desarrollo de modelos mentales] [Sesión de conferencia]. Conference: 2018 IEEE Frontiers in Education Conference (FIE), San José, Estados Unidos. https://doi.org/10.1109/FIE.2018.8659301

Sukirman, Dias Aziz, P., Aziz, A., & Utaminingsih. (2022). Block-Based Visual Programming as a Tool for Learning the Concepts of Programming for Novices [La programación visual basada en bloques como herramienta para el aprendizaje de los conceptos de programación para novatos]. *International Journal of Information and Education Technology*, *12*(5), 365-371. https://doi:10.18178/ijiet.2022.12.5.1628

Supriyadi, M., Waremra, R. S., Hidayati, R. N., Masyur, J., Novia, M., & Kusrini, M. (2018). Students' Mental-Modeling Ability (MMA) in Concept Understanding of Vector [Capacidad de modelado mental (MMA) de los estudiantes en la comprensión conceptual del vector]. *Proceedings of the 1st International Conference on Social Sciences (ICSS 2018)*, 1435-1438. https://doi.org/10.2991/icss-18.2018.302

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

Swidan, A., Hermans, F., & Marileen, S. (2018). *Programming Misconceptions for School Students* [Sesión de conferencia]. ICER '18: Proceedings of the 2018 ACM Conference on International Computing Education Research. https://doi.org/10.1145/3230977.3230995

Teichert, M., Tien, L., Dysleski, L., & Rickey, D. (2017). Thinking Processes Associated with Undergraduate Chemistry Students' Success at Applying a Molecular-Level Model in a New Context [Procesos de pensamiento asociados con el éxito de los estudiantes de química de pregrado en la aplicación de un modelo de nivel molecular en un nuevo contexto]. *Journal of Chemical Education*, *94*(9), 1195-1208. https://doi.org/10.1021/acs.jchemed.6b00762

Tomczak, M., & Tomczak, E. (2014). The need to report effect size estimates revisited. An overview of some recommended measures of effect size [Revisión de la necesidad de informar las estimaciones del tamaño del efecto. Una descripción general de algunas medidas recomendadas del tamaño del efecto]. *Trends in Sport Sciences*, *1*(21), 19-25. https://www.researchgate.net/publication/303919832_The_need_to_report_effect_size_estimates_revisited_An_overview_of_some_recommended_measures_of_effect_size

Twigg, S., Lynne, B., & Emil, W. (2019). *Using children's literature to introduce computing principles and concepts in primary schools: work in progress* [Utilización de la literatura infantil para introducir principios y conceptos informáticos en la escuela primaria: trabajo en curso]. Proceedings of the 14th Workshop in Primary and Secondary Computing Education. https://doi.org/10.1145/3361721.3362116

University of Cambridge. (2021). *Rules of thumb on magnitudes of effect sizes* [Reglas generales sobre las magnitudes de los tamaños del efecto]. https://imaging.mrc-cbu.cam.ac.uk/statswiki/FAQ/effectSize

Vrachnos, E., & Jimoyiannis, A. (2017). Secondary education students' difficulties in algorithmic problems with arrays: An analysis using the SOLO taxonomy [Dificultades de estudiantes de educación secundaria en problemas algorítmicos con arreglos: un análisis utilizando la taxonomía SOLO]. *Themes in Science & Technology Education*, *10*(1), 31-52. https://api.semanticscholar.org/CorpusID:59463886

Weng, C., Matere, I.M., Hsia, C.-H., Wang, M.-Y., & Weng, A. (2022). Effects of LEGO robotic on freshmen students' computational thinking and programming learning attitudes in Taiwan [Efectos de la robótica LEGO en el pensamiento computacional y las actitudes de aprendizaje

Modelos mentales y algoritmos | Revista Virtual Universidad Católica del Norte, **69**, 98-134
ISSN: 0124-5821 (En línea)

de la programación de los estudiantes de primer año en Taiwán]. *Library Hi Tech*, *40*(4), 947-962. https://doi.org/10.1108/LHT-01-2021-0027

Zúñiga, R., Hurtado, J., & Paderewsky, P. (2016). Discovering the mechanisms of abstraction in the performance of work teams in children to solve computational problems [Descubriendo los mecanismos de abstracción en el desempeño de equipos de trabajo en niños para resolver problemas computacionales]. *Sistemas & Telemática*, *14*(36), 69-87. https://www.redalyc.org/articulo.oa?id=411545767002