



Fundamentos de programación: una aproximación al desarrollo del pensamiento

Programming Fundamentals: An Approach to the Development of Thinking

Fundamentos de programação: uma abordagem ao desenvolvimento do pensamento

Jhon Eduardo Villacrés-Sampedro ^I
jhon.villacres@epoch.edu.ec
<https://orcid.org/0000-0002-8064-9680>

Sonia Patricia Cordovez-Machado ^{II}
sonia.cordovez@epoch.edu.ec
<https://orcid.org/0000-0002-2393-7918>

Nancy Maritza Montoya-Ramírez ^{III}
nmontoya@istmas.edu.ec
<https://orcid.org/0000-0001-6439-9477>

Fabián Enrique Calvopiña-Estrella ^{IV}
fabian.calvopina@istcarloscisneros.edu.ec
<https://orcid.org/0009-0005-3330-4582>

Correspondencia: jhon.villacres@epoch.edu.ec

Ciencias de la Computación
Artículo de Investigación

* **Recibido:** 06 de junio de 2024 * **Aceptado:** 11 de julio de 2024 * **Publicado:** 22 de agosto de 2024

- I. Magíster en Desarrollo de la Inteligencia y Educación, Máster Universitario en Tecnología Educativa y Competencias Digitales, Ingeniero en Sistemas, Docente de la Facultad de Informática y Electrónica, Software, Tecnologías de la Información, Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador.
- II. Máster Universitario en Ingeniería de Software y Sistemas Informáticos, Ingeniero en Sistemas, Docente de la Facultad de Informática y Electrónica, Software, Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador.
- III. Magíster en Informática Educativa, Ingeniera en Sistemas Informáticos, Instituto Superior Tecnológico Dr. Misael Acosta Solís, Riobamba, Ecuador.
- IV. Máster en Ciberseguridad y Privacidad, Ingeniero en informática, Docente Carrera Diseño y Mantenimiento de Redes, Instituto Superior Universitario Carlos Cisneros, Riobamba, Ecuador.

Resumen

El presente trabajo de investigación, explora la relación existente entre los fundamentos de programación y el desarrollo del pensamiento, haciendo énfasis en cómo la programación puede transformarse en una herramienta poderosa para fortalecer diversas habilidades cognitivas, enmarcadas en los conceptos del desarrollo del pensamiento. Se ha realizado un estudio bibliográfico, desde la identificación de autores hasta un análisis exhaustivo de la literatura, se tomó como conceptos clave de la programación, los algoritmos, las estructuras de control, las estructuras de datos, y la descomposición de problemas, por otra parte, contribuyeron a los conceptos de desarrollo del pensamiento, el pensamiento computacional, el pensamiento lógico, la resolución de problemas, el pensamiento crítico, la atención y el pensamiento algorítmico. Los hallazgos en la investigación, demuestran que la práctica de la programación no solo mejora las competencias técnicas, sino que también potencia la capacidad de los individuos para enfrentar y resolver problemas complejos de manera efectiva, lógica y coherente. Así mismo, se encuentra, que el desarrollo del pensamiento es un constructo social, influenciado por el entorno cultural y las interacciones sociales. Finalmente, el artículo concluye que los fundamentos de programación ofrecen una aproximación valiosa al desarrollo del pensamiento, sugiriendo su integración en el currículo educativo, para ampliar las opciones cognitivas de los individuos en un mundo tecnológicamente avanzado.

Palabras clave: Programación; Desarrollo del pensamiento; Pensamiento lógico; Lógica de programación; Algoritmo; Pensamiento crítico.

Abstract

This research explores the relationship between programming fundamentals and the development of thinking, emphasizing how programming can become a powerful tool to strengthen various cognitive skills, framed in the concepts of thinking development. A bibliographic study has been carried out, from the identification of authors to an exhaustive analysis of the literature, taking as key concepts of programming, algorithms, control structures, data structures, and problem decomposition, on the other hand, contributed to the concepts of thinking development, computational thinking, logical thinking, problem solving, critical thinking, attention and algorithmic thinking. The findings in the research demonstrate that the practice of programming not only improves technical skills, but also enhances the ability of individuals to face and solve

complex problems in an effective, logical and coherent manner. Likewise, it is found that thinking development is a social construct, influenced by the cultural environment and social interactions. Finally, the article concludes that programming fundamentals offer a valuable approach to the development of thought, suggesting its integration into the educational curriculum, to expand the cognitive options of individuals in a technologically advanced world.

Keywords: Programming; Development of thought; Logical thinking; Programming logic; Algorithm; Critical thinking.

Resumo

O presente trabalho de investigação explora a relação entre os fundamentos da programação e o desenvolvimento do pensamento, enfatizando como a programação se pode tornar uma ferramenta poderosa para fortalecer diversas competências cognitivas, enquadradas nos conceitos de desenvolvimento do pensamento. Foi realizado um estudo bibliográfico, desde a identificação dos autores até uma análise exaustiva da literatura, tomando como conceitos-chave de programação, algoritmos, estruturas de controlo, estruturas de dados e decomposição de problemas, por outro lado, contribuíram para os conceitos de desenvolvimento do pensamento, pensamento computacional, pensamento lógico, resolução de problemas, pensamento crítico, atenção e pensamento algorítmico. Os resultados da investigação demonstram que a prática da programação não só melhora as competências técnicas, como também aumenta a capacidade dos indivíduos para enfrentar e resolver problemas complexos de uma forma eficaz, lógica e coerente. Da mesma forma, verifica-se que o desenvolvimento do pensamento é uma construção social, influenciada pelo ambiente cultural e pelas interações sociais. Por fim, o artigo conclui que os fundamentos da programação oferecem uma abordagem valiosa para o desenvolvimento do pensamento, sugerindo a sua integração no currículo educacional, para expandir as opções cognitivas dos indivíduos num mundo tecnologicamente avançado.

Palavras-chave: Programação; Desenvolvimento do pensamento; Pensamento lógico; Lógica de programação; Algoritmo; Pensamento crítico.

Introducción

Fundamentos de programación

La computación en sus inicios, no fue tan sofisticada como en los tiempos actuales, se limitaba a intentar calcular y automatizar algunos procesos, un ejemplo clásico de lo que se está afirmando es el ábaco, definido por Pomeroy, A. (2007) como una herramienta de cálculo manual, que consiste en cuentas móviles insertadas en varillas, mismas que permiten realizar operaciones aritméticas básicas, aunque se sabe que el ábaco es una herramienta manual, se puede elaborar un algoritmo con las instrucciones necesarias para realizar operaciones matemáticas. Charles Babbage, en el año 1822, propone la máquina de diferencias, para el cálculo de ecuaciones polinómicas, el mismo autor en el año de 1830, propone la máquina analítica, con conceptos más revolucionarios, al ser una computadora mecánica programable, que ya consideraba componentes como memoria y unidad de control, sin embargo, no se concretó su construcción. Con el paso del tiempo otros científicos ponían de manifiesto sus investigaciones y diseños, entre ellos, las tarjetas perforadas, que sirvieron para la tabulación de datos en el censo de 1890 en Estados Unidos. El avance continúa, dando lugar a la aparición de lenguajes de bajo nivel, que según Stallings (2021) y Patterson & Hennessy (2021), están muy cercanos al lenguaje de máquina, es decir al lenguaje nativo de la computadora, por otra parte, surgen los lenguajes de alto nivel, estos por su flexibilidad, son más fáciles de entender y proporcionan muchas facilidades, para que el programador se centre en el problema a resolver y no en los detalles técnicos de la gestión del hardware (Lutz, 2023; Bloch, 2018; Stroustrup, 2013).

Algoritmo

El computador visto desde el punto de vista del hardware, es un dispositivo que ejecuta uno a uno todos los pasos que un programa indica. Este conjunto de pasos o instrucciones finitas que se llevan a cabo para resolver un problema específico o realizar una tarea en particular, es conocido con el nombre de algoritmo (Cormen et al., 2022; Heineman et al., 2016), para clarificar el concepto citamos a Knuth (2011) y Aho et al. (2020), quienes afirman que un algoritmo es un conjunto de reglas bien definidas que conllevan a la resolución de un problema en un conjunto finito de pasos, el mismo que recibe valores de entrada y entrega algún valor o valores de salida.

Para los autores del presente trabajo de investigación, se puede definir un algoritmo, como un conjunto finito de instrucciones lógicas, secuenciales y ordenadas que conllevan a la resolución de un problema, la consecución de un objetivo o la realización de una tarea en particular.

Estructuras de Control

Para que un algoritmo tenga un diseño correcto, el desarrollador, debe poder controlar el flujo del programa, esto se logra con el uso de las estructuras de control, para Tanenbaum (2016), “las estructuras de control son bloques fundamentales en la programación que alteran el flujo de ejecución de un programa. Estas incluyen las estructuras de selección (if-else), iteración (for, while) y secuencia, que permiten a los programas tomar decisiones y repetir acciones” (pp. 114-116). En el mismo contexto, Lafore (2017) manifiesta que, las estructuras de control son mecanismos que permiten dirigir el flujo de ejecución del programa orientándolo hacia la lógica deseada, estas se clasifican en estructuras de selección (if, switch), repetición (for, while, do-while) y secuencia, que facilitan el control del flujo de instrucciones de manera eficiente (pp. 112-115).”

Como se ha visto, las estructuras de control le confieren al programador la posibilidad de controlar el flujo de ejecución del programa, de tal forma que la información se vaya transformando a lo largo del mismo, de acuerdo a lo que la lógica requiere para, proveer entradas al programa y que este proporcione las salidas deseadas por el usuario final, los algoritmos por su característica textual, no evidencian con facilidad la forma como cambia el flujo de los datos, sin embargo los diagramas de flujo, tienen la capacidad de mostrar como la información se va transformado, hasta llegar a un resultado esperado.

Estructuras de Datos

La información en el computador puede guardarse de forma temporal mientras se ejecuta el programa, o permanente, que se almacena en algún dispositivo de memoria secundaria, y tiene la capacidad de ser recuperado después de que se ha terminado la ejecución del mismo, para mejorar la organización de la información y el uso de memoria, los lenguajes de programación utilizan variables, a las mismas que se les asigna un tipo de dato, que puede ser entero, flotante, caracteres, cadena de caracteres, booleano, entre otros, es importante aclarar que existen lenguajes de programación tipados, para los cuales es necesario declarar el tipo de dato de una variable antes de

usarla y no tipados para los cuales no es necesaria una declaración de variables, sin embargo ambos tipos de lenguajes utilizan espacios de memoria para almacenar la información en las variables. Cuando los tipos de datos primitivos, que vienen como parte del lenguaje de programación, no satisfacen las necesidades de almacenamiento de información, entran en escena otras estructuras, para Weiss (2013), "Las estructuras de datos son colecciones de datos organizados de tal manera que permiten un acceso y modificación eficientes. Estas estructuras son esenciales para la implementación de algoritmos y la gestión de datos en la programación", en este artículo no profundizaremos en la teoría sobre las estructuras de datos, no obstante, se mencionará que pueden ser también definidas por el usuario.

Desarrollo del pensamiento

En general y para varios autores reconocidos, el desarrollo del pensamiento es un proceso del ser humano mediante el cual mejora y amplía sus capacidades cognitivas, a medida que transcurre el tiempo, en este sentido se puede hablar de la evolución de habilidades como el razonamiento, la resolución de problemas, la toma de decisiones, la creatividad y el pensamiento crítico, algunos autores afirman que, los factores biológicos, culturales, educativos y sociales, tienen una fuerte influencia en el desarrollo del pensamiento, todo esto contribuye a que el individuo pueda comprender y adaptarse a su entorno, a lo largo de su vida, de una manera eficaz.

Piaget (1952), define al desarrollo del pensamiento, como "un proceso evolutivo que se despliega en etapas específicas, cada una caracterizada por un tipo particular de organización cognitiva", el autor afirma que el pensamiento se desarrolla, mediante la interacción activa del niño con su entorno, a partir del cual, construye esquemas mentales más complejos y abstractos a medida que madura.

Para Vygotsky (1978), el desarrollo del pensamiento es un proceso sociocultural mediado por el lenguaje y la interacción social, el autor afirma que las habilidades cognitivas se desarrollan inicialmente en un nivel social, a través de la comunicación y colaboración con otros individuos, para luego internalizarse en sí mismo.

Gardner (1983), con su teoría de las inteligencias múltiples, afirma que el individuo no tiene una sola inteligencia, si no, varias inteligencias, por lo tanto, el desarrollo del pensamiento, no se da como un proceso uniforme, por el contrario, se desarrollan múltiples inteligencias, entre ellas lingüística, musical, lógica matemática y otras.

Pensamiento computacional

Iniciamos recordando la definición básica de algoritmo, como el conjunto de instrucciones que permiten resolver un problema, de ahí se puede decir que la programación, es la acción de diseñar instrucciones lógicas, detalladas, comprensibles para el computador de tal forma que éste las pueda ejecutar sin dificultades, para la resolución de tareas específicas, este tipo de pensamiento, tiene como base la lógica y las matemáticas, en este contexto, Wing (2006), es quien popularizó el término pensamiento computacional y lo define como una forma de pensamiento que relaciona la formulación de problemas y su resolución, de tal forma que se pueda representar como una secuencia de pasos o instrucciones que el computador comprenda plenamente y los pueda ejecutar (p. 33). Refuerza este concepto Papert (1980), indicando que este tipo de pensamiento hace uso de estrategias como la modelización, la simulación, y el análisis algorítmico, para resolver problemas en distintas áreas y disciplinas.

En este contexto podemos anotar que al conjunto de habilidades cognitivas que dotan al individuo de capacidades para lograr una adecuada descomposición de problemas, identificar patrones, abstraer conceptos y proponer soluciones en forma de algoritmos, se las conoce con el nombre de pensamiento computacional, esta forma de pensamiento, es parte del ser humano que va evolucionando con el tiempo y le permite resolver problemas comunes que no necesariamente estén relacionados a la informática.

Pensamiento Lógico

La lógica desde sus inicios se definió como el estudio del razonamiento válido, donde se explica, que, si las premisas son verdaderas, la conclusión debe también ser verdadera, esta estudia la estructura de los pensamientos y en consecuencia las formas de llegar a conclusiones correctas o verdaderas a partir de alguna información o idea. La lógica es fundamental para el estudio de otros campos como la filosofía, las matemáticas, la informática, y en general en cualquier campo donde se requiera pensar de manera clara y coherente.

En este mismo contexto, para (Martinez et al., 2023), “el pensamiento lógico se refiere a la capacidad de razonar de manera coherente, analítica y sistemática”, utilizando principios de la lógica para concretar el análisis de situaciones, resolver problemas y tomar decisiones. Según (Copi

et al., 2014), este pensamiento tiene su base en la capacidad de identificar relaciones entre varios elementos, establecer conexiones entre ideas, y seguir un proceso ordenado de deducción o inducción, con el propósito de llegar a conclusiones válidas y fundamentadas.

El pensamiento lógico como un instrumento en la programación, ayuda al desglose o segmentación de los problemas en pasos lógicos y secuenciales, de tal forma que un problema grande, pueda ser dividido en problemas más pequeños, para ser analizados, diseñar una propuesta para cada caso y finalmente integrar las diferentes soluciones en un solo algoritmo o programa, que permita obtener las respuestas deseadas luego de su ejecución. En este aspecto, se aprovechan las capacidades de modularidad de los diferentes lenguajes de programación, en donde cada módulo forma parte de una solución más grande, sin embargo, son individualmente funcionales y cada uno tiene una lógica interna clara. De esta manera los fundamentos de programación ayudan a desarrollar el pensamiento lógico, una habilidad esencial que puede aplicarse en diferentes disciplinas.

Como se ha visto, el pensamiento lógico, como parte del desarrollo del pensamiento, es muy útil, para resolver problemas que puedan convertirse en instrucciones lógicas y secuenciales, que demuestren una coherencia implícita en su resolución, por consiguiente, estamos aproximándonos a la definición de algoritmo, en la programación, si hacemos un análisis de las características del pensamiento lógico, reforzamos aún más el vínculo encontrado:

- **Coherencia:** el razonamiento sigue un orden lógico, evitando contradicciones y ambigüedades.
- **Deducción:** actividad de la mente que identifica, extrae y separa, una parte de un todo, para obtener conclusiones particulares a partir de premisas universales, es decir, si las premisas son verdaderas, la conclusión será verdadera.
- **Indicción:** capaz de obtener conclusiones generales a partir de premisas particulares.
- **Abstracción:** proceso mental para representar y manejar conceptos complejos de manera simplificada.
- **Objetividad:** estudia los hechos y reglas de forma clara en lugar de suponer o emplear emociones.

Resolución de Problemas

La programación por sus características, es una actividad encaminada a la resolución de problemas, cada nuevo programa, supone un nuevo desafío que necesariamente, requiere una solución

eficiente, para (Jonassen, 2000), la resolución de problemas se convierte en un proceso estructurado en donde se identifica un problema, se formula una estrategia, se proponen e implementan posibles soluciones y se evalúan los resultados con propósitos de mejorar las decisiones futuras.

De esta manera, se pretende mejorar las habilidades de resolución de problemas y por otra parte se busca fomentar la creatividad. Así, cada vez que se tiene un nuevo problema a resolver, el analista o diseñador del programa, aplica el pensamiento lógico para conseguir la coherencia y más atributos en la solución y el pensamiento lateral, para proponer alternativas de solución, que pueden percibirse como fuera de lo común, esto permite considerar múltiples enfoques para resolver un problema.

Pensamiento Crítico

(Facione, 1990, p. 3), define el pensamiento crítico como "el juicio intencionado y autorregulado que resulta en interpretación, análisis, evaluación e inferencia, así como la explicación de las consideraciones contextuales, conceptuales, metodológicas, normativas o basadas en criterios, sobre la cual se basa ese juicio".

Ampliando el concepto, se puede decir que el pensamiento crítico, es de tipo reflexivo, donde el razonamiento orienta hacia el qué creer o qué hacer, en este sentido, es evidente que el pensamiento crítico es otra habilidad que se desarrolla con la programación. El proceso de resolución de problemas en programación implica varios pasos clave para abordar eficazmente los desafíos técnicos, inicialmente, se puede descomponer el problema en partes más pequeñas y manejables, para facilitar su comprensión y tratamiento, posteriormente, se diseña un algoritmo que servirá de base para la implementación del código, de esta manera se puede asegurar que cada proceso sirve para la consecución del objetivo final. Todos estos pasos requieren una evaluación crítica de las ventajas y desventajas que se presentan en cada enfoque.

Por otra parte, la programación obliga a mejorar la capacidad para cuestionar y analizar en profundidad los problemas propuestos, evitando tener soluciones superficiales que, como consecuencia, pierdan la sostenibilidad en el tiempo y presenten serias dificultades para su corrección y mantenimiento.

Como es evidente, los fundamentos de programación, contribuyen de una forma efectiva al desarrollo del pensamiento crítico, ya que el programador debe comprender el problema al que se enfrenta, para luego razonar sobre el mismo y sus posibles alternativas de solución, la reflexión es

necesaria en todo momento para indicar al individuo que es lo que debe hacer, que estructuras de programación utilizar, que tipos de datos debe declarar, cual es el flujo de la información y como se va transformando la misma, cuáles son los datos de prueba con sus entradas y las salidas esperadas, que procesos elaborar, todo este conjunto de cuestionamientos pueden tener una respuesta efectiva mediante la aplicación del pensamiento crítico.

Atención al Detalle

La atención al detalle es una capacidad cognitiva fundamental del ser humano, que permite enfocarse en los elementos específicos de una tarea, facilitando la identificación y corrección de errores potenciales. Evitar la distracción, mientras se mantiene la concentración en detalles específicos, es una manifestación de poseer la habilidad de atención al detalle (Frontiers, 2024; Forage, 2023).

En programación, la precisión es de suma importancia ya que, incluso desde los errores más pequeños, como la ausencia de una coma, un nombre de variable incorrecto, hasta errores más grandes como un llamado a función omitiendo parámetros, pueden causar fallos en el programa. Frente a esta necesidad de precisión el programador, por obligación, debe mejorar la atención al detalle. De esta manera la programación contribuye para que el encargado de escribir código, lo haga de una forma meticulosa, precisa y consciente, desarrollando una habilidad que es valiosa no solo en la programación, sino también en cualquier tarea que requiera atención al detalle.

Por otra parte, la realización de las pruebas y depuración, para los programadores novatos, requiere de mucha concentración, para ir verificando paso a paso el flujo de la información, mientras se colocan valores a las variables de acuerdo a cada instrucción, así mismo se requiere entender y aplicar las diferentes estructuras de control para que una entrada se convierta en una salida correcta. Se puede hacer uso de la prueba de escritorio. Los procesos de mantenimiento, por su parte, demandan de múltiples habilidades, desde conocer el lenguaje de programación utilizado, hasta comprender al detalle la lógica de la programación empleada en el código que está revisando. Todo lo expuesto, demuestra de que, a través de los fundamentos de programación, se puede desarrollar las habilidades de atención al detalle.

Pensamiento Algorítmico

En párrafos anteriores ya se definió el término algoritmo, como el conjunto de instrucciones, lógicas, secuenciales y ordenadas que permiten resolver un problema o realizar una acción específica, bajo este concepto, la programación, obligatoriamente, enseña a pensar en términos de algoritmos. Esta habilidad fortalece al pensamiento computacional y tiene aplicaciones que están más allá de la programación.

El pensamiento algorítmico contribuye a resolver problemas complejos de manera estructurada y lógica, facilitando su resolución, dichos problemas, pueden salirse del ámbito de la computación, sin embargo, este tipo de pensamiento, sigue siendo efectivo, por ejemplo en el campo de la medicina, un médico experto, puede crear un algoritmo que contenga todas las instrucciones necesarias para realizar una intervención quirúrgica, con éxito, en este caso no se requiere un computador, para implementar las ordenes y luego probarlas, será otro médico quien siga paso a paso todas las instrucciones para cumplir el objetivo planteado.

De esta manera, es evidente, que el pensamiento algorítmico en todas sus concepciones y con todas sus características, contribuye a desarrollar el pensamiento del ser humano.

Materiales y métodos

La presente investigación es de carácter exploratorio y descriptivo. Para llevarla a cabo, se realizó un estudio bibliográfico que aborda dos temas principales: los fundamentos de la programación y el desarrollo del pensamiento. El tratamiento de la información se dividió en cuatro etapas. En la primera, se evaluó la validez y calidad de los autores y publicaciones seleccionadas, se identificaron las fuentes primarias por su contribución novedosa al área de conocimiento, y se definieron las fuentes secundarias en función de su interpretación y posible reorganización de la información primaria, además de analizar la información complementaria. En la segunda etapa, se validó, filtró y seleccionó la información, purgando los contenidos según los autores y textos relevantes ya identificados. La tercera etapa consistió en organizar, ordenar y redactar los contenidos seleccionados para el trabajo. Finalmente, la cuarta etapa se centró en desarrollar la propuesta resultante del análisis de cómo los fundamentos de programación se aplican al desarrollo del pensamiento.

La investigación exploratoria busca comprender la relación entre los fundamentos de programación y el desarrollo del pensamiento, mientras que la investigación descriptiva se enfoca en analizar las características y detalles de los fenómenos o elementos estudiados. Este enfoque es útil en las primeras fases de la investigación, ya que permite definir y resumir en lugar de clasificar o dividir. La investigación proporciona directrices para aplicar las teorías de fundamentos de programación hacia una aproximación al desarrollo del pensamiento. Para la revisión bibliográfica, selección de autores y curación de contenidos, se utilizó el método deductivo, que parte de las generalidades para llegar a las particularidades de los conceptos, procesos, medios y elementos relacionados con los temas principales.

Resultados y discusiones

Resultados

Al analizar toda la temática sobre fundamentos de programación y su impacto en el desarrollo del pensamiento se han encontrado varios hallazgos muy significativos. Como una primera parte, se identificó, que los conceptos básicos de programación, como algoritmos, estructuras de control, estructuras de datos y lógica de programación, juegan un papel muy importante en el desarrollo de habilidades cognitivas. Los estudios revisados, indican que la resolución de problemas basados en los fundamentos de programación y representados por los algoritmos, mejoran las habilidades del pensamiento computacional, crítico, lógico, algorítmico, habilidades para la resolución de problemas y atención al detalle, estas habilidades no solamente son aplicables para el diseño e implementación de programas de computadora, sino que pueden aplicarse a cualquier área del conocimiento.

Como una segunda parte, se evidenció que mediante el aprendizaje de la programación se favorece a la capacidad de descomponer problemas complejos en partes más pequeñas y manejables, una habilidad esencial para el desarrollo del pensamiento analítico.

Como una tercera parte, la investigación demuestra que la programación fortalece la capacidad para pensar de manera abstracta y generalizar soluciones, aplicado al diseño de algoritmos, implica que el programador, debe estar en capacidad de abstraer elementos, objetos, conceptos, etc. del mundo real hacia el programa y por otra parte, entender que un algoritmo debe ser general, es decir que debe funcionar para todos los casos y entregar respuestas correctas, para un conjunto de datos de entrada determinados.

El uso de lenguajes de programación fortalece las habilidades de atención al detalle, de esta manera se consigue diseñar algoritmos coherentes, lógicos y evitar errores de programación como fallas en la sintaxis, por ejemplo, hay lenguajes que utilizan un punto y coma para terminar una instrucción, en este caso, olvidarse de colocar el punto y coma al finalizar la instrucción, es un error de sintaxis, las pruebas y depuración, por sus características, colaboran al desarrollo de la atención al detalle.

El pensamiento algorítmico y computacional son muy cercanos y comparten características como el orden, la lógica, la secuencia, atributos que ayudan a una adecuada descomposición del problema, para lograr una mejor comprensión del mismo, así el desarrollo del pensamiento con la ayuda de estas herramientas permite al individuo, comprender su entorno y adaptarse a él sin dificultades, con propósitos de resolver problemas o alcanzar objetivos.

De lo expuesto en párrafos anteriores, se extraen las ideas fundamentales de lo serían, los elementos de los fundamentos de programación, trabajando para conseguir un desarrollo del pensamiento óptimo, en la tabla a continuación se vincula estos dos conceptos macro.

Aproximación: Fundamentos de Programación – Desarrollo del Pensamiento.

Fundamentos de Programación	de Desarrollo del Pensamiento	Descripción
Algoritmos	Pensamiento lógico y secuencial	Fomenta la habilidad para diseñar pasos lógicos y ordenados en la resolución de problemas.
Estructuras de Datos	Organización y estructura de la información	Contribuye para comprender cómo organizar, almacenar y acceder a la información de manera eficiente.
Lógica de Programación	Razonamiento crítico y deductivo	Desarrolla la capacidad de aplicar reglas lógicas para tomar decisiones y resolver problemas complejos.
Modularidad (Funciones y procedimientos)	Descomposición y abstracción	Potencia la habilidad para dividir problemas complejos en partes más pequeñas y manejables.
Depuración (Debugging)	Pensamiento crítico y resolución de problemas	Fomenta la capacidad de identificar errores, analizarlos y aplicar soluciones correctivas.
Paradigmas de Programación (POO, Funcional, etc.)	Pensamiento abstracto y adaptabilidad	Promueve la habilidad para trabajar con conceptos abstractos y aplicar diferentes enfoques para resolver un mismo problema.

Eficiencia y Optimización	Pensamiento analítico y toma de decisiones	Desarrolla la capacidad de evaluar distintas soluciones y elegir la más eficiente en términos de recursos y tiempo.
Sintaxis y Semántica	Precisión y atención al detalle	Refuerza la importancia de la precisión en la escritura y comprensión de instrucciones para evitar errores.
Recursividad	Pensamiento recursivo y resolución de problemas complejos	Facilita el entendimiento para resolver problemas donde un segmento de código se llama a sí mismo, de forma recursiva.

Fuente: Elaboración propia

Las estructuras de control, permiten al programa, controlar el flujo de la información, determinando los cambios a los que se someten las variables y estructuras de datos, en cada siguiente instrucción, visto de esta manera, el uso de las estructuras de control, son una forma eficiente de conseguir un desarrollo del pensamiento efectivo, en diferentes ámbitos, tal como lo demuestra la siguiente tabla.

Estructuras de control: Una aproximación al desarrollo del pensamiento.

Estructuras de Control	Desarrollo del Pensamiento	Descripción
Secuencial	Pensamiento secuencial y ordenado	Desarrolla la capacidad de seguir y ejecutar instrucciones en un orden específico, fortaleciendo el pensamiento lineal.
Condicionales (if, else, switch)	Razonamiento lógico y toma de decisiones	Potencia la habilidad para evaluar condiciones y tomar decisiones basadas en criterios lógicos, promoviendo el pensamiento crítico.
Bucles (for, while, do-while)	Pensamiento iterativo y resolución sistemática	Fomenta la capacidad de repetir procesos hasta alcanzar un resultado deseado, mejorando la perseverancia y el pensamiento sistemático.
Estructuras de Control Anidadas	Pensamiento jerárquico y organización	Ayuda a comprender la complejidad de problemas mediante la organización de decisiones dentro de otras decisiones, fortaleciendo el pensamiento jerárquico.
Recursividad	Pensamiento recursivo y descomposición	Facilita la comprensión de problemas en los que una solución se basa en resolver una versión más pequeña del mismo problema, promoviendo el pensamiento inductivo.
Interrupciones y Excepciones (try, catch)	Adaptabilidad y manejo de la incertidumbre	Desarrolla la capacidad de anticipar y manejar situaciones imprevistas o errores, mejorando la adaptabilidad y la capacidad para enfrentar la incertidumbre.

Control de Flujo (break, continue)	Flexibilidad y toma de decisiones rápida	Fomenta la habilidad para modificar el flujo de ejecución en respuesta a condiciones específicas, promoviendo la flexibilidad mental.
---	---	---

Fuente: Elaboración propia

Discusión

Los resultados enfatizan la relación positiva entre los fundamentos de programación y el desarrollo del pensamiento, dando lugar a la hipótesis, de que la programación no solamente es una habilidad técnica, sino que también puede ser una herramienta eficaz, para mejorar el pensamiento lógico, computacional, crítico y algorítmico. La capacidad de descomponer problemas complejos en pasos lógicos y la habilidad para pensar de manera abstracta se ven reforzadas por el aprendizaje de la programación, lo cual es consistente con teorías cognitivas que sugieren que la práctica en un campo específico puede desarrollar habilidades generales de pensamiento.

Sin embargo, es importante reconocer que el impacto de la programación en el desarrollo del pensamiento puede variar dependiendo de la metodología de enseñanza y del contexto educativo. La investigación sugiere que una enseñanza efectiva de la programación debe enfocarse en la resolución de problemas y en la aplicación práctica de conceptos, en lugar de centrarse exclusivamente en la memorización de sintaxis o en la ejecución de ejercicios repetitivos.

Se ha encontrado también, que, la relación entre la programación y el desarrollo del pensamiento requiere una evaluación continua y adaptativa. Las herramientas y enfoques pedagógicos, deben apuntar hacia los avances en el campo de la programación, así como también a las necesidades en constante cambio, por parte de los estudiantes. Una propuesta efectiva sería la integración de técnicas de programación en el currículo educativo, un proceso que debe ser cuidadosamente diseñado para potenciar el desarrollo del pensamiento y facilitar la transferencia de habilidades a otros campos del conocimiento.

La programación ofrece un enfoque valioso para el desarrollo del pensamiento, proporcionando a los estudiantes herramientas y técnicas que fomentan habilidades cognitivas esenciales. La implementación efectiva, de los fundamentos de programación en la educación, puede mejorar las competencias técnicas y también enriquecer el pensamiento crítico y analítico de los estudiantes.

Conclusiones

Al constituirse, el desarrollo del pensamiento, como un proceso socio cultural, está en constante evolución y mejora sus capacidades cognitivas con el paso del tiempo, también puede verse afectado de forma positiva cuando se apoya en herramientas y metodologías que desafían y expanden las capacidades del pensamiento. Vinculando a este concepto las teorías de fundamentos de programación, se puede afirmar que estas, fomentan un pensamiento más estructurado, crítico y abstracto. A medida que el ser humano se enfrenta a problemas de programación, desarrolla habilidades esenciales para descomponer problemas complejos, tomar decisiones y generar soluciones innovadoras. Esta evolución cognitiva demuestra que la programación no solo es una competencia técnica, sino también una poderosa herramienta para fortalecer y enriquecer el pensamiento a lo largo del tiempo, preparando a la persona para los desafíos futuros en diversos campos.

Los fundamentos de programación, como la lógica, las estructuras de control y la descomposición de problemas, se han incorporado como elementos clave en el desarrollo del pensamiento. A través de la práctica de programación, los estudiantes no solo adquieren habilidades técnicas, sino que también mejoran su capacidad para pensar de manera lógica, crítica y estructurada. Los hallazgos de esta investigación evidencian que la programación actúa como un catalizador para el desarrollo del pensamiento abstracto y sistémico, permitiendo a los estudiantes abordar problemas complejos con mayor eficacia. La vinculación entre la programación y el desarrollo del pensamiento, demuestra la importancia de integrar en el currículo las técnicas de programación, con el propósito de preparar a los estudiantes para enfrentar desafíos ayudados de un pensamiento más robusto y adaptable.

Los fundamentos de programación ofrecen una aproximación importante al desarrollo del pensamiento, proporcionando herramientas y métodos que fortalecen habilidades cognitivas fundamentales. A través del aprendizaje y la aplicación de conceptos como la lógica, las estructuras de control y la resolución de problemas, se promueve un pensamiento más crítico, abstracto y sistemático. Finalmente, los fundamentos de programación se perfilan como un enfoque integral y valioso para el desarrollo del pensamiento teniendo en cuenta que este es un constructo socio cultural.

Referencias

1. Pomeroy, A. (2007). *The abacus: A history of the world's first computing machine*. Princeton University Press.
2. Stallings, W. (2021). *Computer Organization and Architecture*. Pearson. (Capítulos sobre lenguaje ensamblador y arquitectura de computadoras).
3. Patterson, D. A., & Hennessy, J. L. (2021). *Computer Organization and Design: The Hardware/Software Interface*. Morgan Kaufmann. (Capítulo sobre ensamblador y lenguaje de máquina).
4. Lutz, M. (2023). *Learning Python*. O'Reilly Media. (Referencia fundamental para Python, un lenguaje de alto nivel muy popular).
5. Bloch, J. (2018). *Effective Java*. Addison-Wesley. (Guía avanzada sobre las mejores prácticas en Java).
6. Stroustrup, B. (2013). *The C++ Programming Language*. Addison-Wesley. (Libro fundamental sobre C++, un lenguaje de alto nivel con capacidades de bajo nivel).
7. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to Algorithms* (4th ed.). MIT Press.
8. Heineman, G. T., Pollice, G., & Selkow, S. (2016). *Algorithms in a Nutshell* (2nd ed.). O'Reilly Media.
9. Knuth, D. E. (2011). *The Art of Computer Programming, Volume 1: Fundamental Algorithms* (3rd ed.). Addison-Wesley.
10. Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (2020). *Data Structures and Algorithms* (3rd ed.). Addison-Wesley.
11. Tanenbaum, A. S. (2016). *Structured Computer Organization* (6th ed., pp. 114-116). Pearson.
12. Lafore, R. (2017). *Data Structures and Algorithms in Java* (3rd ed., pp. 112-115). Sams Publishing.
13. Weiss, M. A. (2013). *Data Structures and Algorithm Analysis in C++* (4th ed., pp. 45-48). Pearson.
14. Piaget, J. (1952). *The origins of intelligence in children* (Vol. 8). International Universities Press.

15. Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes* (M. Cole, V. John-Steiner, S. Scribner, & E. Souberman, Eds. & Trans.). Harvard University Press.
16. Gardner, H. (1983). *Frames of mind: The theory of multiple intelligences* (10th ed.). Basic Books.
17. Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
18. Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books.
19. Martinez Ruiz, J. E., Cardenas Rodriguez, M. M., Junco Rosario, G. L., & Cabezas Cabezas, H. S. (2023). El desarrollo del pensamiento lógico a través del proceso de aprendizaje en los estudiantes universitarios. <https://doi.org/10.5281/zenodo.10420749>
20. Copi, I. M., Cohen, C., & McMahon, K. (2014). *Introduction to logic* (14th ed.). Pearson
21. Jonassen, D. H. (2000). Toward a design theory of problem solving. *Educational Technology Research and Development*, 48(4), 63-85. <https://doi.org/10.1007/BF02300500>
22. Facione, P. A. (1990). *Critical thinking: A statement of expert consensus for purposes of educational assessment and instruction*. The California Academic Press.
23. Frontiers. (2024). *Attention in Psychology, Neuroscience, and Machine Learning*. Frontiers Media SA. Recuperado de <https://www.frontiersin.org/articles/10.3389/fnins.2020.00023/full>.
24. Forage. (2023). *Attention to Detail: Definition and Examples*. Recuperado de <https://www.theforage.com/blog>.

© 2024 por los autores. Este artículo es de acceso abierto y distribuido según los términos y condiciones de la licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional (CC BY-NC-SA 4.0) (<https://creativecommons.org/licenses/by-nc-sa/4.0/>).