**Pensamiento algorítmico y pensamiento computacional y la Aplicación de Algoritmos en la Pedagogía Matemática**

El pensamiento algorítmico, una competencia fundamental, es crucial para abordar problemas prácticos formulados matemáticamente, siendo indispensable en campos como la física, las matemáticas, la ingeniería y las ciencias de la computación. Un algoritmo se define como una secuencia organizada y precisa de pasos diseñada para resolver una tarea específica, cuyo origen etimológico se remonta al matemático persa del siglo IX, al-Khowârizmî.

Dentro del ámbito de la enseñanza de las matemáticas, la algoritmización implica la integración de algoritmos y el cultivo del pensamiento algorítmico en el proceso educativo. La naturaleza de las matemáticas escolares es dual: conceptual y algorítmica, y ambos componentes son interdependientes y vitales. Cuando los algoritmos se presentan desvinculados de sus fundamentos conceptuales, corren el riesgo de ser percibidos meramente como procedimientos automáticos para realizar cálculos. El presente ensayo explorará la definición y características del pensamiento algorítmico, su interconexión con el pensamiento computacional, las habilidades que potencia y su aplicación práctica en la pedagogía matemática, así como los beneficios y desafíos inherentes a esta integración.

**1. Fundamentos del Pensamiento Algorítmico: Concepto y Atributos**

Para dilucidar la trascendencia del pensamiento algorítmico en la educación, es indispensable establecer una clara comprensión de sus elementos constitutivos.

**1.1. ¿Qué es un Algoritmo?**

Un algoritmo es, en su esencia, un conjunto finito de instrucciones bien definidas que, cuando se ejecutan en un orden específico, producen una solución a un problema. Se trata de una especificación detallada o un plan de resolución para una tarea, expresado mediante operaciones que pueden ser comprendidas y realizadas por un ejecutor. Representa una sucesión de acciones precisas que conducen a un resultado determinado. En un sentido más amplio y menos formal, un algoritmo puede conceptualizarse como una "receta, proceso, método, técnica o rutina". La existencia de un algoritmo implica la potencial automatización de una tarea.

**1.2. Atributos Esenciales de los Algoritmos**

Para que una serie de instrucciones sea catalogada como un algoritmo, debe satisfacer ciertas propiedades fundamentales:

1. Operaciones Elementales: Cada operación constitutiva del algoritmo debe ser controlable o comprensible para el ejecutor. La elementalidad es un atributo relativo, determinado por el nivel de habilidad o conocimiento del estudiante.
2. Unívocidad (Unambiguity): El algoritmo debe especificar de forma inequívoca la secuencia de operaciones que conducirán al resultado. Esto garantiza que, dados los mismos datos iniciales, el algoritmo siempre generará el mismo resultado.
3. Efectividad (Effectiveness): Un algoritmo asegura que la ejecución de sus pasos resultará en la solución correcta de la tarea en un número finito de operaciones. Cada operación individual debe ser lo suficientemente básica para ser ejecutada con exactitud en un tiempo finito.
4. Generalidad (Generality): Un algoritmo debe ser aplicable a una clase entera de problemas, operando sobre parámetros cuya especificación define una instancia particular de la tarea. Su utilidad no se restringe a una única situación, sino que abarca un grupo de problemas similares.
5. Finitud (Finiteness): Un algoritmo debe culminar su ejecución en un número predefinido y finito de pasos, independientemente de las condiciones iniciales válidas.
6. Entrada/Salida (Input/Output): Un algoritmo procesa datos de entrada y genera datos de salida. Los datos de entrada son símbolos que representan objetos, eventos o propiedades pertinentes para la resolución del problema.

Estas características confieren al algoritmo su naturaleza de herramienta fiable y sistemática para la resolución de problemas, una cualidad vital tanto en el campo de la informática como en el de las matemáticas.

**2. Diversidad de Algoritmos: Un Marco Categórico Amplio y Multidisciplinario**

La concepción de algoritmo trasciende con creces las fronteras puramente matemáticas o computacionales, manifestándose en la práctica totalidad de las facetas de la vida, desde las actividades cotidianas más simples hasta los sistemas de inteligencia artificial más complejos. Para una comprensión exhaustiva de su alcance, resulta útil clasificarlos en función de su ámbito de aplicación, estructura lógica y propósito inherente.

**2.1. Clasificación por Ámbito de Aplicación**

1. Algoritmos Cotidianos: Se refieren a los procedimientos paso a paso que se siguen en las actividades diarias, como el proceso de cepillarse los dientes o la secuencia de una receta de cocina. Son intrínsecamente intuitivos y no requieren una formalización escrita.
2. Algoritmos Matemáticos y Científicos: Corresponden a los métodos empleados para resolver problemas numéricos o teóricos. Ejemplos prominentes incluyen el Algoritmo de Euclides para calcular el Máximo Común Divisor (MCD) y el Método de Newton-Raphson para la aproximación de raíces. Su distintivo es el rigor y la precisión.
3. Algoritmos Computacionales: Son aquellos diseñados específicamente para ser ejecutados por sistemas informáticos. Esto abarca algoritmos de ordenamiento (como QuickSort) y de búsqueda (como la búsqueda binaria). Su optimización se centra en la eficiencia temporal y espacial.
4. Algoritmos de Toma de Decisiones: Estos algoritmos guían la selección de opciones basándose en datos o reglas preestablecidas. Ejemplos incluyen los sistemas de recomendación (como los utilizados por Netflix) y los algoritmos de ruta óptima (empleados en sistemas GPS). Su función es integrar datos y lógica para maximizar resultados.
5. Algoritmos Creativos y Artísticos: Se encargan de generar contenido o soluciones innovadoras y artísticas. Un ejemplo sería la generación de música mediante Inteligencia Artificial (por ejemplo, OpenAI’s MuseNet) o la creación de patrones fractales. Su característica principal es la combinación de creatividad con procesos estructurados.

**2.2. Clasificación por Estructura Lógica**

1. Algoritmos Iterativos: Aquellos que repiten una serie de pasos hasta que se cumple una condición específica, como la búsqueda de un elemento en una lista mediante un ciclo while.
2. Algoritmos Recursivos: Se definen a sí mismos en términos de instancias más pequeñas del mismo problema, siendo el cálculo del factorial (n!) un ejemplo clásico.
3. Algoritmos Heurísticos: Proporcionan soluciones aproximadas a problemas de alta complejidad cuando una solución exacta es inviable o computacionalmente costosa. Los algoritmos genéticos para optimización son un caso representativo.
4. Algoritmos Paralelos y Distribuidos: Ejecutan tareas de manera concurrente en múltiples procesadores o sistemas interconectados, como el paradigma MapReduce en el procesamiento de datos en la nube.

**2.3. Clasificación por Propósito**

1. Algoritmos de Optimización: Su objetivo es maximizar o minimizar una función objetivo bajo ciertas restricciones, como el algoritmo del simplex en programación lineal o el gradiente descendente en aprendizaje automático.
2. Algoritmos de Simulación: Modelan sistemas y procesos del mundo real para predecir comportamientos o analizar dinámicas, ejemplificado por modelos climáticos o simulaciones de tráfico.
3. Algoritmos de Aprendizaje Automático: Permiten a los sistemas aprender patrones a partir de datos sin ser programados explícitamente. Incluyen redes neuronales para clasificación de imágenes y árboles de decisión para agrupamiento de datos (clustering).
4. Algoritmos Éticos o de Fairness: Se diseñan con la intención de asegurar decisiones automatizadas justas y equitativas, como los sistemas de reclutamiento que procuran evitar sesgos demográficos.

La ubicuidad de los algoritmos los convierte en herramientas aplicables en la mayoría de los dominios de la existencia. Como articuló Alan Turing (1936), los algoritmos son "secuencias de pasos" capaces de modelar desde procesos cognitivos hasta el funcionamiento de las máquinas. Esta universalidad subraya su papel indispensable en la era digital.

**3. Métodos Matemáticos: ¿Son Equivalentes a los Algoritmos?**

La correlación entre los métodos matemáticos y los algoritmos es un tema que requiere una consideración matizada, ya que depende intrínsecamente del contexto específico.

**3.1. Métodos Matemáticos con Naturaleza Algorítmica**

Ciertos métodos matemáticos, en efecto, poseen una naturaleza algorítmica, al adherirse a las propiedades de una secuencia de pasos repetibles y deterministas. Ejemplos arquetípicos incluyen el Algoritmo de Euclides para determinar el Máximo Común Divisor, la Regla de Cramer para la resolución de sistemas de ecuaciones lineales, y el Método de Gauss-Jordan para la reducción de matrices. La clasificación de estos métodos como algoritmos se justifica por su inherente determinismo (cada paso conduce a un único resultado definido), su propiedad de terminación (garantizan una solución en un número finito de pasos) y su generalidad (son aplicables a una clase completa de problemas).

**3.2. Métodos Matemáticos que Difieren de los Algoritmos**

No todos los métodos matemáticos pueden ser categorizados como algoritmos. Algunos demandan un nivel de razonamiento abstracto, creatividad o procesos de demostración lógica que no se ajustan a la rigidez de una secuencia mecánica de instrucciones. Por ejemplo:

* Demostraciones por inducción matemática: Aunque estructuradas, requieren la identificación de un caso base, la asunción de una hipótesis inductiva y la prueba de un caso (n+1), lo cual dista de ser un procedimiento estrictamente "paso a paso" y mecánico.
* Demostraciones constructivas versus no constructivas: Mientras que una demostración constructiva puede incorporar un algoritmo para la creación de un objeto, una demostración no constructiva simplemente afirma la existencia de dicho objeto sin detallar un método explícito para encontrarlo.
* Teoría de conjuntos abstracta: Ciertos resultados en este campo, como el teorema de Cantor, se fundamentan en argumentos lógicos profundos y no en procedimientos computacionales.

Estos ejemplos ilustran cómo las matemáticas trascienden el ámbito estrictamente algorítmico, incorporando elementos de abstracción, generalización y razonamiento deductivo que no siempre se traducen directamente en algoritmos (Turing, 1936).

**4. Pensamiento Algorítmico y Pensamiento Computacional: Una Relación Entrelazada**

El pensamiento algorítmico y el pensamiento computacional son constructos esenciales en la era digital, fundamentales para abordar problemas complejos mediante la aplicación de la lógica y una estructuración sistemática. Aunque a menudo se emplean de forma intercambiable, poseen características diferenciadas que los hacen complementarios.

**4.1. El Pensamiento Algorítmico**

El pensamiento algorítmico alude a la capacidad de concebir, interpretar y aplicar algoritmos. Esto implica trabajar con secuencias de pasos finitas, bien definidas y repetibles, orientadas a la resolución de un problema o la ejecución de una tarea específica. Entre sus componentes clave se encuentran: la descomposición (fragmentar un problema en subproblemas manejables), el ordenamiento (establecer el flujo lógico de operaciones), la abstracción (identificar patrones y detalles pertinentes) y la generalización (desarrollar soluciones aplicables a múltiples escenarios). Su génesis se atribuye a figuras históricas como Alan Turing y Edsger Dijkstra, siendo una habilidad crucial en programación, matemáticas y ciencias de la computación (Cormen et al., 2009).

**4.2. El Pensamiento Computacional**

El pensamiento computacional, concepto introducido por Jeannette Wing en 2006, representa un enfoque amplio para la resolución de problemas, sustentado en principios inherentes a las ciencias de la computación. Wing (2006) lo define como la habilidad para descomponer problemas, reconocer patrones, abstraer los detalles irrelevantes y formular soluciones mediante algoritmos o sistemas computacionales. Este enfoque trasciende la mera programación, hallando aplicación en diversas disciplinas, desde la biología hasta el arte.

**4.3. Sinergia entre el Pensamiento Algorítmico y Computacional**

El pensamiento algorítmico constituye un pilar central del pensamiento computacional, aunque este último engloba un espectro más amplio de habilidades. Mientras el pensamiento algorítmico se concentra en la elaboración de secuencias lógicas para resolver problemas específicos, el pensamiento computacional integra un conjunto multifacético de destrezas para abordar desafíos de manera holística. Su relación puede visualizarse como una inclusión jerárquica: el pensamiento computacional abarca el algorítmico, la descomposición, la abstracción y el diseño de sistemas. Existe una interdependencia intrínseca: un algoritmo eficiente requiere del pensamiento computacional para su diseño, y, a su vez, el pensamiento computacional se apoya en algoritmos para su implementación. Por ejemplo, en el desarrollo de una aplicación de entrega a domicilio, el pensamiento computacional guía la modularización del problema (descomposición) y la elección de algoritmos de optimización de rutas (pensamiento algorítmico). Ambos son herramientas interconectadas, cuya relevancia en el siglo XXI es innegable (Wing, 2006).

**5. Habilidades Potenciadas por el Pensamiento Algorítmico**

El cultivo del pensamiento algorítmico no solo requiere, sino que también fomenta, un conjunto robusto de habilidades cognitivas y metacognitivas:

1. Especificación y Análisis de Problemas: Implica la capacidad de definir con claridad un problema y desglosarlo en componentes más pequeños, lo que comprende la descomposición, el reconocimiento de patrones y la abstracción.
2. Identificación de Procesos Básicos: Consiste en discernir las operaciones fundamentales requeridas para resolver una tarea.
3. Construcción de Algoritmos: Se refiere al diseño de secuencias lógicas de pasos, lo que demanda una planificación explícita de operaciones y una reorganización del conocimiento adquirido. Este proceso estimula el pensamiento preciso, lógico y formal.
4. Consideración de Casos y Proposición de Mejoras: Involucra la capacidad de analizar escenarios específicos y generales, así como la evaluación y optimización de la eficiencia y corrección de un algoritmo.
5. Desarrollo del Pensamiento Reflexivo y Autocontrol: Se manifiesta en la habilidad para verificar la funcionalidad del esquema algorítmico mediante la ejecución con diversos parámetros iniciales.
6. Análisis de Algoritmos Existentes: Permite al estudiante apreciar la precisión y simplicidad de una secuencia lógica, distinguir entre elementos fijos y variables, y centrarse en el método subyacente.
7. Conexión con Conocimientos Previos: Implica la facultad de relacionar el problema actual con problemas resueltos anteriormente o con teoremas ya conocidos.

La práctica sostenida es indispensable para el perfeccionamiento continuo de las habilidades de razonamiento y la comunicación efectiva a través del diseño e interpretación de procesos estructurados.

**6. La Aplicación de Algoritmos en la Pedagogía Matemática**

La incorporación consciente de algoritmos y la algoritmización en la enseñanza de las matemáticas conlleva múltiples beneficios educativos. La seguridad que ofrece un algoritmo al conducir a un resultado correcto mejora la eficiencia y posibilita la automatización de los procesos. Los algoritmos aritméticos y algebraicos básicos son fundamentales para el desarrollo del conocimiento matemático en todos los niveles.

La idea es tomar aquellos métodos matemáticos susceptibles de algoritmización y presentar los métodos de solución de manera que el estudiante pueda identificar el contexto de aplicación del método, las entradas (datos a considerar), las salidas (incógnitas a encontrar) y el procedimiento a aplicar (algoritmo o método). De modo que con la aplicación consistente y coherente del algoritmo pueda resolver, comprobar los resultados y reflexionar sobre el procedimiento seguido.

Sin embargo, esto implica que el docente debe hacer esfuerzos adicionales para presentar conceptos claros y entendibles y un proceso que el estudiante pueda seguir sin ambigüedades.

Así podríamos indicar un flujo para la enseñanza de un método donde el docente presente los conceptos necesarios y se asegure que los estudiantes lo han comprendido, presente el algoritmo, su contexto de aplicación sus entradas y sus salidas y el paso a paso para la resolución:

1. Presentación de los conceptos necesarios
2. Explicación del algoritmo involucrado
3. Cuáles son los datos de entrada
4. Cual son las incógnitas
5. Cual el algoritmo (paso a paso) a seguir
6. Como se debe verificar el resultado
7. Que debe haber aprendido el estudiante en el proceso

Es crucial que la introducción de procedimientos algorítmicos esté precedida por una comprensión conceptual profunda del problema. Si los algoritmos prefabricados se presentan sin esta base conceptual, existe el riesgo de que las matemáticas se perciban como un mero compendio de métodos computacionales desvinculados; por ello, los aspectos conceptuales y algorítmicos deben avanzar de forma paralela.

En los materiales educativos de matemáticas, se pueden encontrar diversos ejemplos de cómo se utilizan los algoritmos y la algoritmización:

1. Algoritmos como Patrones de Acción: Se manifiesta en la enseñanza de operaciones escritas (suma, resta, multiplicación, división) o en el estudio del algoritmo de Euclides, que permiten a los estudiantes retroceder a conocimientos previos y racionalizar sus acciones.
2. Árboles-grafos: Representan la secuencia de acciones en expresiones aritméticas, sustituyendo la notación con paréntesis y ayudando a la comprensión del lenguaje matemático.
3. Máquinas Numéricas: Son representaciones gráficas de algoritmos numéricos simples que fomentan la comprensión intuitiva de las funciones y la búsqueda de procedimientos generales.
4. Búsqueda de un Procedimiento General: Actividades que requieren que los estudiantes identifiquen métodos generales para crear secuencias y formular generalizaciones algebraicas, sentando las bases para la notación algebraica.
5. Algoritmos Predefinidos en Diagramas de Bloques: Introducen a los estudiantes a la algoritmización mediante el análisis de diagramas de flujo, fomentando el uso de patrones y la formulación de hipótesis.
6. Creación de Patrones Análogos: Desafía a los estudiantes a diseñar algoritmos para problemas similares basándose en modelos conocidos, fortaleciendo el pensamiento analógico y la generalización.
7. Construcción de Algoritmos por Parte de los Estudiantes: Esta práctica es particularmente valiosa, ya que exige planificación, formulación precisa y la reorganización del conocimiento.
8. Construcciones Geométricas como Algoritmos: Las construcciones geométricas pueden ser interpretadas como planes de operación representables algorítmicamente, lo que facilita su comprensión.

La algoritmización, en su sentido más amplio, emerge como una herramienta poderosa para el enriquecimiento del conocimiento, la adquisición de nuevas habilidades y el fomento de buenos hábitos en la educación matemática.

**7. Estrategias Pedagógicas para la Enseñanza de las Matemáticas con Enfoque Algorítmico**

Para una integración efectiva del pensamiento algorítmico en las clases de matemáticas, se proponen las siguientes estrategias pedagógicas:

1. Equilibrio Conceptual y Algorítmico: Es fundamental que los docentes enfaticen la importancia de ambos aspectos, asegurando que el aprendizaje de algoritmos se acompañe de una comprensión conceptual sólida, evitando presentarlos como meras fórmulas.
2. Fomento de la Práctica Constante: La resolución recurrente de problemas que exijan el diseño o análisis de algoritmos, incluso sin la necesidad inicial de computadoras, es clave para desarrollar esta forma de pensamiento.
3. Uso de Diversas Representaciones: Exponer a los estudiantes a múltiples formas de representar algoritmos (descripciones verbales, diagramas de flujo, pseudocódigo, representaciones gráficas) enriquece su comprensión y capacidad de diseño.
4. Incentivo a la Creación y Análisis de Algoritmos Propios: Motivar a los estudiantes a construir sus propios algoritmos promueve el pensamiento lógico, formal y reflexivo, mientras que el análisis de algoritmos existentes profundiza la comprensión de la lógica y el método.
5. Promoción de la Generalidad y Optimización: Guiar a los estudiantes a identificar patrones y desarrollar procedimientos generales aplicables a clases de problemas, así como fomentar la comparación de soluciones algorítmicas para identificar métodos más eficientes.
6. Conexión entre Problemas Nuevos y Conocimientos Previos: Enseñar a los estudiantes a buscar relaciones entre problemas actuales y soluciones anteriores, así como la utilidad de problemas auxiliares más sencillos.
7. Explicitación del Propósito de los Pasos: Al presentar algoritmos o soluciones, es vital explicar el *porqué* de cada paso, especialmente aquellos menos obvios, para fomentar un razonamiento profundo.
8. Fomento de la Reflexión Post-Solución: Animar a los estudiantes a "mirar hacia atrás" después de resolver un problema, revisando el proceso y considerando si el resultado o método pueden aplicarse en otros contextos.
9. Presentación Concreta de Conceptos Abstractos: Dada la naturaleza abstracta de las matemáticas, se recomienda utilizar ejemplos tangibles siempre que sea posible para facilitar la comprensión.
10. Desafío con Problemas No Rutinarios: Más allá de los problemas de práctica básica, es esencial estimular la curiosidad de los estudiantes con desafíos proporcionales a su nivel de conocimiento, cultivando el pensamiento independiente.

Un docente que comprende los beneficios de la algoritmización puede emplearla para potenciar el pensamiento lógico y matemático de sus estudiantes, reforzar el material aprendido y fomentar la autovalidación de resultados.

**8. Beneficios y Desafíos de la Algoritmización en la Enseñanza Matemática**

La algoritmización en la enseñanza de las matemáticas presenta un balance de ventajas significativas y ciertos desafíos inherentes.

**8.1. Beneficios**

* Comprensión Profunda: Facilita la exploración y el entendimiento profundo de los conceptos matemáticos.
* Desarrollo de Habilidades: Potencia el pensamiento lógico, matemático y el hábito de verificar los resultados.
* Fundamento Conceptual: Proporciona las herramientas técnicas para arraigar el conocimiento conceptual de las matemáticas.
* Aprendizaje Holístico: Permite una comprensión integral al interconectar enfoques conceptuales y algorítmicos.
* Eficiencia y Automatismo: La certeza algorítmica de un resultado correcto mejora la eficiencia y posibilita la automatización de cálculos básicos.
* Análisis y Precisión: El análisis de algoritmos fomenta la precisión, la distinción entre elementos constantes y variables, y el enfoque en el método.
* Creatividad en Representación: Impulsa la creación de formas no verbales para representar relaciones y procedimientos.
* Desarrollo Metacognitivo: La construcción de algoritmos propios exige planificación, formulación precisa y reorganización del conocimiento, cultivando el pensamiento reflexivo y el autocontrol.
* Formalización del Razonamiento: Acostumbra al estudiante a la formalización de cuestiones matemáticas y al razonamiento deductivo.
* Competencias Prácticas: Apoya el desarrollo de habilidades prácticas, incluyendo el uso de técnicas informáticas y la presentación clara de soluciones.

**8.2. Desafíos**

* Desvinculación Conceptual: La enseñanza de algoritmos de forma aislada puede llevar a que las matemáticas sean percibidas como una colección de métodos computacionales sin conexión conceptual.
* Sobrecarga Cognitiva: Una introducción excesiva de algoritmos preestablecidos puede dificultar su memorización y generar confusión.
* Precedencia Conceptual: La enseñanza algorítmica sin una base conceptual previa puede resultar contraproducente; la solución conceptual del problema es esencial antes de introducir el procedimiento.
* Dependencia Excesiva de Herramientas: Una dependencia marcada de herramientas visuales informáticas puede desviar el enfoque del razonamiento lógico intrínseco y de la comunicación humana del algoritmo.

**Conclusión**

El pensamiento algorítmico es una habilidad esencial que trasciende la informática, siendo fundamental para la resolución de problemas en matemáticas y diversas disciplinas. Su integración en la enseñanza matemática, siempre que se equilibren los aspectos algorítmicos y conceptuales, ofrece notables ventajas educativas, ya que desarrolla el pensamiento lógico y formal, mejora la capacidad de análisis y diseño, y permite una comprensión más profunda de los procedimientos matemáticos.

Al dotar a los estudiantes de la capacidad de especificar, analizar y construir algoritmos, se les proporcionan herramientas cognitivas de valor incalculable aplicables en múltiples contextos. El objetivo pedagógico no reside meramente en formar programadores, sino en cultivar un pensamiento algorítmico que, al igual que el dominio de las operaciones básicas, constituye un prerrequisito para un desarrollo matemático más profundo y creativo. Adoptar un enfoque que valore tanto el "qué" (los conceptos) como el "cómo" (los algoritmos) preparará a los estudiantes de manera más efectiva para afrontar los desafíos lógicos y de resolución de problemas en un mundo cada vez más complejo.

**Referencias**

* Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms* (3rd ed.). MIT Press.
* Dijkstra, E. W. (1968). Go To Statement Considered Harmful. *Communications of the ACM*, 11(3), 147-148.
* Pyzara, A. (n.d.). *Algorithmisation in teaching mathem*. (Document excerpt).
* Sipser, M. (2012). *Introduction to the Theory of Computation* (3rd ed.). Cengage Learning.
* Turing, A. M. (1936). On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42(1), 230-265.
* Universidad Nacional Autónoma de México, Facultad de Estudios Superiores Acatlán. (n.d.). *Pensamiento algorítmico y resolución de problemas matemáticos*. (Document excerpt presented at the Octavo Congreso Internacional Sobre la Enseñanza y Aplicación de las Matemáticas).
* Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33-35.