



## EL RAZONAMIENTO LÓGICO MATEMÁTICO Y PENSAMIENTO COMPUTACIONAL: PROGRAMACIÓN SIN COMPUTADORAS

**Josué Raúl Vera Guzmán**

*Benemérita Escuela Normal Veracruzana*

*jverag18@gmail.com*

*Ismael Cuevas Morales*

*Benemérita Escuela Normal Veracruzana*

*ismaelbenv@gmail.com*

**Área temática:** Educación en campos disciplinares

**Línea temática:** Educación Matemática

**Tipo de ponencia:** Reporte final de investigación



### Resumen

Este estudio investigó la relación entre el pensamiento computacional y el razonamiento lógico matemático. Se propone una secuencia didáctica basada en la construcción de algoritmos para fomentar el desarrollo en estas áreas, con énfasis en que sea sin utilizar computadoras. Los resultados mostraron una fuerte prevalencia del pensamiento algorítmico y una manifestación constante de la definición de problemas. Además, se observó que el material didáctico fomenta la colaboración y el pensamiento algorítmico. Se concluye que con esta propuesta se abordan la mayoría de las habilidades del pensamiento computacional, realizando los propios alumnos la acción de computar.

**Palabras clave:** Pensamiento Lógico, Conocimiento Matemático, Programación, Educación y Computación.

### Introducción

El presente documento registra habilidades del pensamiento computacional a través de una situación didáctica con actividades sin computadora. Por ello, se destaca el uso de materiales didácticos, proponiendo una alternativa para programar ante la ausencia de equipos de cómputo. Para ello, partimos de la relación existente entre la programación, el razonamiento lógico matemático y el pensamiento computacional como nodos articuladores que permiten proponer un diseño de actividades inspirados en el entorno de programación Scratch.

Sostenemos la hipótesis que sí es posible trabajar el razonamiento lógico matemático y el pensamiento computacional a través de la programación y sin necesidad de una computadora.

Posterior al diseño, se realizó una intervención en un aula de sexto grado de educación primaria en donde se grabaron las conversaciones entre los estudiantes de primaria para luego analizar cualitativamente dichas interacciones. Finalmente, con base en una matriz reportada por la Coordinación General @prende.mx (2018) se reportan interacciones que pueden ser ligadas a habilidades del pensamiento computacional. De estas interacciones se puede observar una fuerte prevalencia del pensamiento algorítmico, la importancia del material en el intercambio de ideas y el abordaje de temas geométricos de manera autónoma por el estudiantado.

## Desarrollo

El pensamiento computacional comparte características con el lógico matemático, con la crucial diferencia que el primero está dirigido a utilizar una computadora, sujeto a las posibilidades de la computación. Respecto a ello, vemos que el razonamiento lógico matemático abarca habilidades como “1. Conceptualizar, generalizar y utilizar información. 2. Elaboración de modelos personales para resolver problemas complejos. 3. Habilidades para aplicar conocimientos y destrezas en situaciones nuevas” (Leiva, 2016, p. 211). Mientras en el pensamiento computacional refiere habilidades como la “abstracción, automatización, depuración, descomposición, generalización y pensamiento algorítmico” (Coordinación General @prende.mx, 2018, p.34). Podemos observar esta relación en la figura 1.

El presente documento sostiene que la programación informática puede abonar al razonamiento lógico matemático, si mantenemos el énfasis en la resolución de problemas y el diseño de algoritmos. No olvidemos que, en el ámbito escolar, utilizar “el ordenador debe permitir la construcción del conocimiento a través del aprender haciendo y del pensar sobre lo que se está haciendo” (Campos, 2013, p. 69).

Los algoritmos se consideran una noción clave para el desarrollo de la presente secuencia didáctica, es la manifestación de las semejanzas entre el razonamiento lógico matemático y el pensamiento computacional ya que el algoritmo es una “*secuencia ordenada de instrucciones, pasos o procesos que llevan a la solución de un determinado problema*” (López, 2009, p. 21-22). Como se observa en la figura 2, formular algoritmos moviliza saberes y habilidades lógicas para computar y comprobarlos. Para los alumnos, trabajar con algoritmos computacionales les permite transferir el aprendizaje de entornos de programación a otros dominios (Moreno-León, et al. 2016).

No pretendemos que los alumnos aprendan a programar al final de las sesiones de trabajo, sino que movilicen habilidades del pensamiento computacional. Por ello es que la atención está en la situación en la que los alumnos construyen su algoritmo (ejemplo de esa situación en la figura 3), más que en el *output* del mismo. No se busca evaluar las soluciones, sino observar

el razonamiento y las habilidades involucradas desde su primer encuentro con el problema, formulación de soluciones y optimización.

Para la construcción del material se consideró al entorno de programación *Scratch* dada su larga trayectoria en el trabajo de programación en las aulas. Además, se concluyó que era el más pertinente para llevar a una interfaz tangible gracias a ser un lenguaje de tipo visual. El lenguaje de *Scratch* tiene un área de trabajo donde se acomodan los bloques y otra donde ocurre lo descrito en el algoritmo. Así que se decidió mantener esta división fundamental y emergió la idea de trabajar con dos tableros. Un tablero para mostrar las acciones (*display*) y otro para programar.

Se estableció que el tablero de acción se delimitara mediante una cuadrícula, lo cual facilitó identificar las unidades de cada movimiento. Sin embargo, esto limitó las direcciones para girar a 90°, 180° y 270°. Este tablero es un cuadrado de 28 centímetros por lado, con una cuadrícula de 49 cuadrados para desplazarse. Los materiales del tablero consistieron en papel cascarón cubierto con cinta adhesiva transparente, además la figura que se desplaza en la cuadrícula es la mascota del programa *Scratch*. Estos dos son los únicos elementos permanentes del tablero de acción.

Existen dos tipos de casillas que pueden agregarse con plumones: los obstáculos y recompensas. Estos elementos forman parte de las situaciones planteadas y se destina una simbología para obstáculos y otra para recompensas. La ventaja de utilizar plumones es que los alumnos pueden agregar por sí mismos estos obstáculos, facilitando el trabajo guiado y la responsabilidad frente al propio aprendizaje. Como se observa en la [figura 4](#), los alumnos pueden utilizar ambos tableros para entender la situación

El tablero de programación consiste en dos rectángulos de papel cascarón sobrepuestos y cubiertos de fieltro. De acuerdo a la forma de los bloques, se busca que el tablero se use verticalmente, acomodando los bloques debajo del anterior tal como se ve en la [figura 5](#). Los bloques se asemejan a piezas de rompecabezas con un trapecio que sobresale y embona en la parte de arriba. A cada bloque se le agregó un trozo de velcro en la parte de atrás y cinta *masking* para una superficie donde se pueda escribir.

En esta secuencia didáctica, se trabajó con 4 categorías de bloques. De esta manera, tenemos bloques de eventos (amarillo), movimientos (azul), control (naranja) y sonido (morado). En el caso de las categorías de eventos y sonidos, solamente utilizamos el bloque de “Iniciar con” (ver la [figura 6](#)) y el de sonido para decir una frase. La distribución del material didáctico consistió en 3 parejas y 7 tercias en un salón de 27 estudiantes. Para la recolección de datos se decidió elegir un equipo que fungió como grupo focal para el análisis de sus interacciones

La progresión temática de esta situación didáctica se planteó con base en 6 habilidades del pensamiento computacional: pensamiento algorítmico, generalización, descomposición, depuración, automatización y abstracción. A partir de los conceptos de pensamiento computacional (secuencia, *loops* y condicionales), buscamos elementos en común y agrupamos

las habilidades en los primeros tres módulos: Algoritmos, Condiciones y Optimización, y Ciclos (*loops*).

#### Módulo 1 Algoritmos

- Pensamiento algorítmico y abstracción

#### Módulo 2 Condiciones y optimización

- Depuración y descomposición

#### Módulo 3 Ciclos

- Automatización y generalización

#### Módulo 4 Figuras geométricas

- Movilización de todas las habilidades

Para categorizar las interacciones, utilizamos algunos de los ejes de los objetivos de aprendizaje propuestos por la Coordinación General @prende.mx (2018) para agrupar las habilidades observadas y perfilar los resultados de este trabajo. De esta forma, presentamos la tabla 1 de frecuencias con los ejes de cada objetivo de aprendizaje, las habilidades del pensamiento computacional, la frecuencia observada de esa habilidad y fragmentos de la transcripción ejemplificando esas habilidades.

Se puede observar una fuerte prevalencia del pensamiento algorítmico. Esto, dado a que se promueve la acción del estudiantado, la cual puede ser probada y retroalimentada de manera autónoma a través del material. Se presenta así la figura 7, una gráfica de barras con las frecuencias de cada habilidad durante las sesiones de trabajo. De igual manera, agrupamos las barras según los ejes de los objetivos de aprendizaje, representados por los colores crema, azul y fucsia, de Definición de problemas, Algoritmos y desarrollo, y el de Colaboración y creatividad, que aparecen en la Propuesta del Marco de Referencia del Pensamiento Computacional (Coordinación General @prende.mx, 2018). De esta forma, se abarcan tres ejes de los objetivos, y aunque en la definición de problemas no hubo frecuencias tan altas, sí hubo una manifestación constante de este eje.

## Conclusiones

Durante este trabajo se indagó en las interacciones entre el razonamiento humano y las herramientas computacionales ya que, como dice Jeannette Wing (2008, como se citó en Coordinación General @prende.mx, 2018) “la computadora puede ser un humano o una máquina”. De esta manera, no todo uso de la computadora moviliza el pensamiento

computacional, ni el pensamiento computacional necesita en todo momento el uso de la misma. El pensamiento computacional es la sistematización del razonamiento humano para una computadora. Y es a partir de esto donde se fundamenta la propuesta didáctica de este trabajo. Asemejar el material didáctico y las actividades a un ambiente computacional fue crucial para trabajar este tipo de pensamiento desde las limitaciones.

Durante el desarrollo de la propuesta didáctica, fue posible trazar un paralelismo entre las habilidades del pensamiento computacional y las habilidades de razonamiento lógico matemático descrito por Leiva (2016). “Conceptualizar, generalizar y utilizar información” se relaciona con las habilidades del pensamiento computacional de Abstracción y Generalización, las cuales tuvieron una frecuencia de 19 y 8 ocasiones respectivamente como se puede observar en la Tabla 1 de la sección de resultados. Mientras tanto, la elaboración de modelos para resolver un problema y su aplicación en situaciones nuevas (Leiva, 2016) se entrelazaron durante el módulo final de las figuras geométricas. Además, es posible observar cómo desde las interacciones, se fomentan las fases de resolución de un problema reportadas por Guzmán y Moreira (2014):

1. Análisis inicial del problema.
2. Determinación de la vía de solución.
3. Determinación de la forma de introducir la respuesta a la computadora.
4. Realización o ejecución de la vía de solución.
5. Valoración del control del resultado.

Se considera que lo anterior se debe a que el material presenta una realidad con límites bien establecidos (casilla de inicio, de objetivo y casillas especiales) y la única forma de llevarlo a la acción es mediante el acomodo de los bloques para formar un algoritmo; es por ello que la vía de solución sólo puede computarse de una forma. Esta secuencia permite observar de mejor manera la afirmación de que son los alumnos las computadoras que ejecutan y evalúan el algoritmo.

Aunque el desarrollo de las actividades puede parecer más lento que con una computadora, es necesario observar que son los alumnos quienes evalúan los algoritmos de su equipo. Por lo tanto, son los alumnos quienes realizan la acción de computar. Una posibilidad futura sería prescindir de la cuadrícula del tablero de acción para permitir desplazamientos con mayor libertad, para realizar algoritmos con un enfoque más creativo o artístico.

A pesar de que no trabajamos en tecnologías digitales, sí hubo una interacción con algoritmos y una terminal informática (física), el material didáctico correlacionó la colaboración y el pensamiento algorítmico. Los alumnos fueron construyendo algoritmos y conocimiento nuevo en colaboración con sus pares. Es un aspecto importante que trasciende al pensamiento computacional para abonar a los pilares de la formación, los alumnos se acercan, proponen, defienden sus ideas mientras manipulan el material y razonan.

En este sentido, esta metodología es conveniente para un docente frente a grupo, ya que fomenta la autonomía. Cada vez hay más interés por parte de los docentes en acercarse a los alumnos a la informática desde un enfoque transdisciplinario y complejo. Sin embargo, además de las dificultades materiales, aquellos neófitos a menudo se encuentran con un panorama amplio y confuso para aprender herramientas de programación. Es por ello que la comunidad docente no puede ignorar este interés, es necesario dar a conocer y alimentarlo con herramientas más intuitivas y que inviten a los docentes frente a grupo, aún con poca experiencia, a un estudio autodidacta (Wong, et al., 2015).

Así, esta propuesta considera que el razonamiento humano es capaz de realizar por sí solo (con menor velocidad) la mayoría de las habilidades del pensamiento computacional; la computadora no es indispensable, ni un sustituto del razonamiento, es una herramienta para potenciar sus habilidades. En este sentido, la computadora necesita el input humano, y es ahí donde debemos concentrar los esfuerzos educativos, en afinar la capacidad y el razonamiento humano para aprovechar la computación de una manera fructífera ya que esta será la constante a pesar de los vertiginosos cambios e innovaciones tecnológicas. Formar alumnos conscientes, críticos y que movilicen el pensamiento computacional es una forma de evitar la adopción pasiva (y a veces nociva) de la tecnología. Las actividades aquí presentadas fueron desafíos para los alumnos, siendo ellos los que computan los datos y comprueban los algoritmos.

Alejarse de la actitud consumidora hacia una creadora, partiendo desde las condiciones, requiere propuestas inusuales, pero también la comprensión clara de los fundamentos y teorías de lo que se trata. En este sentido, esta tesis no es un sustituto para el trabajo con la tecnología, sino una alternativa que nace de las limitaciones materiales de escuelas rurales, indígenas o en contextos marginados. Esta metodología es un esfuerzo estructurado cuyas posibilidades podrían superar los alcances de las actividades descritas en capítulos anteriores. Al ser un material didáctico novedoso, no se ha agotado su potencial, sin embargo, como se demostró en los resultados, ha logrado reproducir muchas de las habilidades que se creían exclusivas del trabajo con computadora. Este material didáctico favorece su reproducibilidad, así como la exploración libre y creativa por parte de los docentes.

Educar desde las limitaciones permite repensar supuestos aparentemente obvios, y aunque implica mayor dificultad para docentes y alumnos, es precisamente por ello que permite cimentar sus aspectos fundamentales. Quizá los productos y resultados iniciales sean más discretos que aquellos con una computadora, pero con la orientación adecuada los alumnos comprendieron el funcionamiento de la programación informática, con la oportunidad de trasladar esos aprendizajes a situaciones diversas. El objetivo no debe ser formar operadores de cierto lenguaje de programación, memorizando su sintaxis; sino construir el pensamiento computacional aplicable para distintas áreas de su vida, donde los algoritmos sean la manifestación de un razonamiento más profundo.

Tablas y figuras

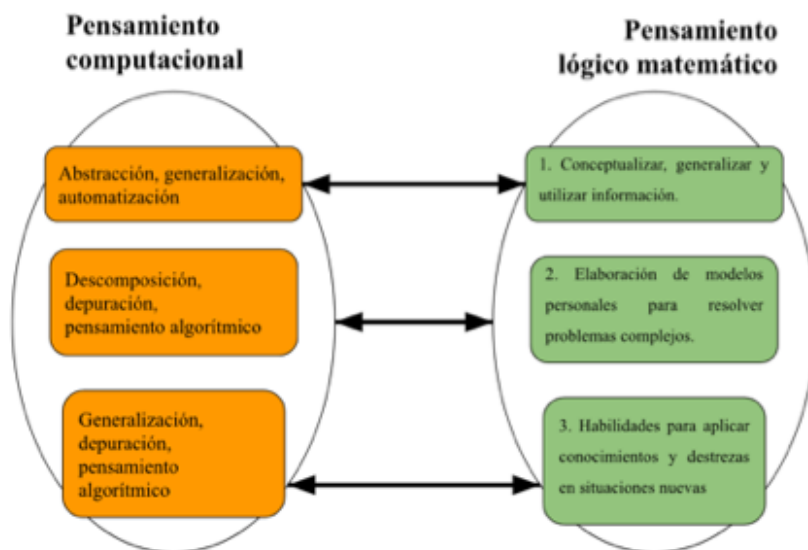
**Tabla 1. Tabla de resultados**

Eje	Habilidades	Ejemplo	Frecuencia
Definición de problemas	Abstracción	<p>“(Discutiendo cómo iniciar el algoritmo)            F: Entonces debería decir “al caminar”, ¿no?            D: (dudosa) No...significa que empieza, empieza cuando tu caminas?            L: Sí, ese es el inicio. ¿Ustedes van a caminar? Es lo que <b>prende el robot</b>, lo hicimos ayer”</p> <p>“L: ¿Y si lo cambiamos mejor? (rotar scratch, cambiar la dirección hacia donde camina)            N: Sí, a la izquierda.            D: Sí <b>giramos</b> a la izquierda sería así.            N: Entonces no”</p> <p>“D: Eso se repite 6 veces. Cuando quede así, se empieza a repetir * ejemplifica la orientación final.* Y mover hacia la derecha 3 cuadros.”</p>	#19
Algoritmos y desarrollo	Automatización		
D. de problemas	Depuración	<p>“F: Ah, chale ¿Y cómo le hacemos para llegar hasta allá? ¿Otra vez lo hacemos, no?            D: (emocionada) ¡Si!            F: No borres este, está bien            L: Ya sé, hay moverlo hasta la orilla”</p> <p>“A3: *comprobando otro algoritmo* Aquí...hacia acá            Maestro: gira 90° cuatro veces....            A3: No, no da.            A1: ¿Se puede mover tres veces?            Maestro: Sí            A3: ¿Así?            A1: Ya ves, te lo dije”</p> <p>“X: Gira 90° a la izquierda, y avanza 8            D: más de 8 porque falta un giro”</p>	#17
D. de problemas	Descomposición	<p>“D: Obvio se puede            N: son 4 casillas            L: ¿4 casillas? okay. Girar 90°. ¿Por qué no así y así? *lo ejemplifica* Alargarlo y luego seguir            N: No creo”</p>	#17
Algoritmos y desarrollo	Generalización	<p>“Maestro: Mueve 4, gira, se mueve cuatro.            E5: ¿Ves? Es como el mío            E1: Pero está mal colocado”</p> <p>“A1: Creo que era como yo dije            A2: No, no se puede hacia otros lados (gitar más de 180°”</p> <p>“F: No se puede, está muy difícil.            [...]            X: Sí tiene solución”</p>	#8

Algoritmos y desarrollo	Pensamiento algorítmico	<p>“D: ¿Cómo empezamos? Aplaudir, no L: silbar. De ahí girar 180° o 90°?          N: 180° es como un búho”</p> <p>“E4: No, ese no se usa, ¿Cómo lo movemos?          E1: Se va a un lado, se va a otro          E4: No, con los pasos, no se puede girar y mover”</p>	#30
Creatividad y colaboración	Colaboración	<p>“E3: Dos pasos          E2: No, era uno          E4: A ver, va por aquí *empieza a contar*          E2: 3 pasos...otro paso          E3: ¿Por qué otro?          E2: Mira, son 4          E3: Contando el punto de partida (E2 corrige)”</p> <p>“F: ¿Y así se quedaría? Pon 90°          X: Sí es cierto, se me olvidó          N: Ahora mover, ¿no?          F: Sí”</p>	#32

Elaboración propia

**Figura 1. Relación entre pensamiento computacional y pensamiento lógico matemático**

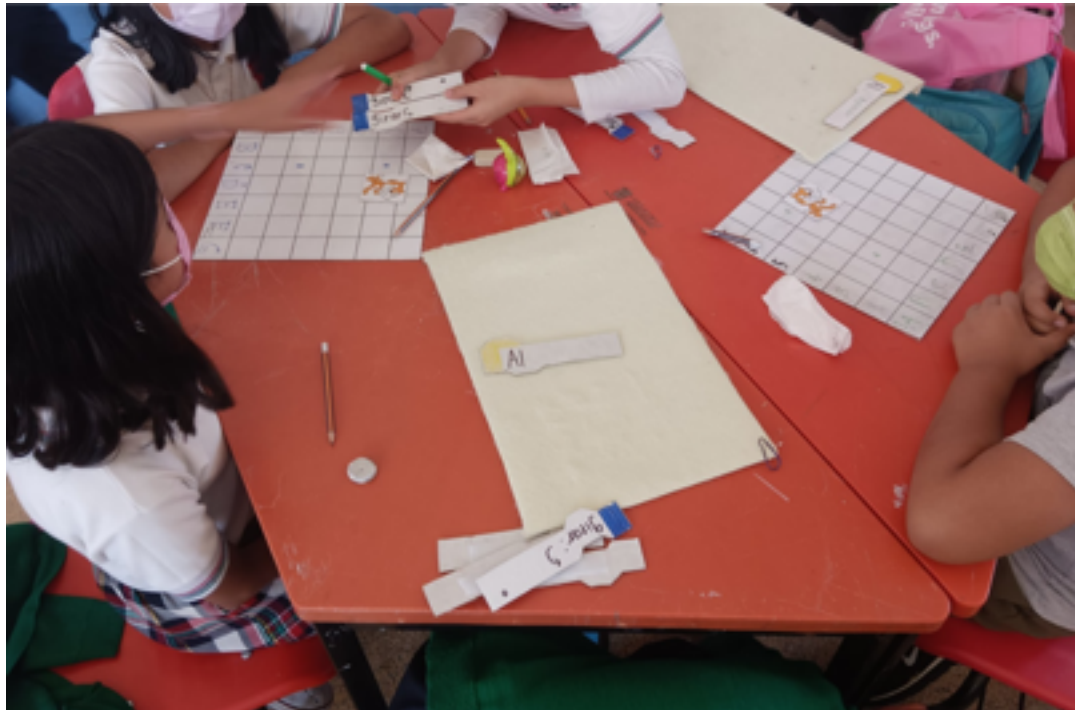




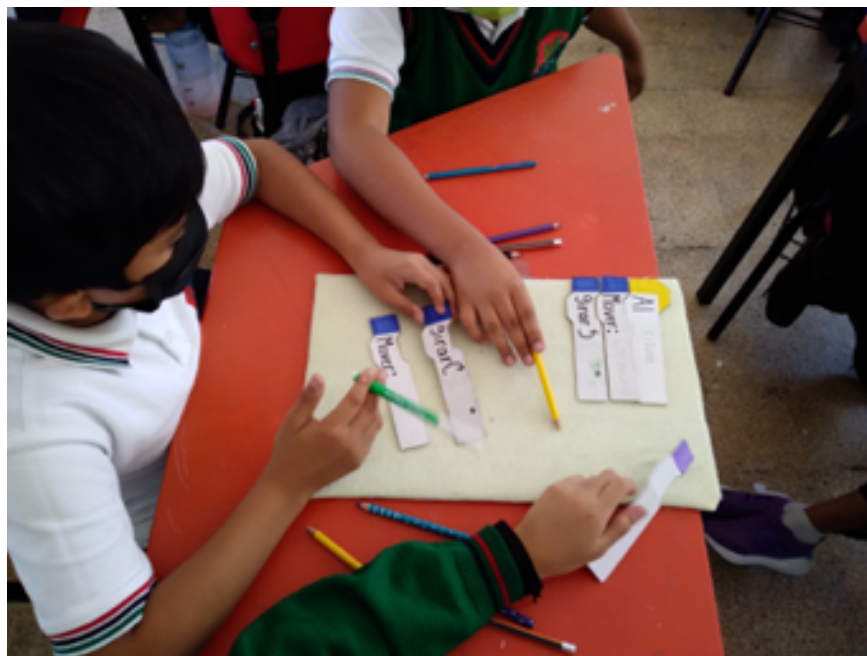
**Figura 2. Los algoritmos como manifestación de razonamiento lógico matemático**



**Figura 3. Programar con materiales manipulables lo vuelve accesible para más contextos**



**Figura 4. Las soluciones pueden surgir desde el código o manipulando el tablero de acción**



**Figura 5. El diseño de algoritmos por equipo fomenta el razonamiento y la colaboración hacia un fin**

**Figura 6. Los bloques embonan para mantener la sensación de secuencia**



**Figura 7: Incidencia de las habilidades del pensamiento computacional durante las sesiones de trabajo**



## Referencias

- Campos, F. (2013). Paulo Freire e Seymour Papert: educação, tecnologias e análise do discurso. Curitiba: CRV.
- Coordinación General @prende.mx (2018). Propuesta del Marco de Referencia del Pensamiento Computacional en educación básica. En Gobierno de México. Recuperado de <https://www.gob.mx/aprendemx/documentos/propuesta-del-marco-de-referencia-del-pensamiento-computacional-en-educacion-basica>
- Guzmán, O., Moreira, Y. (2014). La resolución de problemas geométricos en Matemática utilizando la computadora. *EduSol*, 14(46), 1-8.
- Leiva, F. (2016). ABP como estrategia para desarrollar el razonamiento lógico matemático en alumnos de educación secundaria. *Sophia*, colección de Filosofía de la Educación. 21(2). 209-224.
- López, J. (2009). ALGORITMOS Y PROGRAMACIÓN (Guía para docentes). Bogotá: Fundación Gabriel Piedrahita Uribe.
- Moreno-León, J., Robles, G., Román-González, M. (2016). Code to learn: Where does it belong in the K-12 curriculum?. *Journal of Information Technology Education*. 15(1). 283-303. Recuperado de <http://www.informingscience.org/Publications/3521>
- Piaget, J. (1975). *El desarrollo del pensamiento*. Buenos Aires: Paidós.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of The Royal Society*, 366, 3717–3725. Recuperado de [doi:10.1098/rsta.2008.0118](https://doi.org/10.1098/rsta.2008.0118)

Wong, G. W., Cheung, H., Ching, E. C. C., & Huen, J. (2015). School perceptions of coding education in K-12: A large scale quantitative study to inform innovative practices. <https://doi.org/10.1109/tale.2015.7386007>