

Experiencias y estrategias educativas con TIC para el desarrollo del pensamiento computacional en Iberoamérica

Iberoamerican educational strategies and experiences based on Ict, in order to develop computational thinking

Maira Isbeth Sarmiento Bolívar¹

Recibido: 5-11-2018

Aprobado: 11-3-2019

Resumen

Este artículo presenta los resultados de un análisis realizado a diversas experiencias y estrategias educativas. El objetivo de la investigación se centró en la búsqueda de iniciativas que involucran las Tecnologías de Información y la Comunicación (TIC) en el desarrollo del pensamiento computacional alrededor de Iberoamérica. Primero, se revisa las definiciones dadas por expertos conceptualizando el término pensamiento computacional, luego se contrastan para identificar aspectos comunes y particulares, y de esta manera elegir un concepto como base para esta investigación; segundo, se examina las competencias relacionadas con el pensamiento computacional, las cuales fueron contempladas en el análisis de las propuestas. Posteriormente, se toman las experiencias encontradas, relacionadas con la enseñanza de la programación o temas afines, y se analizan según parámetros previamente definidos que permitieron describirlas uniformemente.

El análisis arrojó como resultado que la mayor cantidad de experiencias encontradas se desarrollaron en Argentina, Chile y España. También que el área de aplicación es institucional, en un ámbito universitario y de secundaria por medio de propuestas educativas de extensión y articulación entre las instituciones; asimismo, se conoce que las herramientas y materiales utilizados son en su mayoría de desarrollo propio y se promueve el uso de herramientas libres y gratuitas, el enfoque de las experiencias busca solucionar problemas planteados desde diversas áreas. Este análisis reveló que la mayoría de las propuestas logran desarrollar la totalidad de los conocimientos, habilidades y actitudes tomados como base, entre ellas: solución de problemas, abstracción de información, diseño de algoritmos, modularización, simulación, implementación, generalización y transferencia, persistencia y trabajo en equipo.

Palabras claves: Tecnologías de Información y la Comunicación (TIC), experiencias educativas, competencias tecnológicas, pensamiento computacional, enseñanza de la programación.

Abstract

This article contains the results of an analysis based on experiences and educational strategies. The aim of this research is focused on the search for initiatives that involved Information and Communication Technologies (ICT) in the development of “computational thinking” in Iberoamerica. First, the definitions given by experts are reviewed, conceptualizing the term “computational thinking”. Then, they are contrasted to identify common and particular aspects in order to choose a concept as the basis for this research. Second, the competences related to computational thinking are examined, which are discussed in the analysis of the proposals. Subsequently, the experiences found, regarding the teaching of programming or related topics, are analyzed according to predefined parameters that allowed describing them uniformly.

¹ Docente e investigadora, especialista en Tecnología Informática Aplicada en Educación por la Universidad Nacional de la Plata, Argentina.

The analysis showed that most of the experiences found took place in Argentina, Chile and Spain. It was also found that the area of application is institutional, both at university and secondary level, through educational proposals of outreach and articulation between the institutions. Also, it is known that the tools and materials used are mostly self-developed and the use of non-paid and free material is promoted. The approach of the experiences seeks to solve problems from diverse areas. This analysis revealed that most of the proposals manage to develop all the knowledge, skills and attitudes taken as a basis, including: problem solving, information abstraction, algorithm design, modularization, simulation, implementation, generalization and transfer, persistence and teamwork.

Key words: Information and Communication Technologies (ICT), educational experiences, technological competences, computational thinking, teaching of programming.

I. Introducción

Desde hace algunos años se ha incrementado la generación de importantes procesos de integración de las TIC en la educación, donde programas públicos de diversos países proporcionan equipos informáticos (PC, *notebook*, *netbook*, tablet, entre otros) a las instituciones educativas y generan propuestas de formación a docentes para un máximo aprovechamiento de estas, con el fin de realizar los cambios pedagógicos necesarios que garanticen que esta inclusión impacte positivamente en la manera de enseñar y aprender. Los procesos de integración TIC que han impulsado estas políticas públicas se vienen gestando alrededor de Iberoamérica y el objetivo que persiguen es el desarrollo de proyectos educativos de calidad, los cuales se encuentran en distintas etapas, algunos con mayores avances y otros en etapas finales (Lugo *et al.*, 2010).

Estos programas buscan generar un impacto social, haciendo que más estudiantes cuenten con acceso a este tipo de tecnologías, reduciendo la brecha digital existente, cuestión que se ha convertido en un desafío en los procesos de integración de las TIC. Las iniciativas generadas reconocen la importancia que tiene el desarrollo de competencias y habilidades en el manejo de las herramientas tecnológicas denominadas *competencias del siglo XXI* por la UNESCO (SITEAL, 2014). A raíz de la importancia de las propuestas que promueven la integración de las TIC en los procesos de enseñanza y aprendizaje, se generó como Trabajo Final Integrador para optar por el título de Especialista en Tecnología Informática Aplicada en Educación de la Universidad Nacional de La Plata (UNLP), Buenos Aires - Argentina,

la investigación “Análisis de experiencias y estrategias educativas con TIC para el desarrollo del pensamiento computacional en estudiantes de secundaria y primeros años de universidad en Iberoamérica”, cuyo proceso y resultados se presentan en este artículo.

II. Marco teórico

A continuación se definen las bases conceptuales utilizadas en esta investigación; primero enfocado en el concepto de pensamiento computacional y sus diversas acepciones, segundo, centrado en el análisis comparativo de estos conceptos, tercero, dando a conocer el concepto designado para trabajar en esta investigación y finalmente, la revisión de los conocimientos, habilidades y actitudes relacionados con el pensamiento computacional.

2.1 Acerca del pensamiento computacional

Con respecto al pensamiento computacional, se encontró que son varios los profesionales en el campo de la educación, la informática y las ciencias que han brindado un concepto sobre lo que es y cómo podría desarrollarse este tipo de pensamiento, aunque a pesar de la cantidad de referentes, no se ha llegado a un acuerdo sobre una definición precisa, pues cada uno de estos aportes han sido generados desde la perspectiva del área en la cual los expertos se desenvuelven. El término pensamiento computacional fue utilizado la primera vez por Jeannette Wing en el año 2006, quien se desempeña como vicepresidente corporativo de Microsoft Research y como profesora del Computer Science Department de la Carnegie Mellon University, y lo definió como: la habilidad

de dar respuesta a problemas haciendo uso de herramientas computacionales. En el 2010 Wing junto con Jan Cuny, de la Fundación Nacional de Ciencias, y Larry Snyder, de la Universidad de Washington (como se citó en Wing, 2011), dieron a conocer el siguiente concepto:

Pensamiento computacional es el proceso de pensamiento donde están involucradas la formulación de los problemas y sus soluciones, donde las soluciones están representadas en una forma que se pueden llevar a cabo con eficacia por un agente de procesamiento de información (Wings, 2011, p.1).

Luego del acercamiento que hizo el equipo de Jeannette Wing al concepto, se han presentado aportes sobre el tema, generados por investigaciones y seminarios realizados desde distintos puntos de América del Norte. En el año 2008 la National Science Foundation (NSF), en conjunto con el National Research Council (NRC), realizó dos talleres con la idea de explorar la naturaleza del pensamiento computacional y sus implicaciones educativas, y de estos talleres se generaron dos informes que fueron presentados en los años 2010 y 2011.

En *Report of a Workshop on The Scope and Nature of Computational Thinking*, el pensamiento computacional se definió de las siguientes maneras:

- David Moursund dice que el pensamiento computacional está relacionado con el pensamiento procedimental, tomando la definición hecha por Seymour Papert. Este tipo de pensamiento incluye pruebas y procedimientos de depuración, junto con un procedimiento paso a paso de instrucciones que pueden ser interpretadas por un ordenador (National Research Council, 2011). El pensamiento computacional también se percibe de alguna manera como la forma en que hacemos cálculos computacionalmente. Se ve, por ejemplo, como una ayuda para el modelado, la representación y la resolución de problemas matemáticos (Moursund, 2007).

- Peter Denning explica que el pensamiento computacional forma parte de la informática y por medio de ella se estudia el proceso de la información. Según Denning, el pensamiento computacional es una de las prácticas claves de la informática, pero no es la única práctica importante en el manejo de la información y tampoco se puede decir que sea la base de la ciencia de la computación (Denning, 2009). Define el pensamiento computacional como un método de aproximación a la solución de problemas representando el problema como un proceso de información con solución algorítmica (Denning, 2010).

En el segundo informe de la NSF, SRC cuyo título fue *Report of a Workshop of Pedagogical Aspects of Computational Thinking*, se dieron a conocer los siguientes conceptos (Linn *et al.*, 2010):

- Para Robert Panoff, el pensamiento computacional es un enfoque que sirve para desarrollar habilidades metacognitivas en los estudiantes quienes adquieren la capacidad de descomponer los problemas en pequeñas partes para así darle solución por medio de una computadora.
- Yasmin Kafai centra el pensamiento computacional hacia el cálculo matemático como medio para dar solución a problemas reales, para ello el estudiante debe trabajar en un proceso cíclico de usar-modificar-crear aplicado a cualquier disciplina, este ciclo facilita el aprendizaje de la computación basado en análisis de datos, visualización y diseño de juegos. Según Kafai, los soportes de programación desarrollan el pensamiento computacional, ayudando a aprender sobre resolución de problemas y estrategias de diseño como la modularización y el diseño iterativo (Kafai *et al.*, 2009).

La Sociedad Internacional para la Tecnología en Educación (ISTE) y la Asociación de Docentes en Ciencias de la Computación (CSTA) dieron a conocer una definición operativa acerca del pensamiento computacional para educación escolar:

El pensamiento computacional PC es un método para resolver problemas de manera que puedan ser implementados con una computadora. Los estudiantes no son solo usuarios de la herramienta sino constructores de ellas; utilizando un conjunto de conceptos, tales como la abstracción, la recursividad y la iteración, para procesar y analizar los datos y crear artefactos reales y virtuales. PC es una metodología de resolución de problemas que se puede automatizar, transferir y aplicar a través de todas las clases. El poder del pensamiento computacional es que se aplica a cualquier otro tipo de razonamiento. Permite realizar todo tipo de cosas: física cuántica, biología avanzada, sistemas computarizados humanos, el desarrollo de herramientas de computación útiles (Barr y Stephenson, 2011, p.10).

Por otro lado, el Centro de Pensamiento Computacional en la Universidad Carnegie Mellon define pensamiento computacional como la manera de solucionar problemas basándose en los conceptos fundamentales de la informática, convirtiéndose en parte fundamental de la forma de pensar y entender el mundo. Pensamiento computacional significa dar soluciones eficientes, justas, seguras, económicas y sociales, utilizando diferentes

niveles de abstracción y pensamiento algorítmico para aplicar conceptos matemáticos (Center for Computational Thinking, 2012).

En el artículo escrito por los profesores de la Universidad de Alicante (España), Patricia Compañ Rosique, Rosana Satorre Cuerda, Faraón Llorens Largo y Rafael Molina Carmona, se expresa que el pensamiento computacional se basa en resolver problemas haciendo uso de conceptos fundamentales de la informática y que la esencia del pensamiento computacional es pensar como lo haría un científico informático cuando se enfrenta a un problema. Para estos docentes, realizar cursos de introducción a la programación de ordenadores es el vehículo perfecto para desarrollar habilidades de pensamiento computacional, ya que implica la resolución de problemas mediante el uso de conceptos informáticos (Compañ Rosique *et al.*, 2015).

La Tabla 1 muestra los términos señalados por Jeannette M. Wing, Jan Cuny y Larry Snyder en la definición sobre pensamiento computacional presentada en el año 2010, estos elementos se utilizan para realizar un análisis comparativo. Para ello, se tomó la conceptualización señalada por cada uno de los autores y fueron contrastadas, identificando aquellos aspectos comunes entre las distintas opiniones y detallando particularidades en la columna “otros”.

Tabla 1. Terminología relacionada con el pensamiento computacional

	Procesos de pensamiento	Problemas y soluciones	Procesamiento de Información	Herramientas, técnicas y métodos	Computación/ tecnología	Pensamiento matemático	Aplicación en diversas áreas	Otros
Wing, Cuny y Sneider	X	X	X	X	X	X	X	
David Moursund	X		X		X			Depuración Instrucciones
Peter Denning		X	X			X		
Robert Panoff	X	X			X	X		
Yasmin Kafai		X	X	X	X	X	X	Soportes de programación
ISTE-CSTA		X	X	X	X		X	
Carnegie Mellon		X	X	X	X	X	X	
Faraón <i>et al</i>		X	X		X			

Fuente: Elaboración propia.

Al realizar el análisis y contrastar las diversas concepciones, se llega a la decisión de elegir la definición realizada por Jeannette M. Wing, junto con Jan Cuny y Larry Sneider, como base para realizar el análisis de diversas propuestas de integración de las TIC. Esto debido a que:

- Fue el primer referente sobre el tema pensamiento computacional.
- La definición ha sido tomada como base de importantes propuestas educativas en el campo de la tecnología e informática, como por ejemplo: de K-12 Estándares para las Ciencias de la Computación y la cartilla *Computational thinking leadership toolkit* realizada por la Asociación de Maestros de Ciencias de la Computación (CSTA) y la *International Society for Technology in Education* (ISTE).
- La mayoría de los términos utilizados en esta definición fueron tomados por otros investigadores para profundizar sobre el tema.

2.2 Características relacionadas con el pensamiento computacional

Después de analizar y elegir el concepto de pensamiento computacional utilizado en esta investigación, se revisan las características relacionadas con este, descritas por profesionales en el área educativa.

Para Wing, el pensamiento computacional significa ser capaz de:

- Comprender qué aspectos de un problema son susceptibles de cómputo.
- Entender las limitaciones y el poder de las herramientas y técnicas computacionales.
- Aplicar o adaptar una herramienta computacional o técnica a un nuevo uso.
- Reconocer la oportunidad de utilizar la computación de una manera nueva.

- Aplicar estrategias de cálculo tales como divide y vencerás en cualquier dominio (Wing, 2010, p. 50).

Según la fundación Google for Education², los conceptos relacionados con el pensamiento computacional que deben conocer los estudiantes son:

Abstracción: La abstracción es la identificación y extracción de información relevante para definir la idea principal.

Diseño de algoritmos: Es la creación de una serie ordenada de instrucciones para la solución de problemas similares, o para hacer una tarea.

Automatización: Utilizar ordenadores o máquinas para hacer las tareas repetitivas.

Recolección de datos: Recopilación de información.

Análisis de datos: Es dar sentido a los datos mediante la búsqueda de patrones o el desarrollo de conocimientos.

Representación de datos: Es representar y organizar datos en gráficos, tablas apropiadas, palabras o imágenes.

Descomposición: Se descomponen datos, procesos o problemas en partes más pequeñas y manejables.

Paralelización: Es el proceso simultáneo para realizar las tareas más pequeñas de una tarea más amplia y llegar de manera más eficiente a un objetivo común.

Generalización de patrones: Es la creación de modelos, reglas, principios y teorías de los patrones observados para poner a prueba los resultados predichos.

² Información extraída del sitio web <https://www.google.com/edu/resources/programs/exploring-computational-thinking/index.html#!ct:overview>

Reconocimiento de patrones: Es la observación de patrones, tendencias y regularidades en los datos.

Simulación: Se desarrollan modelos para imitar los procesos del mundo real.

La ISTE y la CSTA, basándose en la definición propuesta por Jeannette M. Wing, suministraron un marco de referencia donde se brindan las características y actitudes con respecto del pensamiento computacional. Según el documento Definición Operativa de pensamiento computacional para Educación Básica y Media (K-12) (ISTE y CSTA, 2011, p.13), algunas de las características son:

- Formular problemas de manera que permitan usar computadoras y otras herramientas para solucionarlos.
- Organizar y analizar datos de manera lógica.
- Representar datos mediante abstracciones, como modelos y simulaciones.
- Automatizar soluciones mediante pensamiento algorítmico (una serie de pasos ordenados).
- Identificar, analizar e implementar posibles soluciones con el objeto de encontrar la combinación de pasos y recursos más eficiente y efectiva.
- Generalizar y transferir ese proceso de solución de problemas a una gran diversidad de estos.

Estas características están apoyadas por disposiciones y actitudes como:

- Confianza en el manejo de la complejidad.
- Persistencia al trabajar con problemas difíciles.
- Tolerancia a la ambigüedad.
- Habilidad para lidiar con problemas no estructurados (*open-ended*).
- Habilidad para comunicarse y trabajar con otros para alcanzar una meta o solución común.

En los reportes de los talleres realizados por la National Research Council, *Report of a Workshop of Pedagogical Aspects of Computational Thinking*, se considera que el pensamiento computacional busca desarrollar las competencias digitales, la lógica, la programación, el lenguaje, el procesamiento en paralelo, la reformulación de problemas complejos, la verificación de modelos, la automatización de procesos, el diseño de sistemas de diseño, la aproximación a soluciones y la resolución de problemas; a través del desarrollo de videojuegos, la construcción de robots y la experimentación con simulaciones, la planificación, las pruebas, la prevención de errores y la modularización, entre otras. Estas habilidades pueden ser desarrolladas en diferentes disciplinas como la ingeniería, la ciencia, el medio ambiente, los negocios, el periodismo, la geografía, la química, entre otras, o diferentes contextos, como los juegos y la vida diaria (National Research Council, 2010).

Por otro lado, Peter Kemp (2014), perteneciente a la Naace, National Association of Advisors for Computers in Education, y CAS, Computing at School de Reino Unido, se refiere a las competencias del pensamiento computacional como el proceso de reconocimiento de herramientas y técnicas utilizadas para entender y razonar sistemas naturales y procesos artificiales existentes en el mundo que nos rodea.

El pensamiento computacional consiste en:

- Descomponer el problema en sus componentes, cada uno de los cuales se aborda y analiza individualmente.
- Reconocer patrones para buscar similitudes en comportamientos y estados del sistema que se está modelando.
- Abstractar de manera que se utilicen los detalles absolutamente necesarios para el funcionamiento del sistema.
- Generalizar patrones que permitan definir conceptos en forma simple y poder utilizar esta definición para todas las instancias del concepto.
- Diseñar algoritmos siguiendo pasos precisos para la solución del problema.

Para Valerie Barr y Chris Stephenson (2011, p.52), los conceptos básicos de capacidades, disposiciones, predisposiciones y cultura del aula para el pensamiento computacional son:

Capacidades

- Diseñar soluciones a los problemas por medio de la abstracción, automatización, creación de algoritmos, recopilación y análisis de datos.
- Implementar los diseños.
- Prueba y depuración.
- Modelar, correr las simulaciones y hacer un análisis del sistema.
- Reflexionar sobre la práctica y la comunicación.
- Utilizar vocabulario específico del tema.
- Reconocer y moverse entre los niveles de abstracciones.
- La innovación, la exploración y la creatividad a través de disciplinas.
- Resolver problemas de grupo.
- Emplear diversas estrategias de aprendizaje.

Disposiciones y predisposiciones, área de valores, motivaciones, sentimientos, estereotipos y actitudes, aplicables al pensamiento computacional.

- Confianza en el trato con la complejidad.
- Persistencia en el trabajo con problemas culturales.
- Capacidad de manejar la ambigüedad.
- Hacer frente a los problemas de composición abierta.
- Dejar a un lado las diferencias para trabajar con otros para lograr una meta o solución común.
- Conocer las propias fortalezas y debilidades cuando se trabaja con otros.

La cultura del aula

- Mayor uso de vocabulario computacional en estudiantes y docentes para describir los problemas y sus soluciones.
- Aceptación de soluciones con intentos fallidos, reconociendo el fracaso como parte del camino hacia un resultado exitoso.
- Trabajo en equipo.

A partir de las definiciones de los autores y las organizaciones anteriormente nombradas, se muestra en la Tabla 2 una compilación de los conocimientos, habilidades y actitudes relacionados con el pensamiento computacional.

Tabla 2. Conocimientos, habilidades y actitudes relacionados con el pensamiento computacional

Pensamiento computacional	Actitudes
<p>Conocimientos y habilidades</p> <p>Formular problemas posibles de solución mediante una computadora.</p> <p>Abstraer la información relevante para reconocer los datos principales.</p> <p>Diseñar algoritmos para llegar de manera ordenada a la solución.</p> <p>Automatizar tareas repetitivas para optimizar tiempos.</p> <p>Recolectar, organizar y analizar datos de manera lógica.</p> <p>Representar datos de manera adecuada y clara.</p> <p>Descomponer datos, procesos (modularizar) o problemas en partes pequeñas y manejables.</p> <p>Trabajar tareas paralelamente como medio para cumplir el objetivo común de manera efectiva.</p> <p>Generar patrones creando modelos, reglas, principios y teorías de los patrones observados para poner a prueba los resultados predichos.</p> <p>Reconocer patrones observando tendencias y regularidad de datos.</p> <p>Simular modelos de las posibles soluciones antes de ponerlos a prueba.</p> <p>Implementar los modelos para saber si la solución propuesta fue eficiente y efectiva.</p> <p>Generalizar y transferir ese proceso de solución de problemas a una gran diversidad de estos.</p> <p>Utilizar vocabulario computacional.</p>	<p>Confianza en el manejo de la complejidad.</p> <p>Persistencia al trabajar con problemas difíciles.</p> <p>Tolerancia a la ambigüedad.</p> <p>Habilidad para lidiar con problemas no estructurados (<i>open-ended</i>).</p> <p>Habilidad para comunicarse y trabajar con otros para alcanzar una meta o solución común.</p>

Fuente: Elaboración propia.

De la misma manera en que se encuentran diversos conceptos relacionados con el término pensamiento computacional, se identifican variedad de conocimientos, habilidades y actitudes descritos por autores desde áreas educativas. En el análisis realizado a las experiencias encontradas, solamente se toman algunas de estas competencias por la importancia que tienen en el momento de dar solución a problemas a través de recursos informáticos. Las competencias a las que se hace referencia y son la base de esta investigación se encuentran descritas en el apartado de análisis y resultados.

III. METODOLOGÍA

En esta investigación se realizó un estudio bibliográfico relacionado con experiencias o propuestas que promueven el pensamiento computacional vinculado con la enseñanza de la programación o temas afines. Para este propósito, se realizó un análisis comparativo utilizando una serie de criterios descriptivos que permitieron estudiar cada iniciativa de manera uniforme.

Para buscar la información referente al tema se definieron criterios de inclusión para la selección de los documentos, tales como artículos que abordan el concepto de pensamiento computacional, donde se proponen métodos y herramientas para el desarrollo de este tipo de pensamiento, escritos con propuestas de integración TIC en Iberoamérica, que presentan estándares para la promoción del pensamiento computacional, artículos y publicaciones sobre experiencias TIC realizadas con estudiantes de secundaria y primeros años de universidad, y documentos de propuestas sobre enseñanza de la programación. Al realizar la búsqueda y encontrar diversos trabajos relacionados con el tema, se llevó a cabo un proceso de selección final en el que se generó una recopilación con un total de 90 documentos que se utilizaron en el desarrollo de esta investigación.

De la información encontrada en estos documentos, se seleccionó un total de 29 propuestas educativas alrededor de Iberoamérica, las cuales fueron analizadas y, a partir de esto, se obtuvo una serie de conclusiones relacionadas con el tipo de conocimiento, habilidad y actitud que cada una de ellas desarrollaba y las herramientas y métodos utilizados para lograr este objetivo.

IV. Análisis y resultados

Como se dijo anteriormente, para realizar el análisis de las experiencias encontradas, se tomaron en cuenta una serie de criterios descriptivos como fueron:

- Título de la propuesta
- Región
- Ámbito
 - Institucional
 - Áulico
 - Regional
 - De secundaria
 - Universitario
- Descripción de la propuesta
- Tipo de tecnología utilizada
 - Hardware
 - Software o entorno de desarrollo
- Habilidades, actitudes y conocimiento que se promueven
- Evaluación realizada por los autores de la propuesta
 - Resultados destacados
 - Técnicas aplicadas
 - Aspectos positivos
 - Aspectos negativos
- Consideraciones de la autora

Para realizar el análisis no se adoptaron todos los conocimientos, habilidades y actitudes indicados por los autores, sino que se tomaron en cuenta aquellos que son recurrentes y que se destacan por la importancia que tienen en los procesos de resolución de problemas con el uso de herramientas y recursos informáticos.

Los proyectos, propuestas o experiencias se analizaron contemplando si se vinculan con el desarrollo de las siguientes competencias:

- Formular problemas posibles de solución mediante una computadora.
- Abstractar la información relevante para reconocer los datos principales.
- Diseñar algoritmos para llegar de manera ordenada a la solución.

- Descomponer datos, procesos (modularizar) o problemas en partes pequeñas y manejables.
- Simular modelos de las posibles soluciones antes de ponerlos a prueba.
- Implementar los modelos para saber si la solución propuesta es eficiente y efectiva.
- Generalizar y transferir ese proceso de solución de problemas a una gran diversidad de estos.
- Persistir en el trabajo con problemas difíciles.
- Generar habilidad para comunicarse y trabajar con otros para alcanzar una meta o solución común.

4.1 Experiencias analizadas

Después de definir los criterios a tener en cuenta para indagar cada una de las experiencias, se realizó el análisis a las veintinueve (29) propuestas que se muestran en la Tabla 3:

Tabla 3. Propuestas y experiencias encontradas alrededor de Iberoamérica

PAÍS	EXPERIENCIA	REGIÓN
Argentina	Robótica educativa	Tucumán
	Programando con Robots	La Plata
	Diseño de un juego basado en Interacción Tangible para la enseñanza de Programación	La Plata
	Desarrollando juegos educativos para incrementar la participación de los alumnos en una materia de programación	La Plata
	Program.AR	Nacional - Argentina
	Programación de videojuegos	Puerto Madryn
Colombia	Proyecto GEMA	Ituzaingó
	Entrenamiento en la programación de la computadora a partir de una aplicación para competencias	Neuquén
	JET: Java en Escuelas Técnicas – Programando con RITA	La Plata
Costa Rica	CUPI2	Bogotá
	Uso de un ambiente virtual competitivo para el aprendizaje de algoritmos y programación	Medellín
	Robótica e Investigación: Un medio para la innovación	Medellín
Chile	Programa Nacional de Informática Educativa	Nacional-Costa Rica
	Desarrollo del Pensamiento computacional con Scratch	Santiago
	Uso de Scratch y Lego Mindstorms como Apoyo a la Docencia en Fundamentos de Programación	Valparaíso
	Un taller de robótica para el apoyo de la enseñanza de programación de computadores basado en estilos de aprendizaje	Valparaíso
	Talleres de robótica para niños y adolescentes, Universidad de Valparaíso	Valparaíso
	C ¹⁰⁰	Nacional-Chile
	Taller de Jóvenes Programadores	Nacional-Chile
Experiencias Prácticas con el Uso del Lenguaje de Programación Scratch para Desarrollar el Pensamiento Algorítmico de Estudiantes en Chile	Viña del Mar	

España	Código 21 Robótica Móvil y Programación en Educación Secundaria, Robocampeones. Desafío STEM – Telefónica Fundación mSchools Xnergic	Navarra Fuente labrada Nacional-España Cataluña-Barcelona Barcelona
Ecuador	Scientific Kids University	Quito
El Salvador	Robótica educativa	Soyapango
Perú	Los Mundos Virtuales: una plataforma para el desarrollo de habilidades de programación y de interacción social	Lima
Uruguay	Butiá: Plataforma robótica genérica para la enseñanza de la informática	Nacional- Uruguay

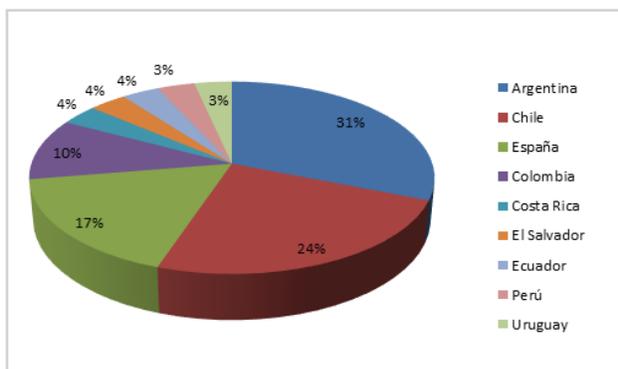
Fuente: Elaboración propia.

Los criterios tomados como referencia sirvieron para hacer un análisis entre las propuestas y experiencias encontradas, de manera que se pudiera conocer los aspectos metodológicos, herramientas y materiales utilizados en cada una de ellas, y así se pudiera definir cuáles fueron las que cumplieron con el objetivo de desarrollar las características relacionadas con el pensamiento computacional y cuáles de estas características no fueron abordadas o poco abordadas.

4.2 Análisis de resultados y conclusiones

Las experiencias analizadas fueron elegidas por ser las más representativas alrededor de Iberoamérica y, según el estudio que se realizó, se pudo conocer con detalle aspectos referentes a las propuestas, entre estos:

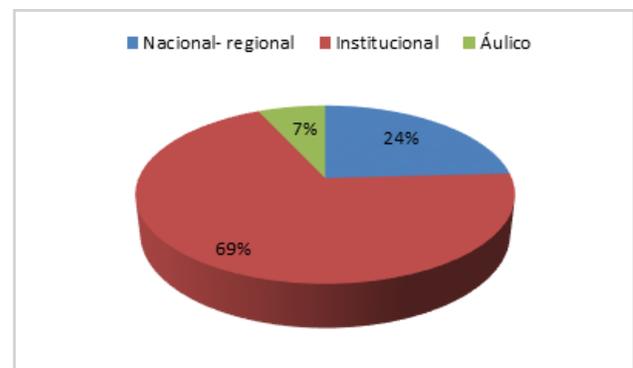
Figura 1. Experiencias por países



Fuente: Elaboración propia.

En la Figura 1 se muestra el porcentaje de propuestas analizadas en cada uno de los países. Como se señala, el país con mayor cantidad de experiencias halladas es Argentina con un 31 %, seguido por Chile con un 24 % de propuestas, Colombia con un 10 %, y los demás países con un porcentaje menor; esto no quiere decir que los sitios con menores propuestas o que no aparezcan señalados no se estén proponiendo estrategias educativas para el desarrollo del pensamiento computacional, sino que la información encontrada al respecto no es suficiente para poder realizar el análisis detallado que se efectuó.

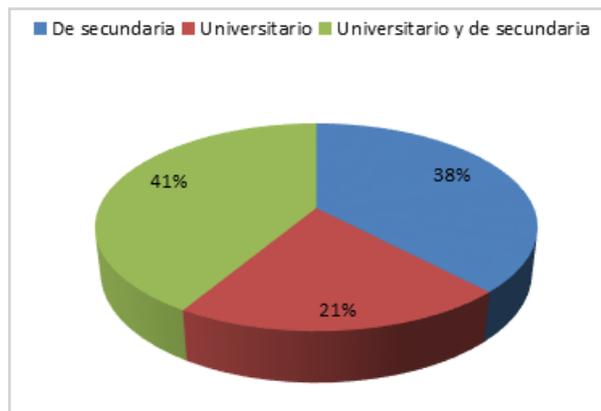
Figura 2. Zonas de aplicación



Fuente: Elaboración propia.

En la Figura 2 se muestra que, de los proyectos encontrados, el 69 % focalizó el trabajo institucionalmente, el 7 % de estos fueron áulicos y el 24 % difundió la propuesta a nivel nacional o regional, como es el caso de PRONIE de Costa Rica, Código 21 en España, Butiá en Uruguay, Desafío Stem en España, Program.AR en Argentina, C¹⁰⁰ y Taller de Jóvenes Programadores en Chile.

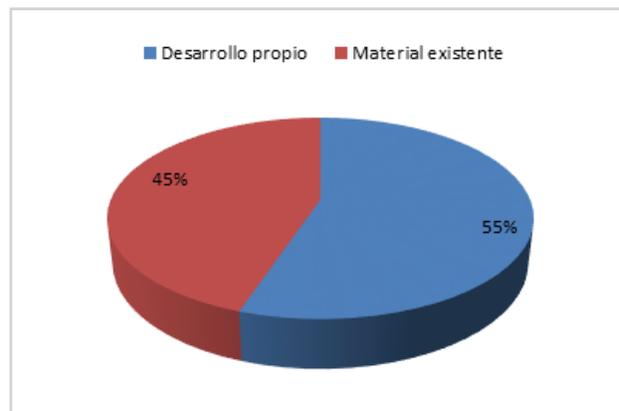
Figura 3. Ámbito de aplicación



Fuente: Elaboración propia.

Uno de los objetivos específicos en este trabajo fue examinar propuestas que desarrollaran habilidades del pensamiento computacional a nivel de secundaria y en los primeros años de universidad. Al realizar el análisis se encontró, como se muestra en la Figura 3, que en un 41 % las iniciativas se aplicaron en un ámbito universitario y de secundaria a la vez; puesto que desde las universidades se están generando e impulsando estrategias educativas que se articulan con el trabajo en instituciones de secundaria, como el caso de “Desarrollando juegos educativos para incrementar la participación de los alumnos en una materia de programación”, donde los estudiantes de segundo año de la Facultad de Informática de la UNLP (Argentina) desarrollan juegos que son utilizados como material educativo en instituciones de secundaria. Del mismo modo, algunas de estas experiencias se enfocan en temas básicos de programación, robótica y afines, cuya temática puede ser trabajada tanto por estudiantes de secundaria como por estudiantes de los primeros años de la universidad. Este es el caso de “Desarrollo del Pensamiento computacional con *Scratch*”, estrategia que se aplicó a estudiantes en nivel básico y nivel universitario, en la Universidad Santiago de Chile (Chile). Por otro lado, se encontró que el 38 % de las propuestas están destinadas al trabajo en la escuela secundaria y un 21 % en los primeros años de universidad.

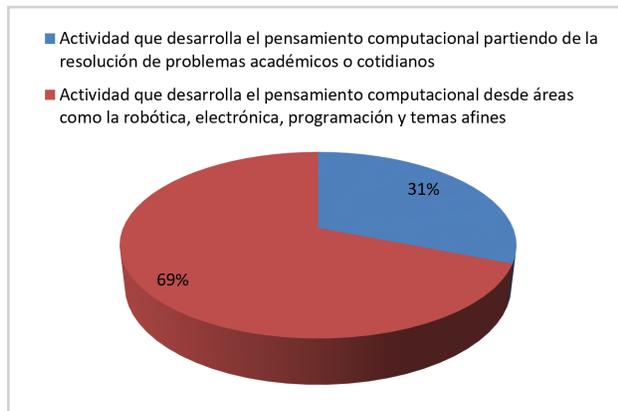
Figura 4. Materiales y herramientas utilizadas



Fuente: Elaboración propia.

Asimismo, como se muestra en la Figura 4, se encontró que en el 55 % de las experiencias se desarrollaron *hardware*, *software* o entornos para la aplicación de sus estrategias, como es el caso de: “Robótica educativa con ICARO” del Ministerio de Educación de la Provincia de Tucumán (Argentina) y “Diseño de un juego basado en Interacción Tangible para la enseñanza de Programación con EPIT” de la Facultad de Informática de la UNLP (Argentina), entre otros. De igual manera, casos como CUIP2 de la Universidad de los Andes (Colombia), Robocampeones de la Concejalía de Educación Madrid (España), Program. AR del Ministerio de Ciencia, Tecnología e Innovación Productiva de la Presidencia de la Nación (Argentina) y “Entrenamiento en la programación de la computadora a partir de una aplicación para competencias” con Hornero de la Facultad de Informática de la Universidad Nacional del Comahue (Argentina), desarrollaron plataformas virtuales por medio de las cuales se trabajó y se tuvo comunicación directa con los estudiantes. Las demás propuestas que alcanzan el 45 % hicieron uso de materiales y herramientas existentes como en la propuesta “Robótica e Investigación: Un medio para la innovación” del colegio Montessori en Medellín (Colombia), en la que se utilizó el *Set Lego Mindstorms*.

Figura 5. Enfoque desde donde se desarrolla el pensamiento computacional



Fuente: Elaboración propia.

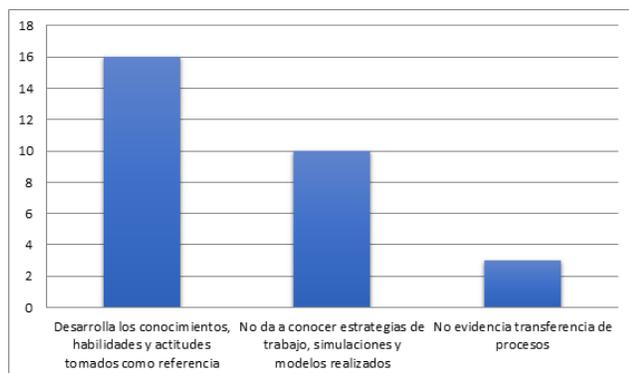
Todas las experiencias analizadas buscan desarrollar habilidades relacionadas con el pensamiento computacional, pero a la vez se observa que algunas de ellas promueven estrategias para solucionar problemas académicos como la deserción escolar, la dificultad en el aprendizaje o la falta de material educativo, aspectos en los que se enfocan las siguientes propuestas: "CUPI2" (Colombia), la cual, por medio de una plataforma educativa, guía y apoya el aprendizaje de los estudiantes brindando materiales e información referente a los temas de estudio; "Diseño de un juego basado en Interacción Tangible para la enseñanza de Programación" (Argentina), en el cual se propone un material educativo para solucionar la dificultad en el aprendizaje de conceptos de programación en estudiantes de primer año en la Facultad de Informática de la UNLP; "Desarrollando juegos educativos para incrementar la participación de los alumnos en una materia de programación" (Argentina): en esta propuesta los estudiantes programan y diseñan aplicaciones o juegos que son utilizados posteriormente en los proyectos de extensión de la Facultad de Informática de la UNLP.

Del mismo modo, otras propuestas buscan satisfacer necesidades de la cotidianidad como brindar ayuda a personas con discapacidad, disminuir la brecha educativa o incentivar vocaciones tecnológicas, tal es el caso de "Robótica e Investigación: Un medio para la innovación" (Colombia), en donde los estudiantes resuelven problemas que encuentran en su entorno haciendo uso de la tecnología

disponible; también el caso de "Robótica educativa" (Argentina), que por medio del trabajo con varias escuelas busca que los estudiantes satisfagan necesidades de su entorno por medio de materiales reciclados; en "Robótica educativa" (El Salvador), la propuesta busca cerrar la brecha educativa existente entre la educación urbana y rural; "Program.AR" (Argentina) pretende motivar el gusto hacia la creación tecnológica; "Proyecto Gema" (Argentina), donde se favorece la articulación de escuelas con el sector productivo, y "C^100" (Chile), que busca incorporar el pensamiento computacional en los programas educativos del país. Este tipo de experiencias ocupan el 31% de las propuestas, como se muestra en la Figura 5.

Por otro lado, el 69 % restante de las propuestas se enfoca en la enseñanza de la programación, electrónica y robótica para la solución de problemas relacionados con esta temática, tal es el caso de las siguientes experiencias: "Programando con Robots" (Argentina), "Uso de un ambiente virtual competitivo para el aprendizaje de algoritmos y programación" (Colombia), "Programa Nacional de Informática Educativa" (Costa Rica), "Desarrollo del Pensamiento computacional con *Scratch*" (Chile), "Uso de *Scratch* y *Lego Mindstorms* como Apoyo a la Docencia en Fundamentos de Programación" (Chile), "Un taller de robótica para el apoyo de la enseñanza de programación de computadores basado en estilos de aprendizaje" (Chile), "Código 21" (España), "Robótica Móvil y Programación en Educación Secundaria, Robocampeones" (España), "Desafío STEM" (España), "*Scientific Kids University*" (Ecuador), "Los Mundos Virtuales: una plataforma para el desarrollo de habilidades de programación y de interacción social" (Perú), "Butiá: Plataforma robótica genérica para la enseñanza de la informática" (Uruguay), "Programación de videojuegos" (Argentina), "Entrenamiento en la programación de la computadora a partir de una aplicación para competencias" (Argentina), "JET: Java en Escuelas Técnicas - Programando con RITA" (Argentina), "Talleres de robótica para niños y adolescentes, Universidad de Valparaíso" (Chile), "Taller de Jóvenes Programadores" (Chile), "Experiencias Prácticas con el Uso del Lenguaje de Programación *Scratch* para Desarrollar el Pensamiento Algorítmico de Estudiantes en Chile" (Chile), "*mSchools*" (España) y "*Xnergic*" (España).

Figura 6. Desarrollo de habilidades relacionadas con el pensamiento computacional



Fuente: Elaboración propia.

En la Figura 6 se muestra el número de propuestas que, según el análisis realizado, cumplen con los conocimientos, habilidades y actitudes tomados como referencia. Sobre un total de 29 propuestas se observa que 16 de las experiencias, equivalentes a 55 %, cumplen con la totalidad de los aspectos, entre las cuales se encuentran las propuestas de Argentina “Robótica educativa”, “Proyecto GEMA”, “Programando con Robots”, “Diseño de un juego basado en Interacción Tangible para la enseñanza de programación”, “Desarrollando juegos educativos para incrementar la participación de los alumnos en una materia de programación”; la propuesta de Colombia “Robótica e Investigación: Un medio para la innovación”, la propuesta de Costa Rica “Programa Nacional de Informática Educativa”, las propuestas de Chile “Uso de *Scratch* y *Legó Mindstorms* como Apoyo a la Docencia en Fundamentos de Programación” y “Un taller de robótica para el apoyo de la enseñanza de programación de computadores basado en estilos de aprendizaje”, las propuestas de España “Código 21”, “Robótica Móvil y Programación en Educación Secundaria, Robocampeones”, “Desafío STEM”, “*mSchools*”, “*Xnergic*”; la propuesta de El Salvador “Robótica educativa” y la propuesta de Uruguay “Butiá”.

Por otro lado, se observa que 10 de las propuestas, equivalente a un 35 %, según la información encontrada, muestran el proceso realizado, pero no se da a conocer aspectos como las estrategias de trabajo, simulaciones y los modelos implementados que permitieron brindar

una solución a los problemas planteados, como en el caso de las propuestas de Argentina “Program.AR”, “Programación de videojuegos” y “Entrenamiento en la programación de la computadora a partir de una aplicación para competencias”, las propuestas de Colombia “CUPI2” y “Uso de un ambiente virtual competitivo para el aprendizaje de algoritmos y programación”, las propuestas de Chile “Desarrollo del Pensamiento computacional con *Scratch*”, “Talleres de robótica para niños y adolescentes, Universidad de Valparaíso” y “C^100”, la propuesta de Ecuador “*Scientific Kids University*” y la propuesta de Perú “Los Mundos Virtuales: una plataforma para el desarrollo de habilidades de programación y de interacción social”.

En el 10 % restante se encuentran 3 experiencias en las cuales la información encontrada no evidencia procesos de transferencia para la solución de problemas en diversos entornos, como ejemplo se pueden mencionar las propuestas “Uso de un ambiente virtual competitivo para el aprendizaje de algoritmos y programación” (Colombia), “Experiencias Prácticas con el Uso del Lenguaje de Programación *Scratch* para Desarrollar el Pensamiento Algorítmico de Estudiantes en Chile” (Chile) y “JET: Java en Escuelas Técnicas – Programando con RITA” (Argentina).

Cabe notar que esta clasificación se realizó desde las consideraciones de la autora y la información encontrada al respecto de cada una de las propuestas analizadas.

Figura 7. Tipo de software utilizado



Fuente: Elaboración propia.

De igual manera, es importante valorar las experiencias que recurren a herramientas gratuitas, libres o de desarrollo propio y que utilizan los equipos informáticos con que cuentan las instituciones, y así se puedan aplicar estrategias educativas de calidad y bajo costo. La Figura 7 muestra que en la información encontrada el 69 % de las propuestas cumplen con estas características, el 21 % de las experiencias hacen uso a la vez de *software* libre o gratuito y *software* propietario, como por ejemplo la propuesta “Uso de *Scratch* y *Lego Mindstorms* como Apoyo a la Docencia en Fundamentos de Programación” (Chile), que utiliza el entorno *Scratch* que es gratuito y el *Set Lego* de carácter propietario; también el “Programa Nacional de Informática Educativa- PRONIE” (Costa Rica), el cual aborda diversas temáticas y, por lo tanto, se utiliza variedad de *software*, y solo el 10 % de las propuestas restantes trabaja con *software* propietario, como es el caso de la propuesta de El Salvador “Robótica educativa”, donde se utilizan los entornos de *Lego Wedo* y *Lego Mindstorms NXT*.

Con respecto a la evaluación realizada por parte de cada uno de los autores sobre sus respectivas propuestas, se pudo conocer a partir de la información encontrada las fortalezas de las estrategias aplicadas y las herramientas utilizadas, así como también algunos de los productos logrados y resultados sobresalientes de cada aplicación. De igual manera, en algunas experiencias se mencionaron los premios obtenidos por la participación en diversos eventos tecnológicos.

La documentación hallada de las diversas experiencias analizadas describe someramente los instrumentos utilizados para su evaluación, así como también las dificultades presentadas durante la implementación de las propuestas. Algunas de las experiencias las cuales disponen de información respecto a los aspectos negativos fueron las siguientes:

- “Programando con Robots” (Argentina), donde se manifestó la dificultad al trabajar con el sistema operativo GNU/Linux, el cual era desconocido por los participantes de la experiencia.

- En la propuesta “Uso de *Scratch* y *Lego Mindstorms* como Apoyo a la Docencia en Fundamentos de Programación” (Chile) la dificultad que se presentó estaba relacionada con la imposibilidad de cubrir toda la temática de una cátedra específica para poder obtener mejores resultados al aplicar la propuesta en el mejoramiento del desempeño académico de los estudiantes.
- También, propuestas como “*Scientific Kids University*” (Ecuador) y “Robótica educativa” (El Salvador) mostraron dificultades por falta de presupuesto y recursos para aplicar sus propuestas de manera gratuita y cubrir la cantidad de instituciones educativas que estaban dentro de su objetivo.
- Otros aspectos negativos se relacionaron con el uso de herramientas no adecuadas según la percepción de los estudiantes, como es el caso de “Los Mundos Virtuales: una plataforma para el desarrollo de habilidades de programación y de interacción social” (Perú).

Por otro lado, las consideraciones de la autora para cada una de las propuestas estuvieron centradas en analizar las habilidades que, al parecer, según la información encontrada, se desarrollaron a partir de la aplicación de las estrategias educativas. Asimismo, se destacan los aspectos sobresalientes de cada una de ellas, como también si resultan opciones adecuadas al entorno donde se aplican.

Como se puede observar del análisis presentado, existen numerosas propuestas, todas con un mismo objetivo, que en este caso es desarrollar habilidades para la resolución de problemas de diversa índole haciendo uso de herramientas tecnológicas. La mayoría de estas experiencias pueden ser adaptadas y utilizadas en otros entornos educativos, puesto que se cuenta con la información referida a las estrategias aplicadas y, en algunos casos, las herramientas utilizadas se pueden adquirir de manera libre o a bajo costo.

V. Recomendaciones

A partir de las particularidades y los alcances de cada una de estas experiencias observadas, resulta importante fortalecer el trabajo en características que son las menos abordadas en las experiencias revisadas, tales como: la implementación de modelos de evaluación para determinar si las soluciones propuestas satisfacen las necesidades planteadas, y el trabajo en equipo para alcanzar una meta o solución común. De igual manera, resulta necesario realizar evaluaciones sistemáticas que permitan medir el impacto de la aplicación de las estrategias en el desarrollo del pensamiento computacional, dado que esta información es carente en gran parte de las experiencias analizadas.

VI. Bibliografía

- Barr, V., y Stephenson, C. (2011). Bringing Computational Thinking to K-12: What is Involved and What is the Role of the Computer Science Education Community ? *ACM Inroads*, 2(1), 48-54. Consultado: Abril 1 de 2016, desde: <https://csta.acm.org/Curriculum/sub/CurrFiles/BarrStephensonInroadsArticle.pdf>
- Center for Computational Thinking Carnegie Mellon. (n.d.). Center for Computational Thinking, Carnegie Mellon. Consultado: Abril 1 de 2016, desde: <http://www.cs.cmu.edu/~CompThink/>
- Compañ-Rosique, P., Satorre-Cuerda, R., Llorens-Largo, F., & Molina-Carmona, R. (2015). Enseñando a programar: un camino directo para desarrollar el pensamiento computacional. *RED. Revista de Educación a Distancia*, 46(11). Consultado: Abril 1 de 2016, desde: <http://doi.org/10.6018/red/46/11>
- Denning, P. J. (2010). What is Computation? *Ubiquity*, (November), 1-11. Consultado: Abril 1 de 2016, desde: <http://ubiquity.acm.org/article.cfm?id=1880067>
- Denning, P. J. (2009). *Taslak The profession of IT Beyond computational thinking*. *Communications of the ACM*, 52(6), 28. Consultado: Abril 1 de 2016, desde: <http://sgd.cs.colorado.edu/wiki/images/7/71/Denning.pdf>
- ISTE, y CSTA. (2011). *Operational Definition of Computational Thinking*. Report, 1030054. Consultado: Abril 1 de 2016, desde: <https://csta.acm.org/Curriculum/sub/CompThinking.html>
- Kafai, Y., Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Silverman, B. (2009). *Scratch: Programming for All*. Consultado: Abril 1 de 2016, desde: <http://web.media.mit.edu/~mres/papers/Scratch-CACM-final.pdf>
- Kemp, P. (2014). *Computing in the national curriculum A guide for secondary teachers* Computing in the. (Computing at School., Ed.). Consultado: Abril 19 de 2016, desde: http://www.computingatschool.org.uk/data/uploads/cas_secondary.pdf
- Linn, M. C., Aho, A. V, Blake, M. B., Constable, R., Kafai, Y. B., Kolodner, J. L., Bradley, S. (2010). *Report of a Workshop on The Scope and Nature of Computational Thinking*. *Thinking* (Vol. 1). Consultado: Abril 1 de 2016, desde: <http://doi.org/10.17226/12840>
- Lugo, M., & Kelly, V. (2010). *Tecnología en educación ¿Políticas para la innovación ?* Instituto Internacional de Planeamiento de La Educación IIPE-Unesco, 13. Consultado: Abril 1 de 2016, desde: www.ibertic.org/evaluacion/sites/.../13_tecnologia_en_educacion_lugo_kelly.pdf
- Moursund, D. (2007). *Computational Thinking and Math Maturity : Improving Math Education in K-8 Schools (Second Edition)*, Computer Science Education, 1-108. Consultado: Abril 2 de 2016, desde: <http://uoregon.edu/~moursund/Books/ElMath/ElMath.html>

- National Research Council. (2011). Report of a Workshop of Pedagogical Aspects of Computational Thinking. Consultado: Abril 1 de 2016, desde: <http://people.cs.vt.edu/~kafura/CS6604/Papers/NRC-Pegagogy-CT.pdf>
- Panoff, R. (2014). Computational thinking for all: the power and the peril. Proceedings of the 45th ACM technical symposium on Computer science education. Consultado: Abril 20 de 2016, desde: <http://dl.acm.org/citation.cfm?id=2554795&dl=ACM&coll=DL&CFID=773650268&CFTOKEN=77132837>
- Sarmiento, M., Gorga, G., y Sanz, C. (2017). Análisis de experiencias y estrategias educativas con TIC para el desarrollo del pensamiento computacional en estudiantes de secundaria y primeros años de universidad en Iberoamérica (trabajo final). Universidad Nacional de La Plata, La Plata, Buenos Aires. Consultado: Febrero 14 de 2018, desde: <http://sedici.unlp.edu.ar/handle/10915/60186>
- SITEAL. (2014). Informe de tendencias sociales y educativas en América Latina 2014. Siteal, 1-262. Consultado: Abril 20 de 2016, desde: http://www.siteal.iipe.unesco.org/sites/default/files/siteal_informe_2014_politicas_tic.pdf
- Wing, J. M. (2006). Computational Thinking. Communications of the Association for Computing Machinery (ACM), 49(3), 33-35. Consultado: Abril 1 de 2016, desde: <https://www.cs.cmu.edu/~15110-s13/Wingo6-ct.pdf>
- Wing, J. M. (2014). Computational thinking. Benefits society. Consultado: Abril 28 de 2016, desde: <http://socialissues.cs.toronto.edu/2014/01/computational-thinking/>
- Wing, J. M. (2012). Computational thinking. Communications of the ACM, 49(3), 33-35. Consultado: Abril 28 de 2016, desde: http://research.microsoft.com/enus/um/redmond/events/asiafacsum2012/day1/Jeanette_Wing.pdf
- Wing, J. M. (2011). Computational Thinking: What and Why? Consultado: Abril 1 de 2016, desde: <http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>