

Una Propuesta para el Desarrollo de Pensamiento Computacional en Niños y Jóvenes

A Proposal for the Development of Computational Thinking in Children and Young People

Maritza García Angarita¹, Claudia Deco^{2,3}, Cristina Bender^{2,3}, Cesar A. Collazos⁴

¹Institución Universitaria Pascual Bravo, Medellín, Colombia

²Departamento de Investigación Institucional, Facultad de Química e Ingeniería del Rosario, Pontificia Universidad Católica, Rosario, Argentina

³Departamento de Sistemas e Informática, Facultad de Ciencias Exactas, Ingeniería y Agrimensura, Universidad Nacional de Rosario, Rosario, Argentina.

⁴Universidad del Cauca, Popayán, Colombia

magangarita@gmail.com, deco@fceia.unr.edu.ar, cbender@uca.edu.ar, ccollazo@unicauca.edu.co

Recibido: 12/02/2020 | Corregido: 16/06/2021 | Aceptado: 27/06/2021

Cita sugerida: M. García Angarita, C. Deco, C. Bender, C. A. Collazos, "Una Propuesta para el Desarrollo de Pensamiento Computacional en Niños y Jóvenes," *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*, no. 30, pp. 16-27, 2021. doi: 10.24215/18509959.30.e2

Esta obra se distribuye bajo **Licencia Creative Commons CC-BY-NC 4.0**

Resumen

En la actualidad existen muchas instituciones, tanto públicas como privadas, preocupadas por las múltiples dificultades que tienen los estudiantes universitarios al enfrentarse con materias como física, matemática, programación, entre otras. Para esto, en Colombia se han propuesto diferentes estrategias a nivel universitario como ser semilleros y clubes académicos, y también a nivel de educación básica secundaria con propuestas de la Secretaría de Educación y de empresas privadas. Algunas de las propuestas son Innobótica, Ingenieríaⁿ e Interchange. Estas propuestas promueven el aprendizaje significativo en varias áreas del conocimiento tales como física, electrónica, matemática, programación, mecánica, entre otras. En este artículo se presenta una propuesta institucional para la Institución Universitaria Pascual Bravo, que propone crear un club enfocado en los más niños, hacia el aprendizaje de software específico para el logro de pensamiento computacional.

Palabras clave: Pensamiento computacional; Propuesta académica; Enseñanza en niños; Enseñanza en adolescentes; Colombia.

Abstract

Currently there are many institutions, both public and private, concerned about the many difficulties that university students face when dealing with subjects such as physics, mathematics, programming, among others. For this, in Colombia different strategies have been proposed at the university level such as seedbeds and academic clubs, and also at the initial level of high school with proposals from the Ministry of Education and private companies. Some of the proposals are Innobótica, Ingenieríaⁿ and Interchange. These proposals promote significant learning in various areas of knowledge such as physics, electronics, mathematics, programming, and mechanics, among others. This article presents an institutional proposal for the Pascual Bravo University Institution. This proposal aims to create a club focused on younger children, towards learning specific software for the achievement of computational thinking.

Keywords: Computational thinking; Academic proposal; Teaching in children; Teaching in adolescents; Colombia.

1. Introducción

En Colombia, así como en muchos países latinoamericanos, es notable el alto grado de deserción en los estudiantes universitarios. Por ejemplo, la noticia "Deserción, ¿Qué estamos haciendo mal?" del periódico El Espectador, Colombia, afirma que para el 2010 se esperaba un descenso de las cifras de deserción (del 45% al 40%), pero por el contrario se mantuvo en 45,8% [1]. En algunas universidades colombianas son aún más preocupantes estas cifras. Los casos que se mencionan en dicho artículo son el Externado con el 53,5% de deserción, la Universidad Central con el 53,2%, la del Bosque con el 50,6% y la de la Universidad Sergio Arboleda con el 49,4%. Los autores de este artículo afirman que muchas son las causas de deserción, tales como la incapacidad de motivar, retener y dar herramientas de bienestar, calidad y financiamiento a los estudiantes por parte de las universidades. Además, la mala alimentación por parte de los universitarios es mencionada por J. Mesa en su artículo "El 60% de los universitarios está mal alimentado" publicado en el mismo periódico [2].

Por otro lado las universidades, en la parte académica, deben buscar diferentes estrategias para que el estudiante pueda motivarse y visualizar un futuro mejor a través de lo que les puede brindar ésta y el programa de formación al cual pertenece. Una de estas alternativas consiste en los clubes académicos y en semilleros de investigación. Pero en la mayoría de los casos estas iniciativas llegan demasiado tarde a los estudiantes, por lo que muchas universidades y secretarías de educación han planteado estas estrategias no sólo a los más grandes, sino a los niños que hasta ahora están en el proceso de desarrollo de su parte lógica. Por esto, la Institución Universitaria Pascual Bravo ha querido innovar y presentar una propuesta para los más chicos en el desarrollo de pensamiento computacional.

Este artículo presenta entonces una propuesta orientada a niños y jóvenes, para el desarrollo de pensamiento computacional. El artículo se organiza de la siguiente forma: En la Sección 1 se presenta la introducción haciendo hincapié en la importancia del desarrollo de este tipo de estrategias. En la Sección 2, se abordan definiciones importantes acerca del pensamiento computacional, de la gamificación y de la importancia de su enseñanza en grados inferiores; así como algunas experiencias importantes con niños y adolescentes en la escuela donde se exponen algunos logros significativos que buscan una educación integral y competitiva. En la Sección 3 se describen las diferentes competencias que pueden ser desarrolladas en proyectos de este tipo. En la Sección 4 se describen varias herramientas que pueden ser usadas para este proyecto y las ventajas que trae cada una de éstas. En la Sección 5 se detalla la propuesta como tal y las etapas que a grandes rasgos deja ver la importancia de cada nivel planteado. Finalmente, se presentan las conclusiones.

2. Conceptos Preliminares

El pensamiento computacional según Wing, es una habilidad fundamental para toda la humanidad y no sólo para las personas relacionadas con el área de la informática, ya que éste permite resolver problemas comunes y reales de una forma rápida, eficiente e innovadora [3]. Según esta misma autora, este tipo de pensamiento genera en las personas la capacidad de realizar análisis ascendente-descendente, además de que incrementa su capacidad creativa, abordando un problema de forma sencilla y creativa.

En este pensamiento se involucra la parte abstracta-matemática e ingenieril, pero a su vez puede ser usado en múltiples escenarios en donde no necesariamente se vincula un lenguaje específico de programación como lo menciona la autora en [4]. Como complemento a esto, [5] plantea que las propuestas de enseñanza de pensamiento computacional ofrecen la posibilidad de apropiarse no sólo de conceptos sino de objetos de estudio (programas, animaciones, videojuegos, robots) logrando un aprendizaje significativo en el estudiante.

Son muchas las estrategias que se usan en el interior del aula y fuera de ella para la enseñanza-aprendizaje del pensamiento computacional. Una de las más importantes por los resultados que se obtienen es el uso de gamificación (analizada en el apartado siguiente 2.1.), ya que ésta proporciona un ambiente más relajado y sencillo para el estudiante.

2.1. El Juego en los Niños

Una de las razones por las que el juego es importante para el aprendizaje, es porque éste es inherente al desarrollo del ser humano, ya que desde que el niño nace lo adopta como un medio para la adquisición de conocimiento. Un ejemplo claro es que los juguetes brindan a los niños la primera forma de interacción no sólo con sus iguales sino además con los adultos, permitiéndole exteriorizar e interiorizar sus emociones como se menciona en el módulo 2 del Diplomado en Pedagogía y Lúdica [6]. Algunos ejemplos de cómo puede visualizarse el juego en los niños desde etapas tempranas y según dicho documento son:

- De 0 a 3 años el juego consiste en la posibilidad de desplazarse (saltar, correr).
- A partir de los 3 años, se usan objetos para simbolizar (teléfono, automóvil, entre otros).
- Entre los 3 y los 6 años, están presentes los juegos de imitación, donde se juega a la casita, a ser papás y mamás, médicos, etc.; lo que refleja que los niños tienen un gran interés en conocer el mundo de los adultos.
- De los 6 años en adelante predominan la práctica de deportes con el objetivo de que el niño aprenda sobre normas y reglas.

Como conclusión, el juego expresa un nivel de desarrollo instintivo en el niño, así como también un nivel de organización de las estructuras cognitivas que Piaget llama estadios evolutivos [7]. Este autor nombra varios estadios. El primer estadio está comprendido entre los 0 a los 2 años donde predomina el juego funcional (estadio sensorio-motor), de los 2 a los 6 años predomina el juego simbólico (estadio pre-operacional), de los 6 a los 12 años el juego de roles (operaciones concretas) y desde los 12 años en adelante predomina la etapa del pensamiento operativo formal donde se tienen la capacidad para razonar de manera lógica y la capacidad para formular y probar hipótesis abstractas. Piaget, además, aclara que el conocimiento en los niños parte de su entorno y de sí mismo, estructurando sus experiencias e impresiones, y organizando sus instrumentos de expresión. En el estadio sensorio-motor, por ejemplo, pueden aparecer juegos no sólo con su cuerpo, sino con objetos (lanzar, golpear, chupar) y con otras personas (sonreír, tocar, etc.). Algunos juegos representativos de este estadio pueden ser localizar objetos, tomar objetos, dejar caer objetos o encontrar objetos escondidos. En la etapa pre-operacional, pueden verse juegos como dar de comer a una muñeca, simular que habla por teléfono, que hace comida o que toca instrumentos, permitiéndole así expresar sus sentimientos y emociones.

Por lo tanto, el juego aporta al niño no solamente en la parte cognitiva, sino también en la social, la emocional y en su desarrollo motor, lo que lo hace tan importante en la vida del ser humano. Sobra decir que el juego influye en la formación no sólo académica sino social que permite una sana adultez. Por otro lado, se indica que el papel del adulto no es más que el de configurador del escenario del juego, el de favorecer la expresión de los niños, ajustar y organizar los juegos según las edades o estadios y finalmente quien observa el desarrollo del mismo. De la misma forma se establece que para que el papel del adulto o mediador sea realmente importante, éste debe indicarles a los niños la intencionalidad del juego (que debe ser analizada con anterioridad) y transmitirla de la forma más clara posible [6]. El adulto o mediador debe buscar la forma de estrechar los lazos y la comunicación entre él y los niños, y entre cada uno de los participantes del juego, además de realizar una buena planeación donde el juego apoye la temática a tratar para que el aprendizaje pueda ser significativo; y debe además ser facilitador en la construcción de conceptos incluyéndolos en distintas situaciones y contextos.

En cuanto al aprendizaje por medio del juego, hay tres teorías fundamentales. La primera hace referencia a la teoría de Piaget, que ya hemos mencionado anteriormente, la de Vigotsky y la de Groos. La teoría vygotskyana establece que el juego es una actividad social, en donde cada integrante tiene un rol definido complementario al propio. Vigotsky, al igual que Piaget, hace énfasis en el juego simbólico como motor para el desarrollo de la imaginación [8]. Por su parte, Karl Groos en su Teoría del Juego como Anticipación Funcional, indica que éste es una preparación para la vida adulta y, basada en la teoría

de Darwin, para la supervivencia [9].

2.2. Los Estilos de Aprendizaje

Ya que todos los seres humanos no aprenden de la misma forma, el juego permite al niño la posibilidad de aprender de una forma más instintiva por medio de la exploración y la socialización. Estas formas de aprendizaje, según Kolb son llamadas estilos de aprendizaje [10].

Según Kolb, en las aulas de clase se pueden encontrar alumnos activos, teóricos, reflexivos y pragmáticos. Los alumnos activos son aquellos que, por ejemplo, se involucran en experiencias nuevas, se entusiasman con las novedades y les gusta llenar sus días de actividades de trabajo en equipo. Los teóricos son aquellos que son capaces de tomar sus observaciones e integrarlas a una teoría específica y son muy lógicos en sus análisis. Los alumnos reflexivos son más observadores, analíticos y de corte más investigativo. Por último, los alumnos pragmáticos son aquellos a los que les gusta comprobar las teorías de forma experimental y les llama la atención poder resolver problemas.

Para este autor los estilos de aprendizaje son cuatro: convergente, divergente, asimilador y acomodador; y van muy ligados a la caracterización de los estudiantes antes mencionada. En el estilo convergente, los estudiantes tienen más habilidades para abstraer y conceptualizar información. A los alumnos de estilo divergente, por su parte, les llama la atención la observación reflexiva, lo que implica que sean más emocionales y creativos (pueden incluirse aquí los artistas). Los asimiladores son expertos en áreas de abstracción, conceptualización y observación reflexiva, la comprensión y creación de modelos teóricos puede ser una de sus mayores fortalezas al contrario del estilo acomodador. Un resumen de las características de los estudiantes con estos estilos de aprendizaje se puede ver en la Figura 1.

Características del alumno convergente	Características del alumno divergente	Características del alumno asimilador	Características del alumno acomodador
Pragmático	Sociable	Poco sociable	Sociable
Racional	Sintetiza bien	Sintetiza bien	Organizado
Analítico	Genera ideas	Genera modelos	Acepta retos
Organizado	Sofidador	Reflexivo	Impulsivo
Buen discriminador	Valora la comprensión	Pensador abstracto	Busca objetivos
Orientado a la tarea	Orientado a las personas	Orientado a la reflexión	Orientado a la acción

Figura 1. Estilos de Aprendizaje

Fuente:

http://www.cca.org.mx/profesores/cursos/cep21/modulo_2/modelo_kolb.htm

2.3. Algunas Experiencias en Colombia

Actualmente en la literatura existen diversos artículos sobre investigación orientada al pensamiento computacional en jóvenes y al logro de competencias no sólo matemáticas sino computacionales, usando diversas tecnologías y programas informáticos. Algunos de los

proyectos son Programación para Niños y Robótica Educativa.

Respecto a *Programación para Niños*, existe ya una gran cantidad de programas y aplicaciones como ser Alice y Scratch. Scratch permite realizar proyectos con juegos y animaciones en los que el conocimiento se debe ir construyendo, como lo indican [5], [11] y [12]. Este programa a su vez permite el aprendizaje autodidacta y el trabajo colaborativo al necesitar grupos para el desarrollo de una propuesta completa, como complementa [13]. De la misma forma, Alice plantea una manera didáctica de aprender a programar, donde también es posible recrear historias, juegos o animaciones.

Con respecto a *Robótica Educativa*, en la Universidad de Cundinamarca, Colombia, junto con colegios ubicados en la Sabana de Bogotá y utilizando como referencia a [14], se crea un curso libre de robótica, donde se pretende construir "robots autónomos seguidores de línea", y en donde el resultado final de esto fue lograr estudiantes muy motivados y comprometidos con su proceso formativo, con mucho compromiso y apropiación del conocimiento.

3. Competencias a Desarrollar

En Colombia, para hablar del desarrollo de pensamiento computacional y de las estrategias para lograr tal fin, se deben considerar las competencias que el Ministerio de Educación (MEN) de dicho país plantea para el área de matemáticas [15], además de las competencias básicas, ciudadanas y laborales propuestas. También se deben considerar las que [16] propone como competencias tecnológicas indispensables a desarrollar tales como el manejo de plataformas virtuales y el manejo de herramientas ofimáticas y de comunicación.

El desarrollo del pensamiento computacional está muy relacionado con el desarrollo de habilidades y competencias en el área de matemáticas que, como lo menciona este Ministerio en [15], está compuesto por cinco procesos generales que son: formular y resolver problemas; modelar procesos y fenómenos de la realidad; comunicar; razonar; y formular, comparar y ejercitar procedimientos y algoritmos. Todos estos procesos van muy de la mano con el pensamiento que se pretende desarrollar. Además dicho documento propone el desarrollo de cinco pensamientos entre los más importantes para el desarrollo de pensamiento computacional, entre los cuales se encuentran los asociados a la parte lógica y matemática, a los sistemas numéricos, métricos y de medidas, entre otros.

Entre las Competencias Básicas se tienen las de pensamiento tecnológico que hacen alusión a la construcción de tecnología y al conocimiento de herramientas, a lo que se suma la comunicativa que permea estas dos.

Las Competencias Laborales se subdividen en dos. Dentro de las competencias Laborales Generales están las asociadas a la toma de decisión y solución de problemas

tanto personales como grupales, muy ligadas a las competencias ciudadanas de las que se habla más adelante. Y las de tipo Organizacional y Tecnológicas que permiten a los jóvenes identificar, transformar e innovar procedimientos.

Por último, las Competencias Ciudadanas permiten una sana convivencia en la vida escolar. Estas pueden lograrse cuando se ejecutan actividades más complejas de forma grupal, donde se debe analizar, debatir y proponer soluciones en pro de algo.

4. Desarrollo de Pensamiento Computacional

En esta sección se muestran algunos juegos para el logro de pensamiento computacional y sus principales características. Estos fueron separados en dos grandes grupos: juegos básicos de lógica y juegos enfocados al aprendizaje de la programación de sistemas informáticos.

4.1. Criterios de Selección de los Juegos

Aunque los juegos proporcionan a los niños y jóvenes el desarrollo de habilidades, es importante resaltar que es el docente o el padre de familia quien debe considerar el juego y el tipo de juego en el que este puede participar. Dentro de los criterios de selección de los juegos se debe tener en cuenta los propuestos por varios autores, Por ejemplo en el caso de los juegos básicos se consideraron los de [17], dentro de los cuales se citan:

- *El momento evolutivo y los objetivos*: hace referencia a los juegos que van de acuerdo al desarrollo del niño como se planteó en apartados anteriores y a los objetivos que se querían conseguir con los mismos. Por eso, en el primer nivel, se verán juegos muy sencillos, ya que está planteado para niños en etapa escolar inicial con un objetivo claro el cual es el ganar habilidad en operaciones básicas, trabajo en equipo, rapidez, anticipación, entre otros.
- *Tiempo, espacio y recursos usados*: se analizaron diferentes variables como la duración del juego, el espacio necesario para el mismo y que los recursos necesarios fueran básicos. Ya que en algunos momentos es complicado contar con espacios fijos dentro de la universidad, se optó por juegos sencillos, con objetos básicos, de corta duración, ya que se sabe que en niños pequeños la concentración se logra por muy poco tiempo.

Para los juegos enfocados al aprendizaje de la programación, se tuvieron criterios adicionales como los que proponen Walker en [18] y Vincent en [19], que se presentan a continuación, vinculados a la aplicación:

- *Incorpora ayuda o tutorial de uso*: Este criterio fue muy importante en la selección de juegos como Scratch, Alice, code.org, Robomind, Js Robot, Codecombat, entre otros, que en algunos casos incluye hasta videotutoriales para su uso.

- La información que contiene es correcta y fiable.
- Su diseño es funcional y visualmente atractivo.
- Los estudiantes pueden salir de la aplicación en cualquier momento sin perder el progreso: En este aspecto juegos como code.org, Robomind, Js Robot, Codecombat, Candy Crush, permiten al estudiante una gran motivación ya que logran visualizar su avance en cualquier momento.
- Es gratuita: Se escogieron aplicaciones libres, ya que la propuesta es para una institución de tipo pública, en la que algunos niveles fueron experimentados en el colegio público de igual nombre.
- Se aplica rápidamente y no se bloquea: Este aspecto es muy importante. Las aplicaciones seleccionadas fueron probadas en diferentes equipos informáticos (computadoras, tablets y equipos telefónicos), donde tuvieron un excelente comportamiento.
- No contiene publicidad: Es importante indicar que aunque la publicidad no es un criterio que permita el descarte de la aplicación, en nuestro caso fue tomada en cuenta, ya que en niños pequeños esta tiende a desconcentrarlos. Así que se optó por aplicación sin publicidad o baja publicidad.
- Promueve la creatividad y la imaginación.
- Proporciona oportunidades de usar habilidades de pensamiento de orden superior.
- Promueve la colaboración y el intercambio de ideas
- Proporciona retroalimentación útil.

4.2. Juegos Básicos de Lógica

Se pueden mencionar dos tipos de juegos: aquellos que pueden ser abordados dentro de un aula convencional y aquellos donde será necesario el uso de un computador.

Para el *primer tipo* podemos encontrar:

- Deducción del valor de la fruta u objeto: Este juego es ampliamente conocido y consiste en colocar varias operaciones con objetos y a partir de estas deducir el valor de cada uno. En el ejercicio de la Figura 2, el resultado de la suma final sería 16, donde la manzana vale 10, la banana 4 y el coco 2.

$$\begin{aligned}
 \text{🍎} + \text{🍎} + \text{🍎} &= 30 \\
 \text{🍎} + \text{🍌} + \text{🍌} &= 18 \\
 \text{🍌} - \text{🥥} &= 2 \\
 \text{🥥} + \text{🍎} + \text{🍌} &= ??
 \end{aligned}$$

Figura 2. Juego Deduciendo Valores de Frutas

- Moviendo cerillas o fósforos: Este juego consiste en analizar qué elemento quitar, para que la operación resultante sea verdadera (ver Figura 3).

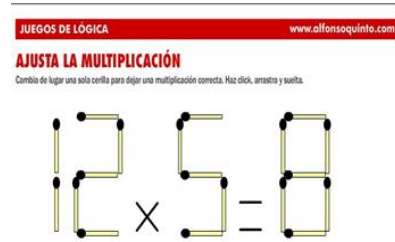


Figura 3. Juego con fósforos o cerillas

Otro juego que se puede realizar con este tipo de elementos consiste en quitar elementos para formar figuras específicas (ver Figura 4).

1. Retiro 2 de los 18 palillos y haz que queden formados 4 cuadrados iguales.	2. Retiro 3 de los 13 palillos y haz que queden formados solo 3 triángulos.	3. Retiro 4 de los 24 palillos y haz que queden formados 5 cuadrados. Halla dos soluciones diferentes.
4. Cambio de lugar 3 de los 12 palillos y haz que queden formados 3 cuadrados iguales.	5. Cambio de lugar 3 de los 12 palillos y haz que queden formados 3 cuadrados iguales.	6. Cambio de lugar 4 de los 12 palillos y haz que queden formados 6 cuadrados.
7. Retiro 4 de los 24 palillos y haz que queden formados 6 cuadrados.	8. Esto es una forma de construir 6 triángulos equiláteros usando 6 palillos. Halla otra forma.	9. Retiro 6 de los 18 palillos y haz que queden formados 4 triángulos.
10. Cambio de lugar 2 de los 12 palillos y haz que queden formados 7 cuadrados.	11. Cambio de lugar 4 de los 12 palillos y haz que queden formados 5 rombos.	12. Retiro 6 de los 24 palillos y haz que queden formados 3 cuadrados.

Figura 4. Formando figuras con cerillas

- Operaciones matemáticas regresivas: Son muchos los ejercicios que podemos encontrar de este tipo, donde a partir de un resultado final se pueden encontrar los valores que lo determinan (ver Figura 5).

		-		66
+	×		-	=
13	12		11	10
×	+		+	-
:	+		×	:

Figura 5. Calculando valores

- Pirámide de números: Para este tipo de ejercicios se deberán llenar los valores de las casillas en blanco de cada una de las pirámides teniendo en cuenta que el número de la fila superior es la suma de los dos valores de la fila inferior (ver Figura 6).

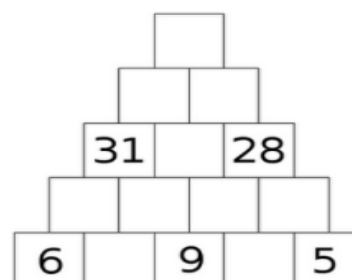


Figura 6. Calculando los valores de la pirámide

- Sudoku: El sudoku es uno de los juegos más conocidos. Consiste en completar una cuadrícula de 9x9 celdas, dividida a su vez en subcuadrículas de 3x3, con las cifras del 1 al 9 partiendo de algunos números ya predefinidos. Las cifras del 1 al 9 no se deben repetir en una misma fila, columna o subcuadrícula (ver Figura 7).

6	9	2			5	3		
			1				9	6
	7	3	2	9	6	4		8
9		1		5				
	3	5	7	8		1		9
		7	9		1	5	4	3
5	2	9	4		8	6		1
	1	6		2	9	8		
				3	9	2		

Figura 7. Ejemplo de cuadrícula en Sudoku

En el *segundo tipo* podemos encontrar:

- Candy Crush Saga: Este juego y su gran cantidad de juegos hermanos poseen múltiples niveles que deben ser superados con lógica. En su fase inicial se deben juntar 3 o más caramelos iguales (ver Figura 8). Pero además cuenta con muchos más retos a medida que avanza de nivel como son: número de movimientos limitados, acabar con la gelatina, bajar los ingredientes, cronómetro y modo de pedidos de caramelos como se indica en la descripción del juego.



Figura 8. Entorno de juego Candy Crush

- Tetris: Es uno de los juegos más antiguos que existe para el desarrollo de lógica. Consiste en dejar caer figuras diferentes (cuadrados, figuras con forma de T, figuras con forma de Z, etc.) y tratar de encajarlas de tal forma que no queden espacios entre ellas, generando líneas horizontales que desaparecen. En este juego es posible rotar las figuras y desplazarlas de izquierda a derecha o al contrario (ver Figura 9).

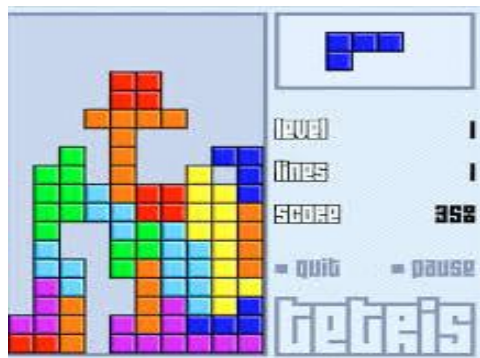


Figura 9. Entorno de juego Tetris

- Mahjon Dimensiones Shanghai: Este juego que en esencia hace parte de los hermanos de Shanghai busca aparear figuras con los mismos símbolos, las cuales van a ir desapareciendo del tablero. Lo importante del juego es que es en un tablero 3D donde las figuras están unas sobre otras y la idea es acabar con todas éstas (ver Figura 10).



Figura 10. Entorno de juego Mahjon

4.3. Juegos para el Aprendizaje de la Programación

Para este primer acercamiento sobre software para aprender a programar en el aula, [17] propone un listado de programas que permiten un aprendizaje progresivo, adaptado, variado y motivador, gracias al uso de bloques lógicos y de animaciones. Algunos de los programas propuestos y sus principales características son:

- Scratch: Este programa permite el aprendizaje en jóvenes. Posee un entorno visual, que hace posible que por medio de bloques o diagramas de flujo puedan ser creadas animaciones de todo tipo [20].

Está dividido de forma general en tres partes. En la primera parte están los bloques lógicos reunidos por categorías. La primera categoría está asociada a los movimientos de los personajes o escenarios, la segunda a la apariencia de sus objetos, siguiendo con los sonidos que pueden emitir los mismos. Luego se tienen algunos bloques de control que van muy de la mano con los sensores, y por último encontramos la posibilidad de crear variables y hacer procesamiento con las mismas. En la segunda parte se encuentra el espacio para el programa o sea para colocar de forma lógica los bloques para que la animación funcione correctamente. Cabe mencionar que es posible tener diferentes escenarios y objetos que a su vez pueden ser modificados por sus disfraces logrando en

la animación un efecto de movimiento. Por último tenemos el espacio para visualizar la animación que se está realizando (ver Figura 11).

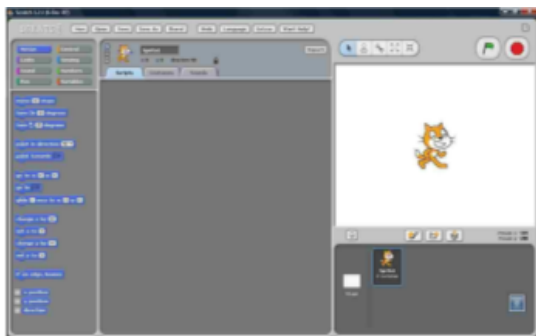


Figura 11. Entorno de programación de Scratch

- Code.org: Este programa es muy similar a Scratch, pero se recomienda para una etapa anterior al uso de éste, ya que propone interesantes tutoriales autoguiados que mejoran el proceso de aprendizaje (ver Figuras 12 y 13).



Figura 12. Tutoriales autoguiados de Code.org



Figura 13. Entorno de Programación Code.org

Incluye también gran cantidad de recursos muy interesantes en esta temática. En una etapa inicial podemos encontrar que el programa nos pide desplazar el ave de color rojo hasta el de color verde, donde es por medio de flechas que se construye el diagrama de flujo.

A medida que se avanza en el aprendizaje del mismo, se pueden encontrar bloques diferentes que pueden incluir "obtener flores" o "recoger miel" como es el caso del ejercicio de la abeja y/o girar 90 grados (a la derecha o izquierda) en el ejercicio de Frozen de recorrer la figura. También es posible encontrar actividades diversas como encontrar la secuencia que hace falta.

- Snap: Es una implementación extendida de Scratch que permite construir bloques o procedimientos propios y es mayormente usado con estudiantes de secundaria o universidad.
- RoboMind y Js Robot: RoboMind se utiliza para programar el comportamiento de un robot por medio de acciones, bucles y condicionales, entre otros. Muy similar a JS Robot con el que es posible aprender JavaScript programando un sencillo juego 2D (ver Figura 14). El juego ofrece varios niveles de dificultad y resulta ideal para todos aquellos que están empezando a aprender JavaScript.



Figura 14. Entorno de programación de Js Robot

- CodeCombat: Es un juego de aventuras que al igual que JS Robot está basado en programación textual, con la ventaja de poder escoger el lenguaje que se desea aprender (ver Figura 15).



Figura 15. Pantalla de Code Combat

En este es posible programar el movimiento del avatar por los distintos escenarios. Al igual que el anterior programa se aprende todo lo relacionado con la sintaxis, los métodos, los bucles y las variables.

- Alice: Este programa aunque es muy parecido a Scratch, ya que permite crear animaciones y juegos, presenta una marcada diferencia ante la propuesta de cómo enseñar a programar a partir de bloques lógicos. En Scratch se utilizan bloques como repetir por siempre, hablar durante x segundos, entre otros, mientras que en Alice es más orientada a objetos. Lo que hace pensar que Alice está diseñada para una población que ya haya tenido un primer contacto con el mundo de la programación [21].

Este programa consta de una pantalla donde se muestra lo recreado hasta el momento, un área para la programación donde podemos encontrar varios eventos y objetos, los menús y herramientas propias del programa, por ejemplo para su edición y ejecución (ver Figura 16).



Figura 16. Partes del Escritorio de Alice

5. Nuestra Propuesta para Niños

En este trabajo se presenta una propuesta para el desarrollo de Pensamiento Computacional para niños de 5 años en adelante, en el marco del proyecto "Sábados en Familia", perteneciente a Bienestar Universitario, dirigido a niños y jóvenes entre los 5 y 17 años (<https://pascualbravo.edu.co/academico/bienest>).

Los criterios que se consideraron a la hora de diseñar la experiencia, fueron basados en el aprendizaje obtenido en las pruebas piloto realizadas en el Instituto Técnico Pascual Bravo y en los análisis anteriores a las mismas, dentro de las cuales se destacan:

- *Edades de los Niños y Número de Participantes:* Se definió que para el desarrollo de la propuesta los grupos no deben sobrepasar los 15 estudiantes, ya que en los primeros niveles propuestos se realizan diferentes actividades, que a su vez son variadas y de corta duración y que deben permitir el intercambio de ideas con ritmos de aprendizaje similares. Los grupos numerosos conllevarían a desorden, desmotivación y en muchos casos se ven ritmos de aprendizaje diversos, que hacen que las actividades en muchos casos no puedan ser desarrolladas. En cuanto a las edades, se plantean a partir de los 5 años en adelante, donde se propone edades propias para cada nivel (de 5 a 7 años nivel 1, de 7 a 8 nivel 2, de 8 a 12 nivel 3, de 12 en adelante para nivel 4 y 5).
- *Niveles, Objetivos, Duración de cada nivel:* En la propuesta, se proponen 5 niveles cada uno de 20 horas de duración, distribuidas en encuentros de 2 horas los días sábado durante 10 semanas. En este caso se plantean actividades cortas, ya que en los primeros niveles se trabajaría con niños muy pequeños, que tienen periodos cortos de concentración. Cada nivel tiene un objetivo claro, en el primero por ejemplo es lograr lógica básica, trabajo

en equipo, motivación, entre otros. En el nivel 2, se brindan los aspectos básicos de la programación, el planteamiento de algoritmos por medio de diagramas de flujo. En el nivel 3 se sumerge al niño en el desarrollo de juegos o aplicaciones sencillas funcionales por medio del diagrama de flujo. En el nivel 4 la idea es enfrentarlo a lenguajes de programación definidos para que así en el último nivel lograr una aplicación de estos programas en Robot.

- *Innovación y Motivación:* La propuesta fue basada en gamificación, que pretende que todos los recursos empleados generen motivación, diversión, retos, premios, trabajo en equipo, entre otros.
- *Que involucra tecnologías y que fuera muy práctico:* En este caso, se pensó en niveles en los que el estudiante no sólo replicara lo que el maestro le está enseñando, sino que además pudiera experimentar nuevas cosas, proponer e innovar. Es por esto que se propone un nivel para el desarrollo de juegos en Scratch y el desarrollo de Robots de competencia.
- *Fácil puesta en marcha y uso de pocos recursos o recursos existentes:* En este caso se tuvo en cuenta no solo el material existente en la universidad (Robots, kits de robótica, DSP, microcontroladores, entre otros) sino además con el que cuenta el Instituto Técnico, ya que son instituciones hermanas que comparten no sólo el Campus sino recursos físicos (material didáctico propuesto para el nivel 1) y humanos (docentes que trabajan en ambas instituciones).

Para el desarrollo de la propuesta se planteó una prueba piloto con estudiantes del Instituto Técnico Industrial Pascual Bravo, en la Técnica en Sistemas y se desarrolló con estudiantes de sexto (nivel 1), séptimo (nivel 2), octavo y noveno (nivel 3) ya que esta institución no tiene educación primaria.

Los niveles son los siguientes:

- 1) **Nivel 1:** Jugando con lógica. Para este nivel se plantean varias clases donde cada una persigue un objetivo diferente, como se puede observar en las distintas secuencias didácticas planteadas por [22]. En este nivel se hace uso de algunos juegos tanto físicos como computacionales para que el estudiante empiece a obtener un poco de lógica. Se trabajan acertijos matemáticos, juegos físicos y juegos como los propuestos en el apartado Juegos Básicos de Lógica. En la prueba piloto se realizaron 5 clases que se detallan a continuación, en donde participaron 27 estudiantes del grado sexto, donde fue posible observar que a medida que avanzaban en las clases lograban realizar los ejercicios en menor tiempo, así como trabajar mejor en equipo.

Clase 1: En esta clase se trabajaron los juegos que aparecen en la Figura 17 (cuadrado, huevo u óvalo, triángulo), en donde el objetivo es que el estudiante logre

habilidades espaciales, donde a partir del despiece de las figuras se puedan obtener otras más, como las mostradas en la Figura 18.



Figura 17. Juegos en 2D

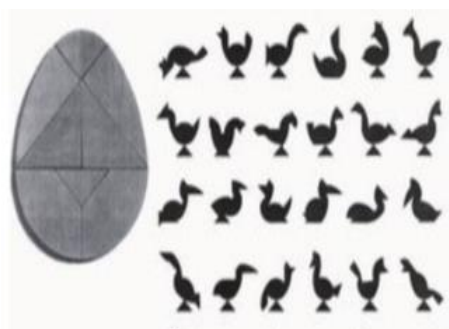


Figura 18. Figuras Para armar con el Ovalo

Clase 2: Se trabajaron algunos juegos en 3D, que consisten básicamente en formar figuras a partir del despiece de las mismas, o en encajar como se muestra en las figuras 19 y 20 las piezas de una forma específica.



Figura 19. Juegos 3D

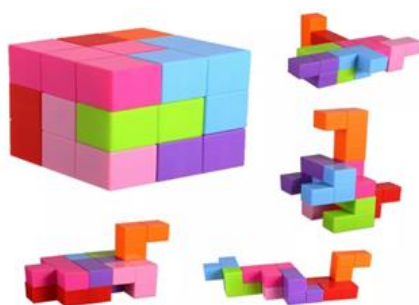


Figura 20. Juegos para armar figuras en 3D

Clase 3: Se trabajaron algunos acertijos como los que se muestran en la Figura 21, donde el objetivo final es extraer un objeto de la figura del alambre sin romperlo o dañar su estructura.

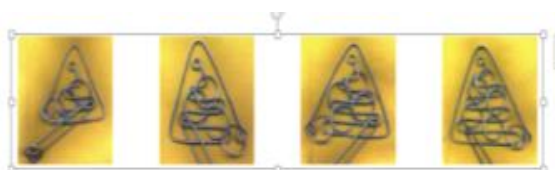


Figura 21. Juegos con figuras de alambre

Clase 4: Se trabajaron algunos acertijos teóricos como los sudokus, las pirámides de valores, entre otros anteriormente descritos.

Clase 5: Se plantearon juegos computacionales para el desarrollo básico de lógica como los descritos en el apartado 4, como Tetris y Candy Crush.

Todas las clases se desarrollaban con la misma metodología, se dividía el grupo en subgrupos de 3 a 4 estudiantes, se les explicaba en qué consiste cada juego, los objetivos, los tiempos esperados, además de recalcarles lo importante del trabajo en equipo y del planteamiento de estrategias de juego (30 minutos). Luego se organizaba el material por mesas de trabajo, para que cada grupo pudiese experimentar con los juegos (1 hora) y por último se organizaba una competencia entre grupos (30 minutos), donde se observaba en ellos no sólo la agilidad sino el trabajo en equipo, los roles adoptados y las estrategias planteadas. Al final se analizaron estos puntos para enriquecer los ejercicios.

- 2) **Nivel 2:** Programando Movimientos. Para este nivel se plantean cinco clases, cada una para enseñar el funcionamiento de programas o aplicaciones como code.org, Robomind, Js Robot, Codecombat, entre otros. Cada una de estas actividades involucran el desarrollo de programas sencillos y con diferentes niveles de dificultad. En este nivel se trabajó de forma individual, para que el estudiante tuviera la oportunidad no solo de conocer los programas sino de retarse a sí mismo. En este caso el docente explicaba el funcionamiento de cada programa, se indicaban ejercicios básicos y a partir de estos se daba vía libre para que el estudiante explorara la aplicación.
- 3) **Nivel 3:** Construyendo Juegos. Para este nivel se trabaja exclusivamente en Scratch y Alice, planteando ejercicios como snake, juego de carros y aviones, así como de programas pedagógicos para la enseñanza de temáticas para niños de primaria. Para este nivel se trabaja en equipos de dos estudiantes, donde inicialmente el docente explica el programa con sus bloques lógicos y luego propone ejercicios a desarrollar (juegos a desarrollar). Los estudiantes al mismo tiempo deben desarrollar una bitácora con el desarrollo del ejercicio, donde debe incluir personajes, funciones de cada uno, características, niveles, sistema de puntuación o premios, entre otros. En este nivel se desarrollaron animaciones sencillas, ejercicios de carros, quien quiere ser millonario, vestir muñecas, tanques, entre otros.
- 4) **Nivel 4:** Lenguajes de Programación: Java y Visual. Consta de cinco clases, donde las primeras 4 son para explicar generalidades de los lenguajes y para la realización de ejercicios básicos. La última clase es para que el estudiante desarrolle un juego sencillo, como por ejemplo un quién quiere ser millonario.
- 5) **Nivel 5:** Programando Robots. Previo a esta propuesta, en 2017 se desarrolló en forma exitosa este nivel de Robots, en el Instituto mencionado [23]. Es

esta la razón fundamental que hace indispensable el desarrollo de una propuesta completa, articulada, progresiva para niños desde los cinco años en adelante. Este nivel se desarrolló con las siguientes fases:

- *Capacitación en el uso de elementos electrónicos.* En esta fase un facilitador o docente, enseña los fundamentos de la robótica, ambiente de programación de Arduino, partes mecánicas del robot como engranajes y poleas, y capacita en la parte electrónica: manejo de baterías, resistencias, sensores, motores, etc.

Estos robots deben cumplir con las tres condiciones que lo hacen ser un robot: ser totalmente autónomo (no tener conexiones físicas con otros dispositivos), ser programable lo que hace que pueda cargarse códigos diferentes para cada función y además contar con un controlador que lo haga tomar decisiones según las variables que se obtengan del medio.

- *Reconocimiento del dispositivo Arduino.* Para esto es necesario que el docente explique las ventajas con las que cuenta la tarjeta a programar (ver figura 22), sabiendo que esta tarjeta presenta características para el área de robótica educativa como su facilidad de programación, su bajo costo y la velocidad de procesamiento [24].

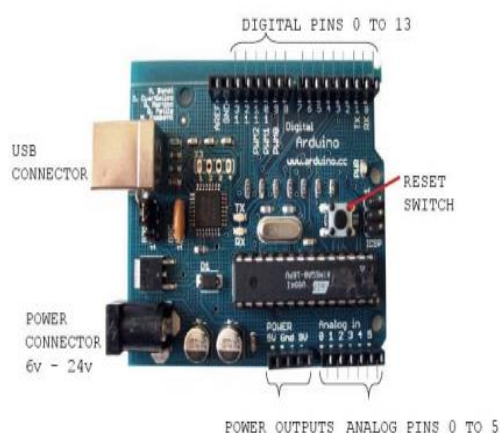


Figura 22. Tarjeta de plataforma Arduino Mega

- *Capacitación en la programación del Robot.* Para crear un proyecto de este tipo se deben conocer los entornos de desarrollo que ofrece Arduino y las ventajas de cada uno. Dentro de las posibilidades que existen para este tipo de hardware están Scratch for Arduino, Eclipse, Visual Studio, Codebender, entre otros. Scratch es un entorno de programación visual, estructurado por bloques que definen el comportamiento de la placa mientras que Eclipse y Visual Studio integran por medio de plugins asistentes para ayudar al programador en la realización del código. El programa que se usará para programar estos dispositivos dependerá de las edades de los niños que participen.

Según la propuesta planteada en el proyecto "Sábados en Familia", para su realización se necesitan recursos tanto

económicos como humanos. Entre estos recursos se deben incluir: elementos de robótica, fotocopias y material didáctico, transporte para estudiantes y capacitadores, alimentación y refrigerio, honorarios del capacitador y espacio para los talleres. Dicho espacio para los talleres debe ser amplio y contar con mesas de trabajo; además se necesita contar con otro espacio con un mínimo 16 computadores.

Algunos resultados obtenidos hasta el momento han sido la aplicación del nivel 5, dentro del proyecto "Sábados en Familia". Asimismo, en 2017 se desarrollaron los niveles 1, 2, 3 y 5 en el Instituto Técnico Industrial Pascual Bravo, donde se observó que el uso de material didáctico (nivel 1), programas o aplicaciones (niveles 2 y 3) y el uso de robótica educativa (nivel 5), generaban competencias y desarrollo de pensamiento computacional, permitiendo observar en los niños grandes conocimientos en la parte de lógica e incluso eléctrica y electrónica, así como en la parte mecánica y de programación avanzada para el caso del diseño de los robots. Esto se evidenció en el lenguaje utilizado por los niños y el desarrollo propio de los diferentes proyectos (nivel 3 y 5), con lo cual se puede afirmar que también desarrollaron competencias de todo tipo mencionadas en el apartado 3.

Aunque la propuesta completa no se ha llevado a cabo en la Universidad por cuestiones de tiempo, dinero y organización, sí se han llevado a cabo pruebas piloto y cursos cortos que permiten un mayor aprendizaje con la estrategia planteada.

Conclusiones

En este trabajo se presenta una propuesta para desarrollar el pensamiento computacional en los niños. Con esta propuesta se pretende fomentar por medio de varias estrategias, que el estudiante se motive hacia la adquisición de conocimiento de una forma divertida, práctica, sencilla y diferente, que lo lleve a generar un aprendizaje sólido y transversal en diferentes áreas del conocimiento. Es importante destacar la importancia de planear los diferentes niveles, ya que cada uno le aporta al niño una estrategia diferente y divertida para motivarse y trabajar en equipo y a la vez de aprehender la programación, comenzando desde pasos básicos hasta finalmente lograr llevar este conocimiento a un objeto real como es el robot. En particular, el uso de robótica educativa para la adquisición de competencias y el desarrollo de pensamiento computacional en el Instituto Pascual Bravo de Colombia, permitió observar en los niños la adquisición de conocimientos en temas de electricidad, electrónica, mecánica y programación. Esto se evidenció en el lenguaje utilizado por los niños y el desarrollo propio del proyecto, con lo cual se puede afirmar que desarrollaron diversas competencias.

Referencias

- [1] S. Silva Numa y M. Baena. Artículo "Deserción, ¿qué estamos haciendo mal?," *El Espectador*, 17 Oct. 2015. [Online], Available: <https://www.elespectador.com/educacion/desercion-que-estamos-haciendo-mal-articulo-593308/>. [Accessed Jul. 2018].
- [2] J. Mesa, "El 60% de los universitarios está mal alimentado," *El Espectador*, 8 Abril 2016. [Online], Available: <https://www.elespectador.com/salud/el-60-de-los-universitarios-esta-mal-alimentado-articulo-625899/> [Accessed Jul., 2018].
- [3] J. Wing, "Computational thinking," *Communications of the ACM*, vol. 49, no. 3, pp. 33-35, 2006. [Online], Available: <https://dl.acm.org/doi/10.1145/1118178.1118215>
- [4] J. Wing, "Research Notebook: Computational Thinking-What and Why?," *The LINK*, 2011. [Online], Available: <https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>
- [5] S. Papert y I. Harel "Situating constructionism," in *Constructionism*, Ed. Norwood, NJ: Ablex, 1991, pp. 1-11.
- [6] Politécnico de Colombia, "El Juego como Potenciador del Aprendizaje," Politécnico de Colombia, Diplomado Virtual en Pedagogía y Lúdica, 2012. [Online], Available: <https://www.politecnicodecolombia.edu.co/diplomados-virtuales-gratis/escuela-de-educacion/pedagogia-y-ludica.html#>
- [7] P. A. Osterrieth, J. Piaget, et al., *Los estadios del desarrollo intelectual del niño y el adolescente*. Buenos Aires: Nueva Visión. 1979.
- [8] L. Vygotsky, *El desarrollo de los procesos psicológicos superiores*. Barcelona: Crítica. 1979.
- [9] K. Groos, *The Play of Man*. New York: Appleton, 1901.
- [10] D. Kolb, *Experiential learning experiences as the source of learning development*. Nueva York: Prentice Hall, 1984.
- [11] Y. Kafai, "Learning design by making games: Children's development of design strategies in the creation of a complex computational artifact," in *Constructionism in practice: Designing, thinking, and learning in a digital world*, Y. Kafai and M. Resnick, Eds. Mahwah, NJ: Lawrence Erlbaum Associates, Inc., 1996, pp. 71-96.
- [12] A. Kay, "Squeak etoys, children, and learning," *Squeakland*, 2010. [Online], Available: <http://www.squeakland.org/resources/articles/> [Accessed Abr. 2019].
- [13] J. Maloney, M. Resnick and N. Rusk, "The Scratch Programming Language and Environment," *ACM Transactions on Computing Education (TOCE)*, vol. 10, no. 16, pp. 1-15, 2010. [Online], Available: <https://dl.acm.org/doi/10.1145/1868358.1868363>
- [14] J. Márquez y J. Ruiz, "Robótica educativa aplicada a la enseñanza básica secundaria," *DIM. Revista Didáctica, Innovación y Multimedia*, año 10, no. 30, diciembre 2014. [Online], Available: <http://dimglobal.net/revistaDIM30/revista30ARrobotica.htm>
- [15] Ministerio de Educación de Colombia, *Estándares Básicos de Competencia en Matemática*, 1998. [Online], Available: https://www.mineducacion.gov.co/1621/articles-116042_archivo_pdf2.pdf
- [16] P. Marqués, "Los Docentes: Funciones, Roles, Competencias Necesarias, Formación," 2000. [Online], Available: <https://bloqs.xtec.cat/formaciotic1/files/2009/03/funciones-de-los-docentes-hoy.pdf>
- [17] F. Posada Prieto, "16 aplicaciones para trabajar el pensamiento computacional en el aula," *CanalTIC.com*, 2016. [Online], Available: <https://canaltic.com/blog/?p=2610> [Accessed Oct. 2019].
- [18] H. Walker, "Evaluation Rubric for Educational aplicaciones," *Learning in Hand*, 2010. [Online], Available: <https://learninginhand.com/blog/evaluation-rubric-for-educational-apps.html> [Accessed May. 2019].
- [19] T. Vicent, "Ways to evaluate educational aplicaciones," *Learning in Hand*, 2011. [Online], Available: <http://learninginhand.com/blog/ways-to-evaluate-educational-aplicaciones.html> [Accessed Abr. 2019].
- [20] J. C. López García, *Programación con Scratch. Cuaderno de trabajo dirigido a los estudiantes de grado 3 a 6*, 4a Ed. Fundación Gabriel Piedrahita Uribe, 2011.
- [21] P. J. Fonseca Solano, *Programe en Alice 2.2*. [Online], Available: <https://www.escinf.una.ac.cr/pfonseca/Alice.pdf> [Accessed Nov. 2019].
- [22] M. García, *Diario de campo: Experiencias significativas en el área de programación*. Instituto Técnico Industrial Pascual Bravo, 2019.
- [23] M. García, C. Deco and C. Collazos, "Robotics Based Strategies to Support Computational Thinking: The Case of the Pascual Bravo Industrial Technical Institute," *Journal of Computer Science and Technology*, vol. 17, no. 1, pp. 59-67, April 2017. [Online], Available: <https://journal.info.unlp.edu.ar/JCST/article/view/450>.
- [24] O. Torrente, *Arduino: curso básico de formación*. Alfaomega, 2013.

Información de Contacto de los Autores:

Maritza García Angarita
Medellín
Colombia
magangarita@gmail.com

ORCID: <https://orcid.org/0000-0001-5159-0280>

Claudia Deco

Av. Pellegrini 3314
Rosario
Argentina

cdeco@uca.edu.ar, deco@fceia.unr.edu.ar

ORCID: <https://orcid.org/0000-0002-7146-1571>

Cristina Bender

Av. Pellegrini 250
Rosario
Argentina

bender@fceia.unr.edu.ar, cbender@uca.edu.ar

ORCID: <https://orcid.org/0000-0001-6961-1983>

Cesar A. Collazos

Popayán
Colombia

ccollazo@unicauca.edu.co

ORCID: <https://orcid.org/0000-0002-7099-8131>

Maritza García Angarita

Magister en Software Libre, Especialista en Gestión de Proyectos Informáticos e Ingeniera Electrónica. Doctoranda en Facultad de Humanidades (UNR). Docente e investigadora en el Instituto Universitario Tecnológico Pascual Bravo.

Claudia Deco

Doctora en Ingeniería y Magister en Informática. Profesora e investigadora en la Universidad Nacional de Rosario y en la Universidad Católica Argentina Campus Rosario, Argentina.

Cristina Bender

Magister en Informática e Ingeniera Electrónica. Profesora e investigadora en la Universidad Nacional de Rosario y en la Universidad Católica Argentina Campus Rosario, Argentina.

Cesar A. Collazos

Doctor en Ciencias de la Computación, Ingeniero en Sistemas y Computación. Profesor y Coordinador del Grupo de Investigación IDIS, Departamento de Sistemas, Universidad del Cauca, Colombia.