

Pensamiento computacional mediante el uso de robots en Educación Primaria — Ozobot y Mbot

Computational thinking using robots in Primary Education — Ozobot and Mbot

Manuel García Piqueras ⁽¹⁾, María Sotos Serrano ⁽¹⁾, José Manuel Diego Mantecón ⁽²⁾ y Teresa F. Blanco ⁽³⁾

manuel.gpiqueiras@uclm.es, maria.sotos@uclm.es, diegojm@unican.es, teref.blanco@usc.es

⁽¹⁾ Universidad de Castilla-La Mancha (España), ⁽²⁾ Universidad de Cantabria (España), ⁽³⁾ Universidad de Santiago de Compostela (España)

Resumen

El objetivo de este artículo es mostrar actividades planteadas en el entorno de la robótica con el propósito de desarrollar el pensamiento computacional, en Educación Primaria. Nos apoyaremos en dos tipos de robots: Ozobot y Mbot. Los Ozobot pueden utilizarse a partir de los 3 años de edad mientras que los Mbot son más adecuados a partir de los 6 años. La resolución de un problema planteado en el entorno de la robótica requiere, en general, de dos fases: 1) obtener una primera solución que no haga uso del instrumento robótico y 2) generar un algoritmo que, implementado en el robot, reproduzca y complete la solución hallada en la fase anterior.

Palabras clave: Educación primaria, Robótica, Pensamiento computacional

Abstract

This paper aims to present activities in a robotics environment with the purpose of developing computational thinking in Primary Education. We rely on two types of robots: Ozobot and Mbot. The Ozobot robot can be used from 3 years of age while the Mbot robot is more suitable from 6 years of age. The resolution of a problem involving robotics requires, in general, two phases: 1) obtaining a first solution without using the robotic instrument, and 2) generating an algorithm that, implemented in the robot, reproduce and complete the solution found in the previous phase.

Keywords: Primary education, Robotics, Computational thinking

1. Introducción

El *pensamiento computacional*, término acuñado por Wang (2006), ha sido objeto de estudio e investigación en el entorno educativo en las últimas décadas. Se define como “el proceso que permite formular problemas de forma que sus soluciones puedan ser representadas como secuencias de instrucciones y algoritmos” (Aho, 2012, p.1). The National Research Council (NRC) referencia el pensamiento computacional como uno de los principales retos de la instrucción actual en materia de educación (Kelley y Knowles, 2016). Nuestra sociedad requiere una destreza computacional, acorde con los avances tecnológicos, no solo como meros consumidores sino también como agentes activos, independientemente del sector donde se trabaje (Segredo, Gara y León, 2017). En este sentido el pensamiento computacional es una poderosa herramienta para la innovación en prácticamente todos los ámbitos de la economía y otras áreas del conocimiento. Aunque a nivel internacional la robótica está integrada, al menos parcialmente en muchos de los currículos educativos, la puesta en práctica en el aula es todavía limitada (Blanco et al., 2017). Esta limitación se debe entre otros factores a que la formación del profesorado no progresa al mismo ritmo que los avances tecnológicos. Se hace imprescindible una transformación de la enseñanza educativa que permita desarrollar las competencias que demanda nuestra sociedad. Trilling y Fadel (2012) clasifican estas competencias en tres bloques principales.

- a) Competencias sobre aprendizaje e innovación: creatividad e innovación, comunicación y colaboración, pensamiento crítico y resolución de problemas.
- b) Competencias sobre alfabetización digital: en materia de información, de medios digitales, e ICT (Information and Communication Technology).
- c) Competencias sobre desarrollo profesional y personal: flexibilidad y adaptabilidad, iniciativa y auto-dirección, interacción social y multicultural, productividad y responsabilidad, y liderazgo.

2. Características de los robots Ozobot y Mbot

El Ozobot 2015 es un robot educativo diseñado para el período de tres a doce años. Se puede utilizar de dos maneras; como un robot que sigue líneas, o programándolo con un lenguaje sencillo a través de bloques visuales. En el primer caso los robots siguen

circuitos sobre una plantilla que, como muestra la Imagen 1, tiene líneas que contienen franjas de colores. Estas franjas proporcionan diferentes instrucciones al robot dependiendo de la combinación de los colores. Por ejemplo, una franja con dos colores azules indica ir rápido, una franja con dos colores rojos ir despacio, y una franja con colores azul, rojo y verde indica girar a la derecha.



Imagen 1. Plantilla Ozobot con diferentes franjas de colores

El kit de Ozobot proporciona varias plantillas con franjas en blanco para colorear, de acuerdo con las instrucciones que se quieran transmitir al robot. La Imagen 2 muestra la tabla de códigos, con las combinaciones de colores; Ozobot las interpreta como instrucciones de programación.



Imagen 2. Tabla de códigos de Ozobot

El Ozobot también puede utilizarse mediante un sencillo entorno de programación, a través de bloques visuales que generan un código que luego se transmite al robot (ver Imagen 3). Una vez implementado el código en el robot este llevará a cabo las instrucciones que se hayan incluido en dicho programa.

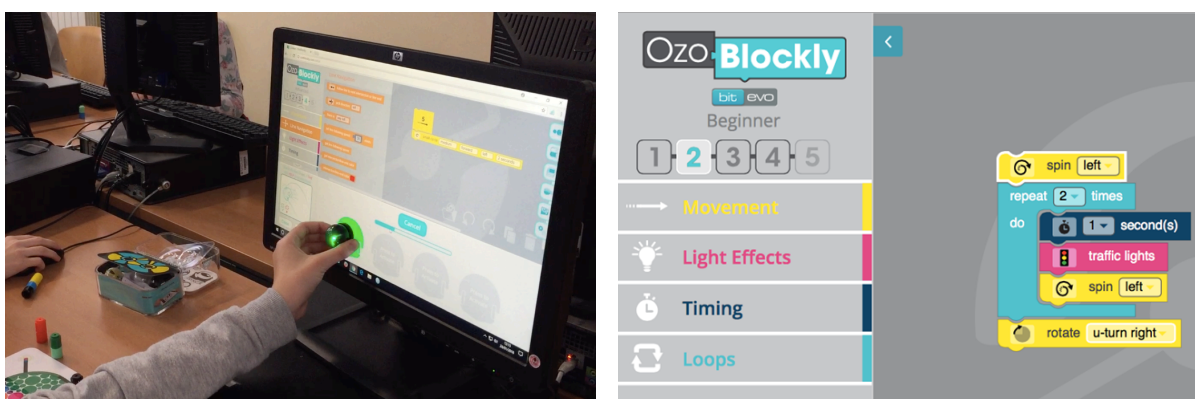


Imagen 3. Entorno de programación por bloques visuales para Ozobot

El Mbot 2017 es un robot educativo recomendado para niños a partir de los seis años de edad. Hay distintos tipos, en este caso nos referimos al kit Mbot 2.4G que aparece en las imágenes 4 y 5. Si bien los Ozobot vienen completamente montados, el Mbot debe montarse bajo la supervisión de un adulto si el alumno tiene menos de 12 años. Una vez montado, el Mbot tiene muchas posibilidades, puesto que además de tener un sensor sigue líneas, también dispone de un sensor de ultrasonidos, un sensor de luz, y dos motores independientes para las ruedas izquierda y derecha. Esta ampliación de posibilidades conlleva un aumento de la dificultad de programación, cuyo proceso puede realizarse mediante el entorno básico de programación mBlock (Imagen 6). Este entorno se basa en el lenguaje de programación visual por bloques Scratch y Arduino para implementar el programa en el robot.



Imagen 4. Robot Mbot 2.4G durante su montaje

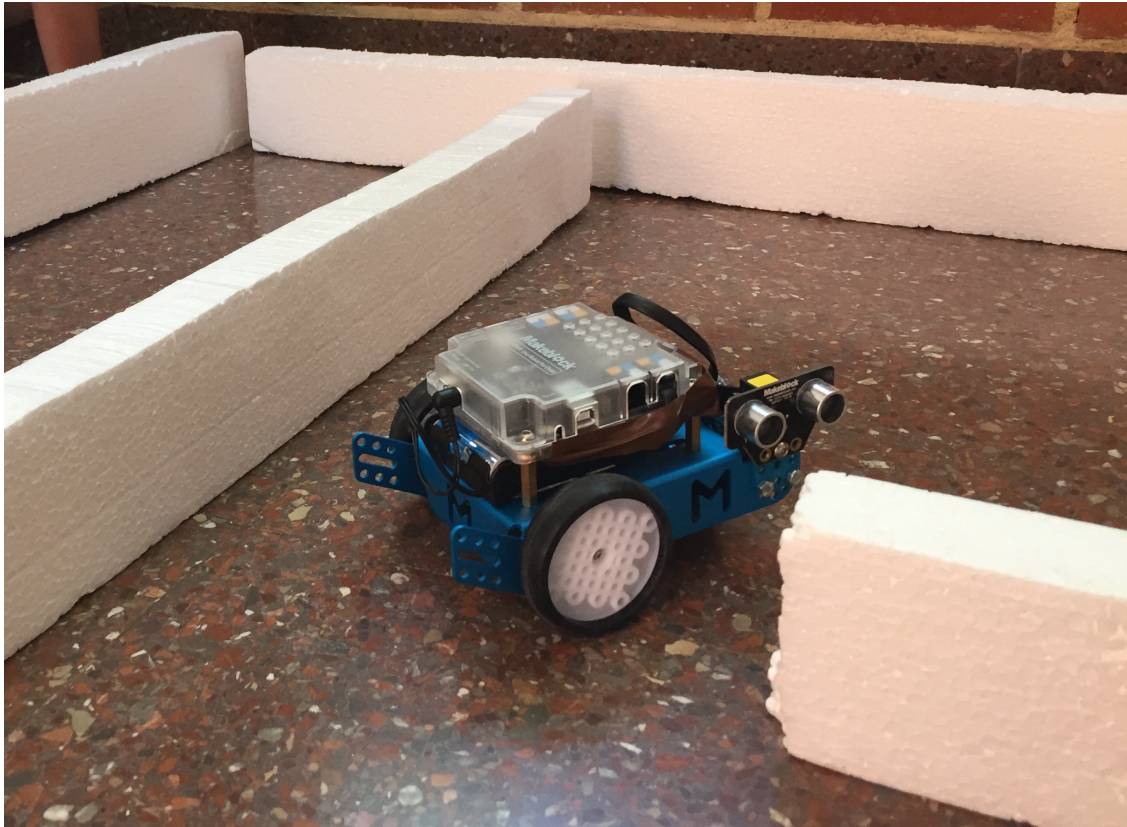


Imagen 5. Robot Mbot 2.4G ya montado

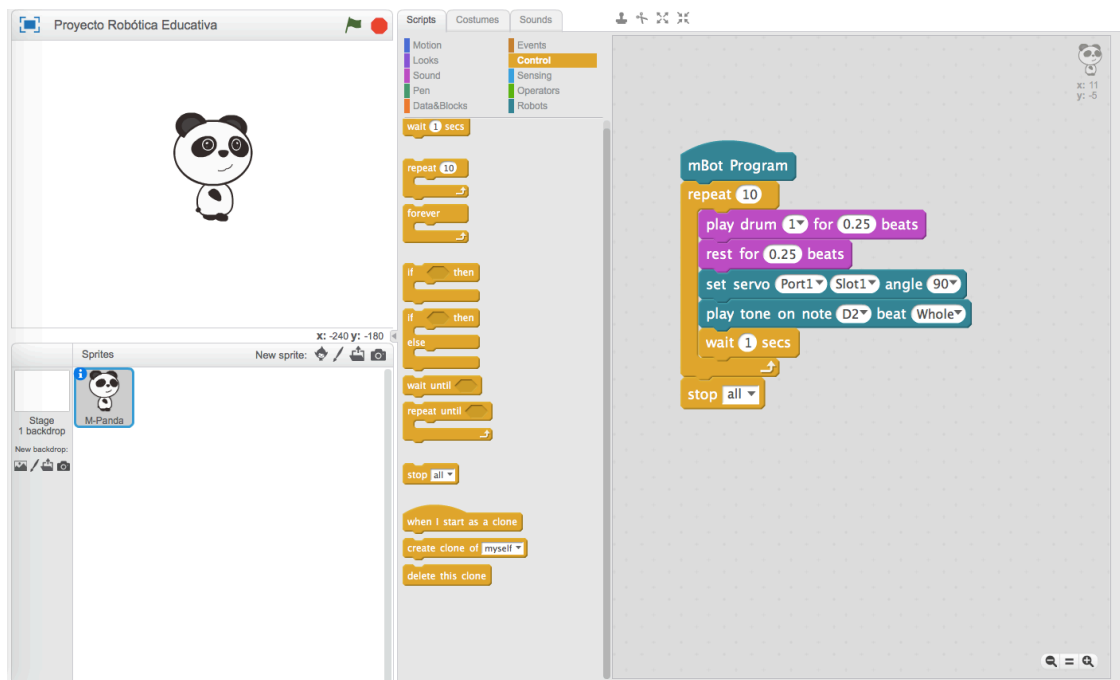


Imagen 6. Entorno de programación básica mBlock para Mbot

4. Ejemplos de problemas computacionales resueltos con Ozobot y Mbot

A continuación, se presenta una experiencia que incluye dos problemas: el problema de los bolos y el problema del laberinto sin islas. Esta experiencia se llevó a cabo con 20 alumnos españoles de 10-11 años de sexto de primaria. Todos eran alumnos del programa bilingüe, con un rendimiento académico alto en ciencias. Estaban familiarizados con las TIC, aunque desconocían los robots aquí presentados, así como los entornos de programación utilizados. Los problemas fueron resueltos por parejas, durante varias sesiones del curso académico 2017-2018, dentro del proyecto STEMforYouth. Este es un proyecto de carácter internacional en el que los alumnos de seis países europeos realizan actividades STEM, para presentarlas en diferentes eventos educativos como ferias de la ciencia o congresos. Información más detallada sobre el proyecto puede encontrarse en Diego-Mantecón, Sáenz de la Torre y Brzozowy (2017) y Brzozowy et al. (2017). Resolvieron los problemas por parejas.

Problema de los bolos resuelto con Ozobot

El *problema de los bolos* consiste en tirar todos los bolos que se sitúan en la plantilla de la Imagen 7, en el menor número de movimientos posibles y partiendo desde la parte inferior del circuito. En las intersecciones marcadas con un círculo rojo se colocan los bolos. A los alumnos se le presentó inicialmente la plantilla con los bolos y se le planteó el problema. A continuación, se les enseñó en qué consistía el robot Ozobot y se les instruyó en su manejo, explicándoles tanto su programación por ordenador como el significado de los códigos de colores incrustados en las trayectorias que siguen los robots.

La resolución del problema se realizó en dos fases. En la primera fase, el alumnado resolvió el problema con lápiz y papel; es decir, sin hacer uso del robot. Obtuvieron una trayectoria mínima que pasaba por todos los bolos, como se observa en la Imagen 8.

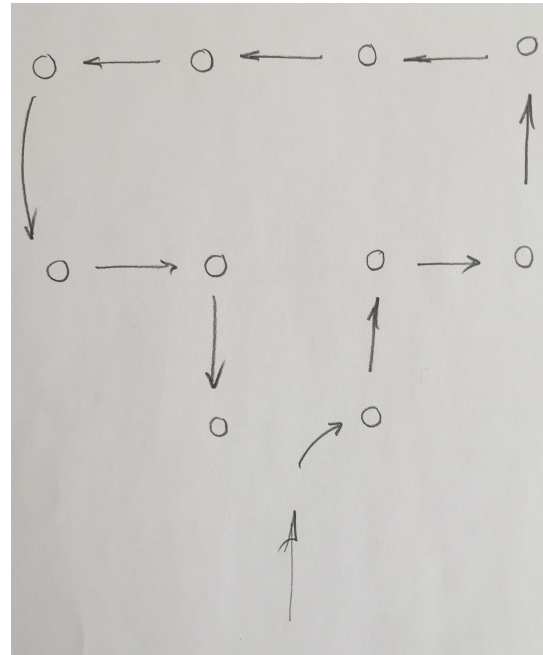
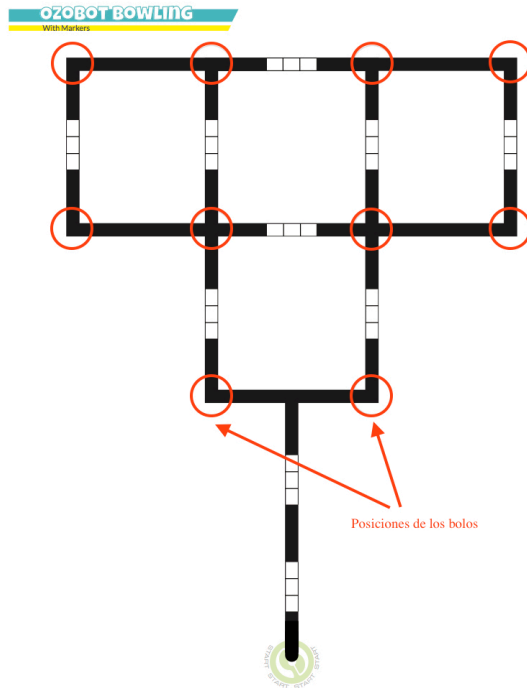


Imagen 7. Plantilla que representa el problema de los bolos Imagen 8. Resolución con lápiz y papel del problema

En la segunda fase, una vez obtenida la trayectoria mínima, colorearon los códigos necesarios en los huecos reservados de color blanco, para que el robot pudiese girar a izquierda o derecha en cada intersección, y fuera más rápido o más despacio tirando todos los bolos (Imagen 9).

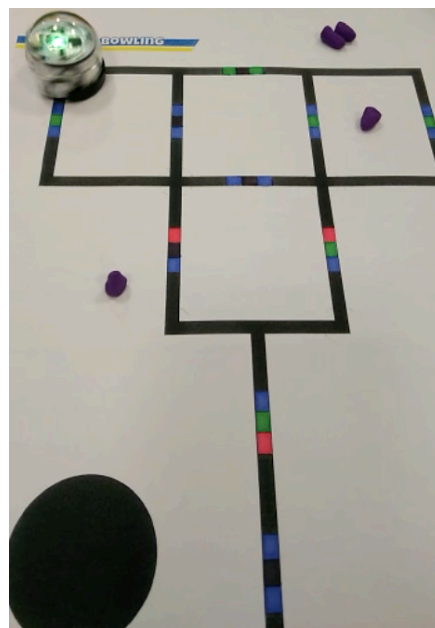


Imagen 9. La resolución del problema de los bolos en la plantilla Ozobot

En nuestro caso, los alumnos evidenciaron mayores dificultades en la parte 1, donde tenían que encontrar una primera solución al problema planteado. Aunque, también se observaron dificultades a la hora de introducir la información en el robot y computar la aplicación. El alumno tiene problemas para asimilar que toda acción debe programarse explícitamente, ya que el robot solo realiza aquellas funciones indicadas previamente a través del código de programación. Durante el desarrollo de la actividad los estudiantes manifestaron un alto interés, tanto a la hora de aprender a manejar el robot, como en el momento de resolver el problema.

Problema del laberinto sin islas resuelto con Mbot

Los laberintos *sin islas*, como el de la Imagen 10, pueden resolverse mediante un robot que esté programado para seguir las paredes y girar siempre en el mismo sentido (en este caso hacia la izquierda). En un laberinto *con islas*, como el de la Imagen 11, el robot puede entrar en bucle y no salir nunca. El *problema del laberinto sin islas*, que aquí planteamos, consiste en encontrar un algoritmo que resuelva cualquier laberinto de este tipo y, posteriormente, programar el robot Mbot para que implemente dicho algoritmo.

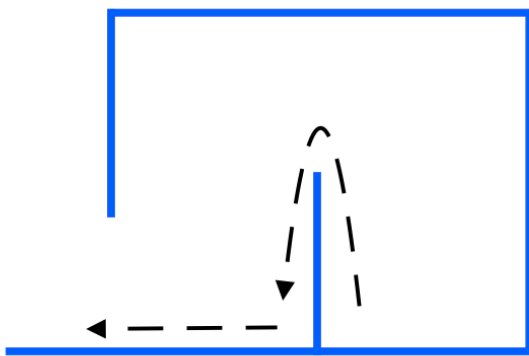


Imagen 10. El laberinto sin isla o bucle

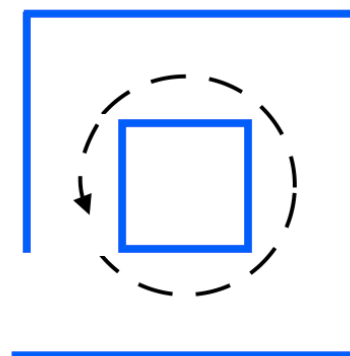


Imagen 11. El laberinto con isla o bucle

Este problema se resolvió también en dos fases. En primer lugar, se pidió a los alumnos estudiar distintos laberintos sin islas, observando que, si se gira siempre en un mismo sentido, antes o después dicho laberinto será resuelto. Una vez diseñada la

estrategia, se procedió a implementar el algoritmo en el Mbot. Esto requirió un conocimiento del lenguaje de programación, así como de las posibilidades que ofrece la combinación de los sensores. En este caso, se empleó un sensor de ultrasonidos, que fue situado en el mismo lado para el que gira el robot y un sensor sigue líneas, colocado en la parte frontal del mismo. El sensor de ultrasonidos controla la distancia lateral a la pared, mientras que el sensor sigue líneas de la parte delantera sirve para detectar y evitar un choque frontal con la pared. En la Imagen 12 se observa a una pareja de alumnos supervisando el comportamiento del Mbot en el laberinto construido por ellos mismos con trozos de polispán.

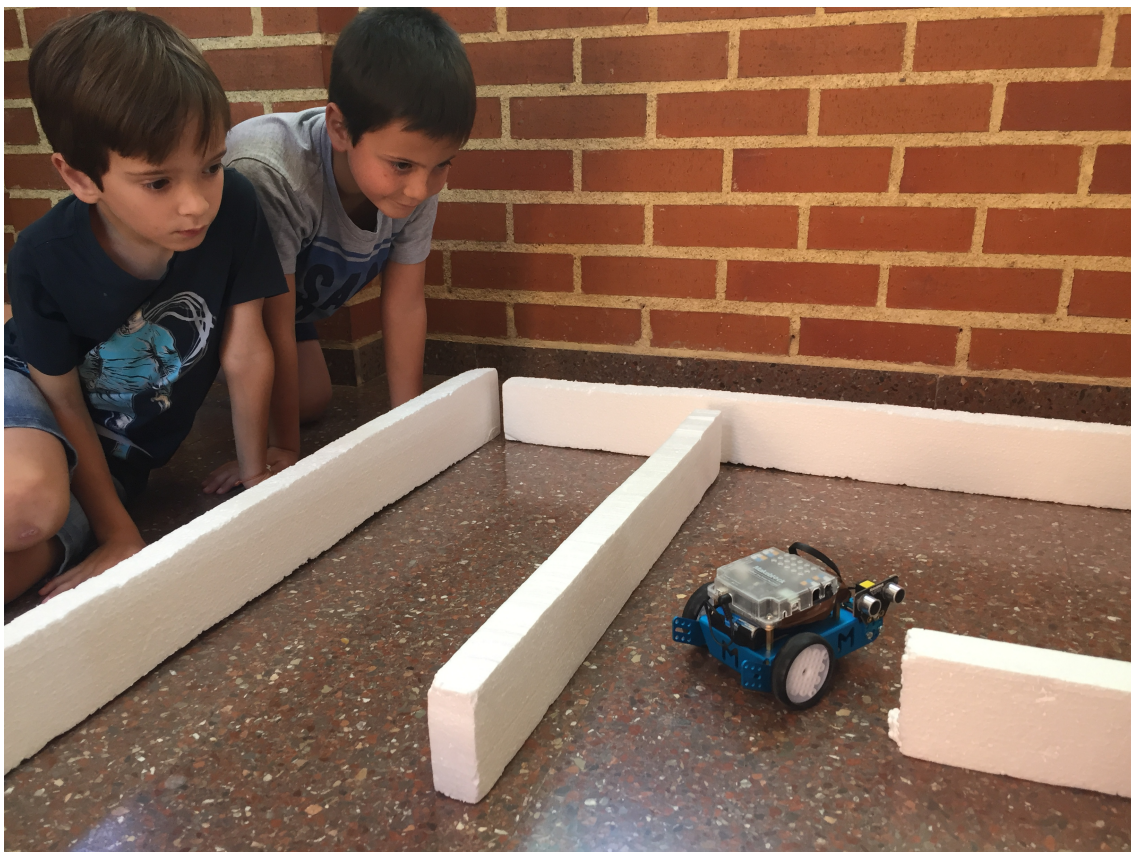


Imagen 12. Alumnos de sexto de primaria trabajando con el Mbot en un laberinto

4. Conclusiones

Las experiencias mostradas anteriormente pueden ser útiles a la hora de transformar la enseñanza tradicional en una metodología aplicable a las necesidades tecnológicas requeridas en el siglo XXI. Esta metodología, basada en el pensamiento computacional y la resolución de problemas, puede empezar a implementarse en los primeros cursos de educación primaria, e incluso en los últimos de educación infantil.

En concreto en este estudio se observó que las actividades propuestas son muy útiles para potenciar el aprendizaje colaborativo. En este sentido, los distintos miembros del grupo tuvieron que interactuar y compartir información e ideas para poder realizar la tarea con éxito. Fomentan la creatividad mediante el pensamiento divergente (Allen, Lewis y Fleming, 2017) ya que el alumnado disfruta de libertad de enfoque y genera soluciones diferentes para un mismo problema, que pueden incluso no ser previstas o contempladas a priori por el propio profesor.

El hecho de abrir el enfoque y generar varias soluciones permite, a su vez, que el alumnado trabaje la optimización o grado de ajuste de las soluciones. Cuando encuentra la solución al problema planteado, busca si esta es óptima, es decir, si no hay otras soluciones mejores según un criterio establecido de antemano. Para el problema de los bolos, por ejemplo, fue necesario asegurarse de que no había otro recorrido con un menor número de movimientos. Para el problema del laberinto la solución podría mejorarse simplificando, por ejemplo, el algoritmo que se implementaba en el robot para realizar la tarea requerida.

Agradecimientos

El proyecto STEMforYouth está financiado con fondos del programa de la Unión Europea —Promotion of Stem Education by Key Scientific Challenges and their impact on our Life and Career Perspectives, Horizon 2020- Seac-2015-1-710577.

Referencias

- Aho, A.V. (2012). Computation and computational thinking. *Computer Journal*, 55(7), 833-835. <https://doi.org/10.1093/comjnl/bxs074>
- Allen, Courtney; Lewis, Melanie A.; and Fleming, David (2017). Paint This Picture: Infusing Creativity Through Divergent Thinking Across All Content Areas. *National Youth-At-Risk Conference Savannah*. 35. https://digitalcommons.georgiasouthern.edu/nyar_savannah/2017/2017/35
- Blanco, T. F.; Salgado Somoza, M.; Gorgal Romarís, A. y Diego-Mantecón, J. M. (2017). La robótica en el aprendizaje de conceptos geométricos en Educación Primaria. *Actas XXX Congreso ENCIGA*. Galicia. España. pp. 103-104. ISSN:0214-7807
- Brzozowy, M., Hołownicka, K., Bzdak, J., Tornese, P., Lupiañez-Villanueva, F., Vovk, N., ... Moussas, X. (2017a). Making STEM education attractive for young people by presenting key scientific challenges and their impact on our life and career perspectives. In *INTED2017 Proceedings* (pp. 9948–9957). IATED. <https://doi.org/10.21125/inted.2017.2374>
- Clements, D. H., & Gullo, D. F. (1984). Effects of computer programming on young children's cognition. *Journal of Educational Psychology*, 76 (6), 1051-1058. doi: <http://dx.doi.org/10.1037/0022-0663.76.6.1051>
- Diego-Mantecón, J. M., Bravo, A., Arcera, O., Cañizal, P., Blanco, T. F., Recio, T., González-Ruiz, I. and Istúriz, M. P. (2017). Desarrollo de cinco actividades STEAM con formato KIKS. In *Proceedings of VIII CIBEM*. Madrid, Spain.
- Diego-Mantecón, J. M., Sáenz de la Torre, J.J., Brzozowy, M. (2017). Proyecto STEMforYouth. In *Proceedings of VIII CIBEM*. Madrid, Spain.
- Kelley, T. R. & Knowles, J. G. (2016). A conceptual framework for integrated STEM education. *International Journal of STEM Education*. 3:11. doi: <https://doi.org/10.1186/s40594-016-0046-z>
- Segredo, E., Gara, M. y León, C. (2017). Towards the Education of the Future: Computational Thinking as a Generative Learning Mechanism. *Education in the Knowledge Society (EKS)*, 18 (2), 33-58. doi: <https://dx.doi.org/10.14201/eks2017182335>