



MONOGRÁFICO: MATEMÁTICA, TECNOLOGÍA Y ENTORNOS DE ENSEÑANZA-APRENDIZAJE

Estudio exploratorio sobre lenguajes simbólicos de programación en tareas de resolución de problemas con Bee-bot

Gema Pérez Buj y *Pascual D. Diago Nebot

Departament de Didàctica de la Matemàtica. Universitat de València

PALABRAS CLAVE

Resolución de problemas
Pensamiento computacional
Robótica educativa
Programación

KEYWORDS

Problem-solving
Computational thinking
Educational robotics
Coding

RESUMEN

En los últimos años movimientos relacionados con la robótica educativa, el movimiento maker o el movimiento *STEM* han potenciado la integración en las aulas de contenidos de matemáticas con otros propios de las ciencias de la computación. El uso de entornos tecnológicos en educación, así como el resurgimiento de la programación en la escuela son tendencia consolidada en el panorama educativo. En este estudio pretendemos explorar la capacidad de estudiantes de infantil y primaria para elaborar esbozos de programas a partir de su lenguaje propio así como explorar la conveniencia del uso de un lenguaje de programación simbólico para resolver un problema. Así, desde la perspectiva de la resolución de problemas de matemáticas, observaremos actuaciones en las que el estudiante debe idear, generar, desplegar y gestionar estrategias que le permitan abordar el problema (con o sin éxito) con la restricción de que la solución obtenida debe poder implementarse en un entorno tecnológico.

Exploratory study on symbolic programming languages in problem-solving activities with Bee-bot

ABSTRACT

In recent years, movements related to educational robotics, the maker movement or the *STEM* movement have strengthened the integration of mathematics contents with other ones of computer science. The use of technological environments in education, as well as the resurgence of school programming are a consolidated trend in the educational panorama. In this study we intend to explore the ability of kindergarten and primary school students to prepare program sketches from their own language as well as explore the convenience of using a symbolic programming language to solve a problem. Thus, from the perspective of solving mathematical problems, we will observe actions in which the student must devise, generate, deploy and manage strategies that allow him to approach the problem (with or without success) with the restriction that the solution obtained must be implemented in a technological environment.

Universitat de València

Autor de correspondència: * Pascual D. Diago Nebot. E-Mail: Pascual.Diago@uv.es. Departament de Didàctica de la Matemàtica. Facultat de Magisteri - Campus de Tarongers. Avda. Tarongers, 4, 46022 València (Espanya). +34 963 983 293

Recibido el 01/06/2018 - Aceptado el 03/08/2018

Introducción

El fuerte desarrollo de la tecnología, y de los ordenadores en particular, permitió en las décadas de los 60 y 70 del pasado siglo un acercamiento de la programación a las escuelas. Desde que LOGO o BASIC empezaran a utilizarse en el aula (Clements y Sarama, 1997; Noss, 1985; Papert, 1981) hasta la actualidad, con la fuerte presencia de lenguajes de programación por bloques (con Scratch¹ como máximo representante), las investigaciones acerca de las ventajas e implicaciones cognitivas de la programación en edades escolares no han hecho más que crecer. En lo que concierne a la educación matemática, desde hace tiempo se incide en que las tareas propias de las ciencias de la computación favorecen los procesos de razonamiento matemático y las habilidades en resolución de problemas (Clements y Sarama, 2002; Hatfield y Kieren, 1972; Hoyles y Lagrange, 2010; Kaput, 1992; Papert, 1972; Shute, Sun, Asbell-Clarke, 2017). De forma paralela, los enfoques y paradigmas con los que estudiar la interacción de la tecnología con los procesos de enseñanza-aprendizaje de las matemáticas escolares se han ido diversificando (Drijvers y cols., 2010; Kaput, 1992).

La aparición de movimientos relacionados con la *robótica educativa* (Barker y Ansorge, 2007), como el movimiento *maker*, el *internet de las cosas* (*Internet of Things*, abreviado *IoT* en inglés) o el movimiento *STEM* (*Science, Technology, Engineering and Mathematics*, de sus siglas en inglés) han potenciado en los últimos años la integración de contenidos de matemáticas y ciencias con aspectos propios de la tecnología, la ingeniería y la computación en las aulas (Grover y Pea, 2013).

Entornos tecnológicos en educación matemática y programación en bloques

El desarrollo tecnológico y su facilidad de integración a todos los niveles han derivado en un auge del uso de la tecnología en el aula en muchos de los contextos diarios. No es diferente el caso de la educación matemática, que se ha visto rodeada por un nuevo ecosistema basado en diferentes entornos tecnológicos adaptados a niveles escolares (Aldon, Hitt, Bazzini y Gellert, 2017; Clark-Wilson, Robutti, y Sinclair, 2014; Hoyles y Lagrange, 2010). En especial, los entornos tecnológicos basados en *programación en bloques* (ya sean robots, software o simuladores) son tendencia consolidada en el panorama educativo (Benton et al., 2017; Fessakis, Gouli y Mavroudi, 2013; Kazakoff, Sullivan y Bers, 2013; Leidl, Bers y Mihm, 2017; Sáez y Cózar, 2017; Sáez-López, Román-González, y Vázquez-Cano, 2016; Sullivan y Bers, 2016). Estos entornos, exportados de las ciencias de la computación, necesitan de un programa (secuencia de instrucciones a ejecutar) y de un lenguaje de programación (sintaxis o vocabulario propio que el entorno tecnológico es capaz de entender) para poder interactuar con el usuario. Las particularidades de la programación en bloques son i) que las instrucciones para elaborar un programa se organizan en bloques de órdenes pre-programadas que pueden secuenciarse una detrás de otra directamente; y ii) que la interacción con el usuario se realiza mediante lenguaje natural. Así, pese a que los entornos de programación basados en bloques son percibidos por los estudiantes como menos potentes y auténticos (Weintrop y Wilensky, 2015), el uso de bloques aporta entornos listos para ser explorados por los estudiantes con la transparencia y la facilidad de uso de poder realizar manipulaciones directamente sobre la interfaz del entorno tecnológico.

Resolución de problemas y pensamiento computacional

Nuestro marco instrumental se fundamenta en el uso de los entornos tecnológicos de programación en bloques para el

aprendizaje de la resolución de problemas de matemáticas a través del llamado *pensamiento computacional* (Wing, 2006). Este término lo entendemos como proceso de resolución de problemas en el que el estudiante debe idear, generar, desplegar y gestionar estrategias que le permitan abordar la tarea (con o sin éxito) con la restricción de que la solución obtenida debe poder implementarse en dicho entorno tecnológico. Este enfoque no se centra en el fomento de la competencia digital ni en los aspectos instrumentales de la herramienta tecnológica, sino que pretendemos que el estudiante tome consciencia de los procesos que la herramienta tecnológica es capaz de llevar a cabo. Esto nos sitúa en una perspectiva de la enseñanza de la resolución de problemas independiente del contenido, calificada como *heurística matemática* (en el sentido de Polya, 1945) o como *pura resolución de problemas* (en el sentido de Puig, 1996). Esta perspectiva pone de manifiesto el interés por enseñar a resolver problemas y define la propia tarea como un escenario privilegiado para el aprendizaje (Puig y Cerdán, 1988). Así, se enfrentará al estudiante a situaciones en entornos no familiares, que deberá ser capaz de gestionar para elaborar un plan que le permita hacer frente al problema planteado.

Como se ha puesto de manifiesto en Diago, Arnau y González-Calero (2018) o Puig (2018), las tareas en las que los estudiantes deben programar robots mediante secuencias de bloques se convierten típicamente en situaciones problemáticas ya que son capaces de entenderlas, pero no disponen de un medio directo para su realización. La naturaleza matemática del problema la podemos identificar en las ideas puestas en juego a la hora de organizar los fenómenos presentes que podría formalizarse en un nivel matemático superior. Así, por ejemplo, a la hora de pensar en el programa que un robot debe ejecutar, la elaboración de la secuencia de acciones a realizar y la representación de las mismas con los medios del estudiante (esquema, lenguaje verbal, sistema de signos, etc.) podría formalizarse matemáticamente como la elaboración de un programa en pseudo-código (creado por el propio alumno). De igual forma, a la hora de introducir dicho programa en el robot, será pertinente que el estudiante hable el mismo lenguaje (lenguaje de programación) que el robot, por lo que deberá aprender o interpretar dicho lenguaje para poder comunicarse de forma eficiente con el entorno tecnológico y que éste realice aquello que el estudiante está pensando. Desde este punto de vista, no son nuevos los estudios que plantean cómo los estudiantes aprenden nuevos lenguajes de programación y su relación con diferentes aspectos de las matemáticas (Chen y cols., 2017; Fessakis, Gouli y Mavroudi, 2013; Milojkovic, 1984; Noss, 1985; Shute, Sun, y Asbell-Clarke, 2017; por citar algunos ejemplos).

Objetivo

Para los estudiantes de infantil y primaria nos planteamos los siguientes objetivos de investigación:

- 1) Explorar la capacidad para esbozar y elaborar programas mediante su propio lenguaje (gestual, verbal o de signos).
- 2) Explorar las limitaciones y ventajas de un lenguaje simbólico de programación por bloques (las tarjetas de comandos); en concreto, determinar si problemas que no han podido ser resueltos sin hacer uso del lenguaje simbólico de las tarjetas pueden resolverse haciendo uso de ellas.

Material

El robot Bee-bot

Bee-bot es un robot adaptado a primeras edades escolares en cuanto a su sencillez y uso. Es clasificado como un interfaz tangible para el usuario (*Tangible User Interfaces - TUI* en inglés) en la taxonomía de Strawhacker y Bers (2015) debido a que la inte-

¹ <https://scratch.mit.edu/>

racción con el robot se realiza únicamente mediante los botones físicos situados en su parte superior. Como se puede ver en la Figura 1, todos los bloques de programación disponibles tienen que ver con el movimiento del robot, atendiendo a su propio sis-

tema de referencia. Es importante hacer notar que los bloques de giro (a derecha o izquierda) corresponden a giros de 90° sobre sí mismo (en sentido horario o anti-horario), sin que el robot se traslade.



Figura 1. El robot Bee-bot y sus botones de programación

Dado que el movimiento del robot está estipulado en 15cm, las tareas escolares en las que se hace uso de Bee-bot suelen realizarse sobre tableros o escenarios con una cuadrícula de 15cm de lado sobre los que se desplazará el robot (ver Figura 2). Todos los

problemas abordados en este estudio consistirán en llevar a Bee-bot desde una posición inicial a otra final (identificada con una flor) siguiendo un camino dado marcado en color rojo.

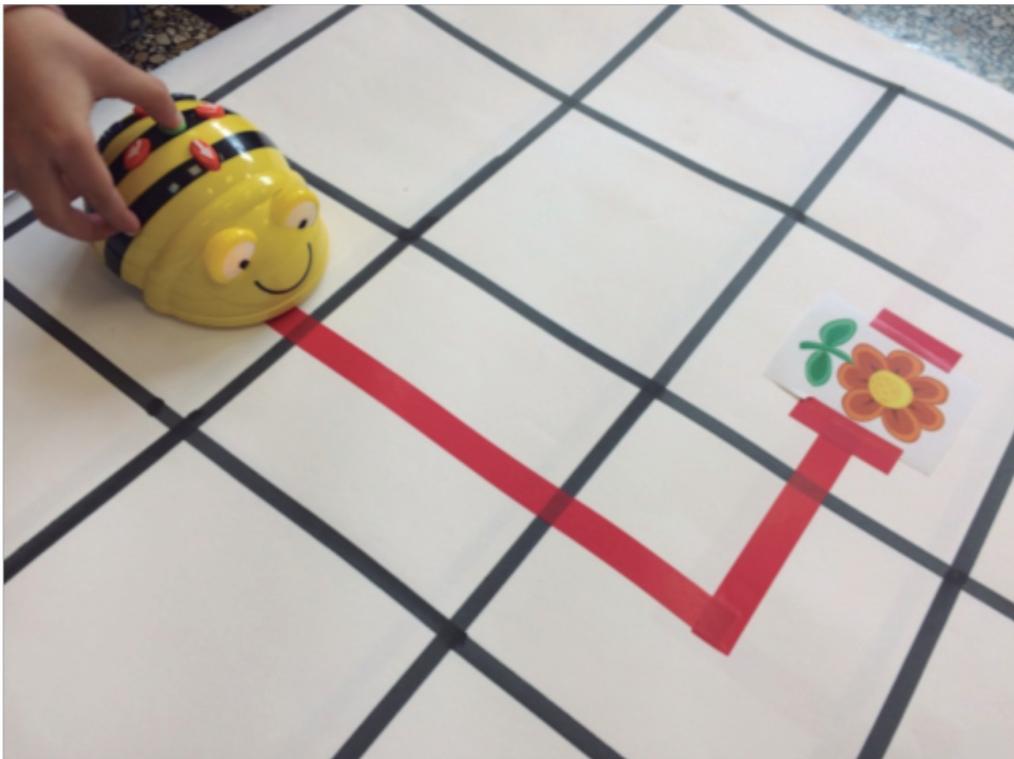


Figura 2. Vista principal de un problema presentado a los estudiantes de este estudio

Las tarjetas y la caja de secuenciación

Con el fin de determinar las limitaciones y ventajas de un lenguaje de programación simbólico de bloques que permita programar al robot Bee-bot, en el sentido propio de las ciencias de la computación, se hará uso de un sistema de tarjetas en el cual se

representa cada uno de los bloques (o instrucciones) que puede ejecutar el robot (Figura 3). A los estudiantes se les facilitará un espacio físico, al que denominamos *caja de secuenciación*, en el que disponer las tarjetas en el orden en que se introducirán posteriormente dichos comandos en el robot (Figura 4).



Figura 3. Tarjetas de programación correspondientes a los movimientos de Bee-bot



Figura 4. Ejemplo de uso del lenguaje simbólico de las tarjetas de instrucciones y de la caja de secuenciación en una tarea con el robot Bee-bot

Tanto la caja como las tarjetas actuarán a modo de destreza heurística (Diago, Arnau y González-Calero, 2018; Puig, 1996), en el sentido en que será un medio de representación de la trayectoria que el robot ha de seguir, sin que ello suponga un éxito en la resolución del problema.

Las variables de tarea y el diseño de los problemas

Dado que se trata de un estudio sobre resolución de problemas que involucran a estudiantes resolviendo tareas en un deter-

minado contexto, será necesario definir las siguientes variables relacionadas con la tarea (Kilpatrick, 1978):

- Número de tarjetas (bloques) para completar la trayectoria
- Número de giros a realizar para completar la trayectoria
- Las veces que el sistema de referencia del robot se desliga del sistema de referencia del estudiante a lo largo de la trayectoria

Teniendo en cuenta estas variables de tarea se diseñan diez problemas, mostrados en la Tabla 1, graduados en complejidad.

Tabla 1. Problemas elaborados en base a las variables de tarea descritas en el estudio

T01	T02	T03	T04	T05
T06	T07	T08	T09	T10

En el diseño de los problemas se deciden fijar algunas variables. En concreto, aquella que da cuenta de la posición de los estudiantes para resolver la tarea. Así, no se permitirá que los resolutores se muevan a lo largo del tablero, teniendo que permanecer en la zona que se corresponde con la parte inferior de los tableros mostrados en la Tabla 1. Como se pretende estudiar la capacidad para elaborar y pensar un plan de resolución a modo de secuencia de instrucciones, tras la realización de cada intento de resolución (con o sin tarjetas), el robot *Bee-bot* se vuelve a colocar en la posición inicial. Para poder estudiar el uso, manejo y conveniencia del lenguaje de programación de las tarjetas de comandos, después de ejecutar un plan fallido, se dejarán en la caja de secuenciación las tarjetas secuenciadas para que el estudiante decida si empezar una nueva secuencia o modificar la secuencia previa. Además, tampoco se permitirá realizar resoluciones parciales del problema (con o sin tarjetas), pese a que se dejará a los estudiantes hacer uso de esta herramienta heurística (Diago, Arnau y González-Calero, 2018; Puig, 1996) si aparece de forma natural en su discurso argumentativo.

Método

Participantes

El estudio que aquí se describe se llevó a cabo con siete parejas de estudiantes de un centro concertado del sistema educativo de la Comunitat Valenciana. Las características y codificación para cada una de las parejas se describen en la Tabla 2.

Tabla 2. Codificación y descripción de los estudiantes de cada pareja participante en el estudio

Pareja	ID de estudiante	Edad (años; meses)
5P1	5P1.a	5;8
	5P1.b	5;7
5P2	5P2.a	5;7
	5P2.b	5;6
6P1	6P1.a	6;6
	6P1.b	6;8
6P2	6P2.a	6;10
	6P2.b	7;1
6P3	6P3.a	6;9
	6P3.b	6;11
10P1	10P1.a	10;2
	10P1.b	10;1
10P2	10P2.a	10;1
	10P2.b	10;1

La elección de los estudiantes no respondió a ningún criterio concreto puesto que se trata de un estudio exploratorio sobre el uso y conveniencia de los lenguajes de programación en edades escolares. En consecuencia, se aceptó como válida la selección de alumnos realizada por las tutoras de cada clase. Todos los estudiantes confirmaron que no tenían ninguna experiencia previa con el robot *Bee-bot*.

Procedimiento

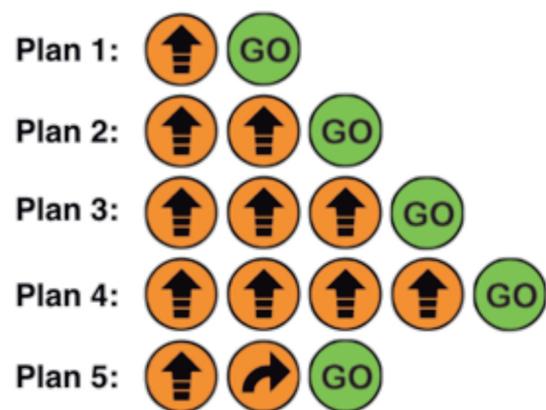
Los datos se obtuvieron al enfrentar las parejas de estudiantes arriba mencionadas a la resolución de la colección problemas descritos. Las parejas de estudiantes resolvían la tarea de forma simultánea, y se les pedía que colaboraran para planificar, organizar e idear un plan que resolviera el problema planteado. Ambos alumnos participaban del proceso de resolución y podían intervenir en cualquier momento, pues no se establecían

turnos de intervención. Las sesiones se grabaron en vídeo y se transcribieron posteriormente a un protocolo escrito. Dado que se pretende observar la resolución de problemas, se intentó que el grado de intervención de los investigadores fuera muy bajo (Schoenfeld, 1985). Sin embargo, no pudo ser inexistente, pues se tuvo que intervenir en las situaciones típicas asociadas al manejo de *Bee-bot* (por ejemplo, recordar la necesidad borrar el plan previo en el robot) o para remediar la ausencia de comunicación o los diálogos inaudibles propios de los estudiantes de las etapas educativas involucradas.

Para la consecución de los objetivos de investigación se diseñaron tres experimentos, con dos fases previas de enseñanza. La Tabla 3 muestra un cuadro resumen del conjunto de la planificación. A continuación, aportamos una breve justificación del diseño experimental:

Fase 1 (enseñanza). La primera de las fases, de enseñanza, se utilizó para familiarizar a los estudiantes con el entorno tecnológico *Bee-bot* y con los modelos de problemas, presentados en tableros. Para ello, el investigador ejecutó las secuencias de instrucciones (llamadas *planes* en lo que sigue) mostradas en la Figura 5 (directamente secuenciadas en el robot *Bee-bot*, sin hacer uso de tarjetas de comandos) y se observó y comentó el movimiento del robot.

Figura 5. Planes utilizados en la fase de enseñanza con las parejas de estudiantes



Fase 2 (experimento 1). El primero de los experimentos tiene como fin saber qué problemas pueden resolver los estudiantes y cuales no, haciendo un uso directo del robot *Bee-bot*. Con esto se pretende obtener una idea de la relación entre la complejidad del problema y la dificultad experimentada por los resolutores. En esta fase no hay interacción con las tarjetas de comandos. Con algunos de los problemas que consiguen resolver se pasa a la Fase 3 (experimento 2, no con todos por falta de tiempo, ya que los estudiantes se muestran cansados tras un periodo superior a 30 minutos).

Fase 3 (experimento 2). El segundo de los experimentos se lleva a cabo con algunos de los problemas que los estudiantes consiguen resolver. El fin es explorar la capacidad de los estudiantes para elaborar representaciones, a modo de esbozos de programa, que conecten el mundo real que perciben con el aspecto computacional y la resolución de problemas por medio del análisis de su lenguaje verbal, gestual o de los signos creados por ellos mismos en papel y lápiz. Para ello, se les presenta alguno de los problemas que hayan podido resolver y se les da la consigna *¿Cómo explicarías a tu compañero las instrucciones que has de darle a Bee-bot para que llegue a la flor?*. Se les facilitan hojas y lápices para que esbocen o representen sus ideas, y se les indica que pueden hacer uso del lenguaje escrito, de esquemas, dibujos o cualquier tipo de representación que se les ocurra.

Fase 4 (enseñanza). Esta fase corresponde a la enseñanza de las tarjetas de comandos y de la caja de secuenciación. Se presentan ambos materiales y se profundiza en el uso y la comprensión de las tarjetas de comandos, que simbolizan el movimiento del robot por el tablero. Para ello se configuran algunos de los planes de la Fase 1 en la caja de secuenciación con las tarjetas correspondientes, se ejecutan en *Bee-bot* y se comenta lo sucedido.

Fase 5 (experimento 3). En este último experimento se les presentan algunos de los problemas que no han conseguido resolver en la Fase 2 (experimento 1) y se les pide que intenten resolverlo nuevamente haciendo uso de las tarjetas de comandos. Por cuestiones de tiempo, se eligen solamente algunos de ellos. Igualmente, en algún caso puntual se elige un problema que sí que había sido resuelto en las fases anteriores, pero que había presentado dificultades.

Tabla 3. Resumen del desarrollo de las fases llevadas a cabo para este estudio exploratorio

Fase	Propósito	Objetivo	Desarrollo
1-Enseñanza	Familiarizar con <i>Bee-bot</i>	-	Se presenta el robot <i>Bee-bot</i> , sin hacer referencia a las tarjetas de comandos. Se explican los bloques situados en el propio robot y se les permite interactuar con él.
2-Experimental	Identificación complejidad-dificultad	-	Se presentan los problemas T01 a T10, graduados en complejidad. No se hace uso de las tarjetas ni de la caja de secuenciación. Si no saben resolver algún problema se pasa al siguiente.
3-Experimental	Creación de "programas"	O1	Se presenta alguno de los problemas que hayan podido resolver . Se les facilitan hojas y lápices para que esbocen o representen sus ideas.
4-Enseñanza	Familiarizar con tarjetas y caja secuenciación	-	Se presentan tanto las tarjetas de comandos como la caja de secuenciación haciendo uso de los planes de la Fase 1.
5-Experimental	Uso del lenguaje de programación de las tarjetas	O2	Se les presentan problemas que no han conseguido resolver y se procede igual que en la fase anterior para ver la influencia del sistema de tarjetas en la resolución del problema.

Es importante hacer notar que durante todas las fases de experimentación, para cada problema, se insta a los estudiantes a que discutan el plan que debe seguir el robot *Bee-bot* para completar el camino mostrado previamente a manipular el robot. En la Fase 5 (experimento 3) se exige que los estudiantes organicen un plan previo usando las tarjetas de comandos correspondientes en la caja de secuenciación para, posteriormente, introducir dichas órdenes en el robot y comprobar si el plan pensado resuelve o no el problema presente en el tablero.

Resultados

Resultados del experimento 1 (Fase 2)

La Tabla 4 muestra un resumen de los resultados del primero de los experimentos, en el que los estudiantes resolvían los problemas de llevar a *Bee-bot* por una trayectoria dada sin hacer uso de las tarjetas de comandos, programando directamente las instrucciones en el robot. En la tabla resumen se indica con un número la cantidad de intentos necesarios para resolver el problema, X indica que el problema no pudo ser resuelto después de los intentos mostrados entre paréntesis, el signo - indica que el problema no fue abordado en esta fase y * indica que el problema se utilizó para la Fase 5 (experimento 3).

Tabla 4. Número de intentos necesarios para resolver cada uno de los problemas durante la Fase 2 (experimento 1)

ID	Problema									
	T01	T02	T03	T04	T05	T06	T07	T08	T09	T10
5P1	1	2	3	2	1	X(4)*	-	*	-	-
5P2	2	1	3	2	4	X(4)*	-	-	-	-
6P1	1	1	9*	4	2*	*	-	-	-	-
6P2	1	1	2	2	2	3	2	X(3)*	-	*
6P3	1	1	4	1	4	X(4)*	1	3	-	*
10P1	1	1	1	1	-	2	1	1	1*	X(2)*
10P2	1	1	1	2	1	1	1	1*	.*	.*

Resultados del experimento 2 (Fase 3)

En el Anexo 1 se muestran las Tablas 5, 6, 7 y 8 que recogen las evidencias elaboradas por los estudiantes indicados durante el experimento 2 (Fase 3). En esta fase se pidió a los estudiantes que elaboraran con sus propios medios un conjunto de instrucciones que reflejaran el comportamiento del robot *Bee-bot* con el fin de que se consiguiera, con ellas, resolver el problema.

Resultados del experimento 3 (Fase 5)

En el Anexo 2 se recogen las Tablas 9 a 14, en las que se reflejan los planes elaborados por los estudiantes para la resolución de los problemas marcados con * en la Tabla 4. En esta ocasión los estudiantes hicieron uso tanto del sistema de tarjetas, a modo de lenguaje simbólico de programación, como de la caja de secuenciación para secuenciar los bloques necesarios a ejecutar por

el robot *Bee-bot* para resolver el problema. En las Tablas indicadas se utiliza la notación siguiente para hacer referencia a los bloques de programación de las tarjetas: ↑ (adelante), ↓ (atrás), > (giro de 90° a la derecha), < (giro de 90° a la izquierda) y GO (ejecutar órdenes en el robot *Bee-bot*).

Discusión

En referencia al experimento 1, se observa que el número de intentos (Tabla 4) aumenta significativamente en aquellos problemas que involucran giros, de forma especial en los estudiantes de 5 y 6 años. Los estudiantes de 10 años no parecen experimentar dificultad en relación a los giros. Así mismo, se observa como los estudiantes de 10 años pueden resolver, en el mismo tiempo, más problemas de este tipo, pues la dificultad experimentada se advierte menor.

Con respecto al experimento 2, los esbozos de programas ideados por estudiantes de 5 años contienen una carga mayor de significado “espacial” de la situación problemática (como observamos en las producciones 5P1a y 5P1b de las Tablas 5 y 6, respectivamente). Los estudiantes de 6 años, en menor proporción, también dan muestras de instrucciones ligadas a la representación espacial de la situación (6P2a de la Tabla 7). A pesar de ello, alumnos de estas mismas edades son capaces de realizar un paso de abstracción más, mostrando flechas en el sentido de los botones de movimiento manipulados y utilizados al mover al robot. En el caso de 6P1 (Tabla 5) el programa se realiza con lenguaje verbal. Es destacable el hecho que en edades tempranas, 5 y 6 años, ya aparezcan de forma natural signos referentes a las repeticiones de instrucciones a modo de “acción reiterada” (5P1b y 5P2a, Tabla 5), al orden en las secuencias de comandos (5P2b y 6P1b, Tabla 7) o instrucciones relativas al fin de programa (6P3a, Tabla 6). Esto, podría ser debido al aspecto físico del robot y a un pensamiento de tipo “iterativo” (como se deriva de la producción 6P1a de la Tabla 5). En las elaboraciones de los estudiantes de 10 años se observa una mayor capacidad para elaborar programas más largos, con más instrucciones, y una mejor representación de la estructura secuencial propia de lenguajes de programación más formales (10P1a y 10P1b, Tabla 8). Con todo, siguiendo la línea de estudios como los de Kazakoff, Sullivan y Bers (2013) o Sáez-López, Román-González y Vázquez-Cano (2016), concluimos que las tareas de resolución de problemas basadas en el uso del pensamiento computacional ayudan en el aprendizaje y desarrollo de conceptos básicos ligados a la programación visual, la secuenciación, la generalización y la abstracción.

Por último, pasamos a discutir los resultados del experimento 3, en el que se evalúa la conveniencia o no de un lenguaje de programación simbólico de tarjetas. Si comparamos dos a dos los problemas abordados en los experimentos 2 (sin tarjetas) y 3 (con tarjetas) podemos observar como el uso del lenguaje de programación de las tarjetas permite a los estudiantes resolver problemas que sin ellas no habían podido resolver; así es el caso de los problemas 5P1-T06, 6P2-T08, 6P3-T06 y 10P1-T10 (Tablas 9, 11, 12 y 13, respectivamente). En los estudiantes de 5 y 6 años el uso del lenguaje de programación de las tarjetas parece mejorar cualitativamente sus procesos de razonamiento relacionados con las matemáticas y la lógica (como apuntan Fessakis, Gouli y Mavroudi, 2013; Sáez-López, Román-González y Vázquez-Cano, 2016), pues les permite plantear una secuencia de instrucciones, comprobarla con el robot y modificar, en su caso, los pasos incorrectos.

Consideraciones finales

Somos conscientes de que este estudio tiene limitaciones importantes al tratarse de un estudio exploratorio, pero a su vez, el análisis cualitativo inicial de las actuaciones de los estudiantes nos ha permitido detectar puntos de interés en los que preten-

demos profundizar en un futuro. Con respecto al primero de los objetivos se concluye que ya los estudiantes de 5 años son capaces de pasar de una representación espacial del problema (dibujo o esquema) a un programa elaborado mediante lenguaje verbal, gestual o de signos que contiene el plan para obtener la solución a problemas consistentes en tareas de programación. Estudiantes de 5 y 6 años ya incluyen características propias de los lenguajes de programación en estas producciones idiosincrásicas (como son las iteraciones o la indicación del fin de programa). Existe una diferencia cualitativa, en referencia a la capacidad para elaborar programas más complejos, en estudiantes de 10 años. Con respecto a las ventajas e inconvenientes de usar un lenguaje de programación basado en tarjetas, podemos concluir que en estudiantes de 5 y 6 años el uso de este lenguaje simbólico es crucial a la hora de enriquecer el proceso de resolución del problema. En casi todas las parejas el uso de este lenguaje de programación simbólico permite obtener la solución de problemas que anteriormente (sin las tarjetas) no habían podido ser resueltos, traduciéndose en una ayuda real a la resolución de este tipo de tareas que les permite pensar de forma más estructurada, evaluar y modificar los planes pensados. En cambio, para estudiantes de 10 años este lenguaje puede llegar a derivar en superfluo, pues perciben de forma natural los botones de movimiento del robot *Bee-bot* como bloques de órdenes que pueden ser secuenciados. Aun así, es necesario un estudio sistemático más profundo que permita determinar de forma más precisa el alcance e implicación del uso de un lenguaje simbólico de programación basado en bloques en estas edades.

Las tareas de resolución de problemas de programación, basadas en entornos tecnológicos del tipo robot *Bee-bot*, permiten a los estudiantes desarrollar su competencia en resolución de problemas. Gracias al *feedback* inmediato que el robot ofrece con su movimiento el estudiante puede modificar la solución ejecutada y pensar un nuevo plan para comprobarlo en un nuevo intento mediante el movimiento del robot. Además, los lenguajes de programación visual por bloques permiten al estudiante pensar en los errores cometidos y modificar el plan inicial de acuerdo a la actuación mostrada por el robot. Así, este tipo de propuestas contribuyen a la adquisición propia de contenidos de matemáticas a la vez que intentan conectar aprendizajes basados en el enfoque STEM, fomentando el interés y la motivación del alumno por la tecnología y la ingeniería.

Agradecimientos

Este trabajo ha contado con el soporte del proyecto EDU2017-84377-R (MINECO/FEDER).

Referencias

- Aldon, G., Hitt, F., Bazzini, L., y Gellert, U. (Eds.). (2017). *Mathematics and Technology*. Cham, Switzerland: Springer International Publishing.
- Barker, B. S., y Ansorge, J. (2007). Robotics as Means to Increase Achievement Scores in an Informal Learning Environment. *Journal of Research on Technology in Education*, 39(3), 229–243. <http://doi.org/10.1080/15391523.2007.10782481>
- Benton, L., Hoyles, C., Kalas, I., y Noss, R. (2017). Bridging Primary Programming and Mathematics: Some Findings of Design Research in England. *Digital Experiences in Mathematics Education*, 23–29.
- Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., y El-toukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers and Education*, 109, 162–175. <http://doi.org/10.1016/j.compedu.2017.03.001>
- Clark-Wilson, A., Robutti, O., y Sinclair, N. (Eds.). (2014). *The Mathematics Teacher in the Digital Era. An International Perspec-*

- tive on Technology Focused Professional Development. Dordrecht: Springer. <http://doi.org/10.1007/978-94-007-4638-1>
- Clements, D. H., y Sarama, J. (1997). Research on Logo: a decade of progress. *Computers in the Schools*, 14(1), 9–46. <http://doi.org/10.1300/J025v02n02>
- Clements, D. H., y Samara, J. (2002). The Role of Technology in Early Childhood Learning. *Teaching Children Mathematics*, 8(6), 340–343.
- Diago, P. D., Arnau, D., y González-Calero, J. A. (2018). Elementos de resolución de problemas en primeras edades escolares con Bee-bot. *Edma 0-6: Educación Matemática En La Infancia*, 7(1), 12–41. <http://www.edma0-6.es/index.php/edma0-6/article/view/49>
- Drijvers, P., Kieran, C., Mariotti, M.-A., Ainley, J., Andresen, M., Cheung, Y., Dana-Picard, T., Gueudet, G., Kidron, I., Leung, A. y Meagher, M. (2010). Integrating Technology into Mathematics Education: Theoretical Perspectives. En C. Hoyles y J.-B. Lagrange (Eds.), *Mathematics Education and Technology-Rethinking the Terrain* (pp. 89–132). New York: Springer. https://doi.org/10.1007/978-1-4419-0146-0_7
- Fessakis, G., Gouli, E., y Mavroudi, E. (2013). Problem solving by 5-6 years old kindergarten children in a computer programming environment: A case study. *Computers and Education*, 63, 87–97. Retrieved from <http://dx.doi.org/10.1016/j.compedu.2012.11.016>
- Grover, S., y Pea, R. (2013). Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher*, 42(1), 38–43. <http://doi.org/10.3102/0013189X12463051>
- Hatfield, L. L., y Kieran, T. E. (1972). Computer-Assisted Problem Solving in School Mathematics. *Journal for Research in Mathematics Education*, 3(2), 99–112.
- Hoyles, C., y Lagrange, J.-B. (Eds.). (2010). *Mathematics Education and Technology-Rethinking the Terrain: The 17th ICMI Study*. New York: Springer.
- Kaput, J. J. (1992). Technology and Mathematics Education. En D. A. Grouws (Ed.), *Handbook of research on mathematics teaching and learning* (pp. 511–556). Reston, VA: NCTM.
- Kazakoff, E. R., Sullivan, A., y Bers, M. U. (2013). The Effect of a Classroom-Based Intensive Robotics and Programming Workshop on Sequencing Ability in Early Childhood. *Early Childhood Education Journal*, 41(4), 245–255. <http://doi.org/10.1007/s10643-012-0554-5>
- Kilpatrick, J. (1978). Variables and methodologies in research on problem solving. En L. L. Hatfield (Ed.), *Mathematical problem solving: papers from a research workshop*. ERIC.
- Leidl, K. D., Bers, M. U. y Mihm, C. (2017). Programming with ScratchJr: a review of the first year of user analytics. En S. C. Kong, J. Sheldon y K. Y. Li (Eds.), *Conference Proceedings of International Conference on Computational Thinking Education 2017* (pp. 116–121). Hong Kong: The Education University of Hong Kong.
- Milojkovic, J. D. (1984). *Children learning computer programming: cognitive and motivational consequences*. Stanford University.
- Noss, R. (1985). *Creating a mathematical environment through programming: a study of young children learning LOGO*. University of London.
- Papert, S. (1972). Teaching Children to Be Mathematicians versus Teaching About Mathematics. *International Journal of Mathematics Education and Science Technology*, 3, 249–262.
- Papert, S. (1981). *Mindstorms - Children, Computers and Powerful Ideas*. New York: Basic Books.
- Polya, G. (1945). *How to Solve It*. Princeton, NJ: Princeton University Press.
- Puig, L. (1996). *Elementos de resolución de problemas*. Granada: Comares.
- Puig, L. (Marzo de 2018). Tareas de programación en edades tempranas vistas como tareas de resolución de problemas. En P. D. Diago (presidente del SOC), *1er Workshop sobre Entornos Tecnológicos en Educación Matemática (ETEM)*. Facultat de Magisteri, Universitat de València, València.
- Puig, L. y Cerdán, F. (1988). *Problemas aritméticos escolares*. Madrid: Síntesis.
- Sáez, J. M., y Cózar, R. (2017). Pensamiento computacional y programación visual por bloques en el aula de Primaria. *Educación*, 53(1), 129–146.
- Sáez-López, J. M., Román-González, M., y Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using “scratch” in five schools. *Computers and Education*, 97, 129–141. <http://doi.org/10.1016/j.compedu.2016.03.003>
- Schoenfeld, A. H. (1985). *Mathematical Problem Solving*. Academic Press: Orlando, FL.
- Shute, V. J., Sun, C., y Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142–158. <http://doi.org/10.1016/j.edurev.2017.09.003>
- Strawhacker, A., y Bers, M. U. (2015). “I want my robot to look for food”: Comparing Kindergarten’s programming comprehension using tangible, graphic, and hybrid user interfaces. *International Journal of Technology and Design Education*, 25(3), 293–319. <http://doi.org/10.1007/s10798-014-9287-7>
- Sullivan, A. y Bers, M. U. (2016). Robotics in the early childhood classroom: learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. *International Journal of Technology and Design Education*, 26(1), 3–20.
- Weintrop, D., y Wilensky, U. (2015). To Block or not to Block, That is the Question: Students’ Perceptions of Blocks-based Programming. En *Proc. IDC '15. ACM* (pp. 199–208). <http://doi.org/10.1145/2771839.2771860>
- Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33–35. <http://doi.org/10.1145/1118178.1118215>

ANEXO 1

Tabla 5. Representaciones de los estudiantes indicados para los problemas T01 y T02

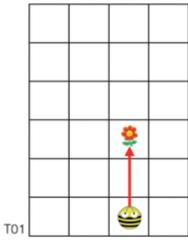
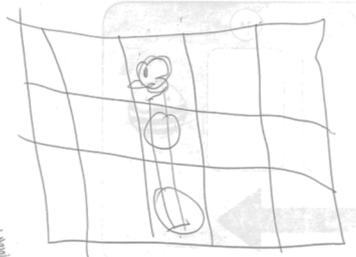
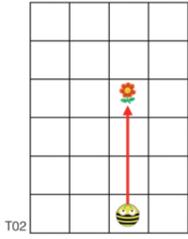
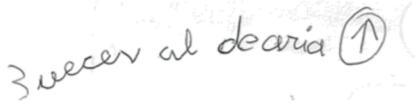
T01		
		
	5P1a	5P1b
T02		
		
	5P2a	6P1a

Tabla 6. Representaciones de los estudiantes indicados para el problema T03

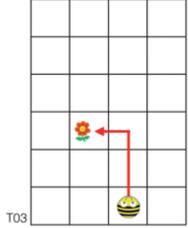
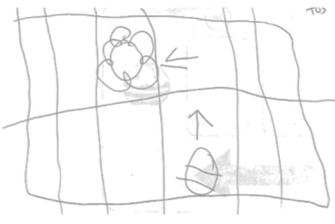
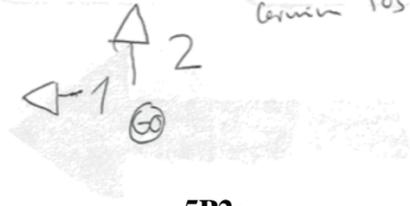
T03		
		
	5P1b	5P2a
		
	6P3a	10P2b

Tabla 7. Representaciones de los estudiantes indicados para los problemas T04 y T05

T04		
<p>T04</p>	<p>5P2b</p>	<p>6P1b</p>
T05		
<p>T05</p>	<p>5P1b</p>	<p>6P2a</p>

Tabla 8. Representaciones de los estudiantes indicados para el problema T09

T09		
<p>T09</p>	<p>10P1a</p>	<p>10P1b</p>

ANEXO 2

Tabla 9. Planes elaborados por las parejas 5P1 y 5P2 en los problemas seleccionados para la fase experimental 3

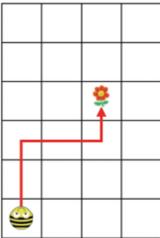
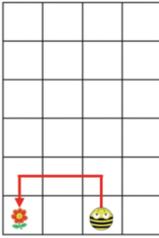
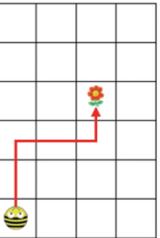
	5P1 – T06	5P1 – T08	5P2 – T06
Planes elaborados con las tarjetas			
1	[↑, ↑, >, ↑, ↑, >, ↑, GO]	[↑, <, ↑, ↑, ↑, ↓, GO]	[↑, ↑, >, <, ↑, GO]
2	[↑, ↑, >, ↑, ↑, >, <, GO]	[↑, <, ↑, ↑, GO]	[↑, ↑, >, ↑, ↑, GO]
3	[↑, ↑, >, ↑, ↑, <, ↑, GO]	[↑, <, ↑, ↑, ↓, GO]	X
4		[↑, <, ↑, ↑, ↑, >, ↑, GO]	

Tabla 10. Planes elaborados por la pareja 6P1 en los problemas seleccionados para la fase experimental 3

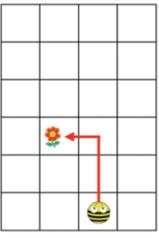
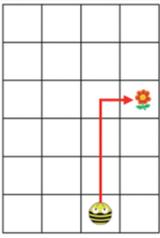
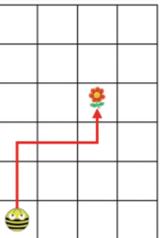
	6P1 – T03	6P1 – T05	6P1 – T06
Planes elaborados con las tarjetas			
1	[↑, ↑, <, GO]	[↑, ↑, ↑, >, GO]	[↑, ↑, >, >, ↑, GO]
2	[↑, ↑, >, >, GO]	[↑, ↑, ↑, >, ↑, GO]	[↑, ↑, ↑, >, ↑, ↑, GO]
3	[↑, ↑, >, ↑, GO]		[↑, ↑, ↑, >, ↑, ↑, ↑, GO]
4			[↑, ↑, ↑, >, ↑, <, ↑, GO]
5			[↑, ↑, ↑, >, ↑, <, ↑, ↑, ↑, GO]
6			[↑, ↑, ↑, >, ↑, ↑, GO]
7			[↑, ↑, ↑, >, ↑, ↑, <, ↑, GO]

Tabla 11. Planes elaborados por la pareja 6P2 en los problemas seleccionados para la fase experimental 3

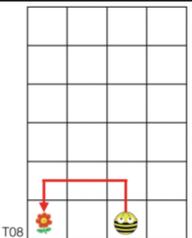
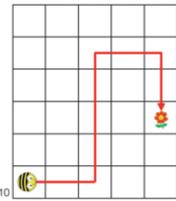
	6P2 – T08	6P2 – T10
Planes elaborados con las tarjetas		
1	[↑, >, <, ↑, GO]	[↑, ↑, <, ↑, ↑, ↑, <, ↑, ↑, >, ↑, ↑, GO]
2	[↑, <, ↑, ↑, <, ↑, GO]	[↑, ↑, <, ↑, ↑, ↑, >, ↑, ↑, >, ↑, ↑, GO]

Tabla 12. Planes elaborados por la pareja 6P3 en los problemas seleccionados para la fase experimental 3

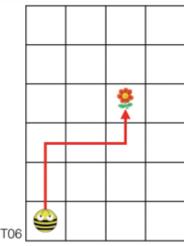
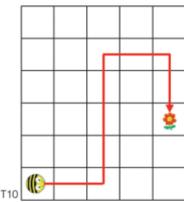
6P3 – T06		6P3 – T10	
			
Planes elaborados con las tarjetas			
1	[↑, ↑, >, ↑, >, ↑, GO]	[↑, ↑, >, ↑, GO]	
2	[↑, ↑, >, ↑, ↑, >, GO]	[↑, ↑, >, ↑, ↑, ↑, >, ↑, ↑, >, GO]	
3	[↑, ↑, >, ↑, ↑, ↑, GO]	[↑, ↑, >, ↑, ↑, ↑, >, ↑, ↑, >, ↑, GO]	
4	[↑, ↑, >, ↑, ↑, <, GO]	X	
5	[↑, ↑, >, ↑, ↑, <, ↑, GO]		

Tabla 13. Planes elaborados por la pareja 10P1 en los problemas seleccionados para la fase experimental 3

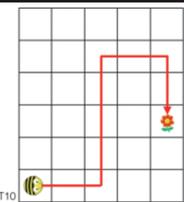
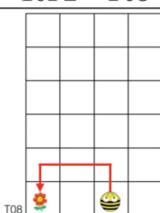
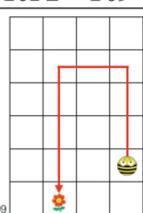
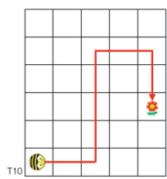
10P1 – T10	
	
Planes elaborados con las tarjetas	
1	[↑, ↑, >, ↑, ↑, ↑, ↑, >, ↑, ↑, >, ↑, GO]
2	[↑, ↑, <, ↑, ↑, ↑, ↑, >, ↑, ↑, >, ↑, GO]

Tabla 14. Planes elaborados por la pareja 10P2 en los problemas seleccionados para la fase experimental 3

	10P2 – T08	10P2 – T09	10P2 – T10
			
			
			
Planes elaborados con las tarjetas			
1	[↑, ↑, <, ↑, ↑, <, ↑, GO]	[↑, ↑, ↑, <, ↑, ↑, <, ↑, ↑, ↑, ↑, GO]	[↑, ↑, <, ↑, ↑, ↑, >, ↑, ↑, >, ↑, GO]