

Fundamentos de diseño de un entorno tecnológico para el estudio de las habilidades en resolución de problemas en primeras edades escolares

Fundamentals of the Design of a Technological Environment for the Study of the Problem-Solving Abilities in Early Childhood

Fonaments de disseny d'un entorn tecnològic per a l'estudi de les habilitats en resolució de problemes en primeres edats escolars

Pascual D. Diago^{1,*}, José A. González-Calero², David Arnau¹

1 | Departament de Didàctica de la Matemàtica, Universitat de València, Valencia, España

2 | Departamento de Matemáticas, Universidad de Castilla-La Mancha, España

*Autor para correspondencia: Pascual.Diago@uv.es (Pascual D. Diago)

Recibido: 30/04/2019 | Aceptado: 31/05/2019 | Publicado: 24/06/2019

Cómo citar: Diago, P. D., González-Calero, J. A., y Arnau, D. (2019). Fundamentos de diseño de un entorno tecnológico para el estudio de las habilidades en resolución de problemas en primeras edades escolares. *Research in Education and Learning Innovation Archives*, 22,58-76. [10.7203/realia.22.14113](https://doi.org/10.7203/realia.22.14113)

Copyright: El/La Autor/a.
Open Access: Este es un artículo de acceso abierto distribuido bajo los términos de la licencia Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

Financiación: Este trabajo ha contado con el apoyo de los proyectos UV-SFPIE_GER18-848319 (Universitat de València) y EDU2017-84377-R (MINECO/FEDER) y del programa propio de ayudas a investigación de la Universitat de València (Estancias de PDI, 2019).

RESUMEN: En este trabajo se presentan los fundamentos de diseño de un *software* capaz de monitorizar y registrar datos referentes a la gestión de procesos de resolución de problemas matemáticos. El *software* permite una recogida sistemática de las variables del proceso de resolución a partir de las cuales es posible obtener diferentes métricas de aprendizaje del estudiante. Al no centrar la atención exclusivamente en el resultado final del proceso de resolución, es posible identificar tendencias cognitivas de los estudiantes. En este manuscrito describimos las características de este entorno tecnológico junto con la base didáctica que justifica el interés de la investigación en niveles escolares en los que el escaso conocimiento limita la formalización de situaciones de resolución de problemas. Se adjuntan los primeros resultados de un estudio piloto que ha permitido evaluar fortalezas y debilidades de la versión en desarrollo. Finalmente, se discute sobre futuras direcciones para este entorno tecnológico como un paso más tanto en innovación como en investigación en Educación Matemática.

PALABRAS CLAVE: Entornos Tecnológicos; Resolución de problemas; Primeras edades escolares; Métricas de aprendizaje

ABSTRACT: In this paper we present the fundamentals of software designs capable of monitoring and recording data related to the management of mathematical problem solving processes. The software allows a systematic collection of the variables of the resolution process from which it is possible to obtain different student-learning analytics. By not focusing exclusively on the final outcome of the resolution process, it is possible to identify the cognitive tendencies of the students. In this manuscript, we describe the characteristics of this technological environment together with the didactic basis that justifies the interest in research at school levels in which poor knowledge limits the formalization of problem solving situations. The initial results of a pilot study that has allowed for the evaluation of the strengths and weaknesses of the developing version are attached. Finally, future directions for this technological environment are discussed as another step further

in both innovation and research in Mathematics Education.

KEYWORDS: Technological environments; Problem solving; Early school years; Learning metrics

RESUM: En aquest treball es presenten els fonaments de disseny d'un programari capaç de monitoritzar i registrar dades referents a la gestió de processos de resolució de problemes matemàtics. El programari permet una recollida sistemàtica de les variables del procés de resolució a partir de les quals és possible obtenir diferents mètriques d'aprenentatge de l'estudiant. Al no centrar l'atenció exclusivament en el resultat final del procés de resolució, és possible identificar tendències cognitives dels estudiants. En aquest manuscrit descrivim les característiques d'aquest entorn tecnològic juntament amb la base didàctica que justifica l'interès de la recerca en nivells escolars en els quals l'escàs coneixement limita la formalització de situacions de resolució de problemes. S'adjunten els primers resultats d'un estudi pilot que ha permès avaluar fortaleses i debilitats de la versió en desenvolupament. Finalment, es discuteix sobre futures adreces per a aquest entorn tecnològic com un pas més tant en innovació com en recerca en Educació Matemàtica.

PARAULES CLAU: Entorns Tecnològics; Resolució de problemes; Primeres edats escolars; Mètriques d'aprenentatge

Notas de aplicación práctica

Qué se sabe sobre este tema

- El desarrollo de sistemas tecnológicos orientados a la enseñanza de la resolución de problemas es, desde la década de los 80, un tema de interés del área de la Educación Matemática. Sin embargo, por diversas razones existe una escasez notable de entornos específicos para la resolución de problemas en la etapa de Educación Infantil.

Qué aporta este trabajo

- Este manuscrito describe los fundamentos de diseño de un entorno tecnológico orientado a la resolución de problemas en estudiantes de niveles escolares iniciales, en los que el contenido matemático puesto en juego es apropiado para estas edades y no está ligado a habilidades de lecto-escritura.

Implicaciones sobre la práctica y / o política

- Se presenta un conjunto de herramientas orientadas a la obtención de métricas de aprendizaje basado en el conocimiento específico del dominio, como punto de partida para la implementación de un entorno tecnológico de resolución de problemas matemáticos para primeros niveles escolares.

1. INTRODUCCIÓN

1.1. Métricas de aprendizaje en contextos educativos

Entre las tendencias reseñadas en el horizonte a cinco años presentado por [Freeman, Adams-Becker, Cummins, Davis, y Hall-Giesinger \(2017\)](#) se incluye, como reto a adoptar a medio plazo en la Educación Infantil y Primaria, el aumento de la atención a las métricas de aprendizaje. Como allí se indica, esto será posible, por un lado, gracias al incremento de la presencia de la tecnología y sus aplicaciones en contextos de enseñanza y aprendizaje; y por otro, también gracias al exponencial desarrollo tecnológico que se está llevando a cabo. Así, de la misma manera que hoy estamos expuestos al análisis de datos de consumo en tiempo real mientras hacemos uso de las redes sociales, compramos por internet o visionamos nuestras series favoritas online, las próximas generaciones de estudiantes estarán también expuestas al análisis de su aprendizaje, facilitando a profesores y escuelas informes y diagnósticos en tiempo real sobre su aprendizaje y nivel académico.

Debido a la relativa juventud del término *learning analytics* (métricas de aprendizaje) en contextos educativos, en la actualidad no encontramos una única definición para

el mismo. Dependiendo del propósito educativo que se persiga, diferentes dimensiones derivadas del análisis de estas métricas de aprendizaje pueden utilizarse para confeccionar una definición que responda a objetivos concretos (Viberg, Hatakka, Bälter, y Mavroudi, 2018). Por ejemplo, Mah (2016) centra la atención del análisis de métricas de aprendizaje en la obtención de “información dinámica sobre los estudiantes y sus entornos de aprendizaje con el fin de acceder, obtener y analizar dichos entornos para modelizar, predecir y optimizar el proceso de aprendizaje” (p. 288). Para Caballé y Clarisó (2016) el centro de interés es monitorizar la actividad y progreso de los estudiantes con el objetivo de identificar factores críticos y definir acciones de mejora por parte del profesorado. El informe del U.S. Department of Education, Office of Educational Technology (Bienkowski, Feng, y Means, 2012) establece que las métricas de aprendizaje en escenarios educativos permiten crear aplicaciones que influyan en la práctica educativa. En esta línea, otros trabajos definen las métricas de aprendizaje en educación en términos de ayudas reales al profesorado. En concreto, ayudas dirigidas a proporcionar información sobre las características del proceso de aprendizaje del estudiante, favorecerlo y cambiar su entorno de aprendizaje como consecuencia (Drachsler y Kalz, 2016). Por último, se ha de destacar que el papel de las métricas de aprendizaje en educación puede ser crucial para una mejor comprensión de las necesidades educativas especiales; en particular para detectar y articular estrategias de intervención útiles en estudiantes de altas capacidades o de poblaciones en riesgo de exclusión (Freeman et al., 2017).

1.2. Nuevas perspectivas en Educación Matemática

En la conferencia inaugural del ICMI Study 17 de 2006, titulada *30 years of digital Technologies in Mathematics Education and the Future*, Seymour Papert argumentó cómo el fácil acceso a los ordenadores ofrecía la oportunidad de dotar a la Educación Matemática del momento de un nuevo enfoque (Hoyles y Lagrange, 2010). Además, el profesor Papert planteó también que, si bien era importante considerar cómo el conocimiento existente podría ser abordado de forma mejorada mediante el uso de la tecnología, deberíamos reservar al menos el 10% de nuestro tiempo y energía para considerar qué nuevos tipos de conocimiento y prácticas matemáticas podrían surgir como resultado de lograr un uso efectivo de las tecnologías digitales (un tema que previamente ya había explorado en Papert (1996). Así, además del elemento motivacional que ofrecen los entornos tecnológicos (algo ya puesto de manifiesto en Papert, 1980), los entornos computacionales basados en programación ofrecerían nuevos objetivos pedagógicos que contribuirían al desarrollo del pensamiento matemático en los niños (Beatty y Geiger, 2010). A partir de aquel momento se comenzó a pensar no solo en el diseño de actividades digitales que favorecieran el razonamiento matemático, sino en el diseño de los propios entornos con el objetivo de que fueran capaces de ofrecer un contexto en el que el razonamiento matemático ocupara un papel central (Fuglestad, Healy, Kynigos, y Monaghan, 2010). Así, floreció un conjunto de nuevas herramientas digitales orientadas a que el estudiante desarrollara habilidades propias del razonamiento geométrico, la visualización, el razonamiento estadístico y probabilístico, el pensamiento numérico, etc. (Hoyles y Lagrange, 2010). Como justificaremos más adelante, el caso del desarrollo de las habilidades propias de la resolución de problemas no fue distinto y nuevos entornos tecnológicos han surgido con este firme propósito.

Este cambio de paradigma nos invita a repensar los contenidos y procedimientos matemáticos que los estudiantes son capaces de aprender cuando hacen uso de herramientas tecnológicas. Así, si somos capaces de integrar el conocimiento didáctico específico, tendremos la oportunidad de diseñar entornos capaces de generar información sobre lo que el estudiante está haciendo en cada momento, obteniendo medidas del aprendizaje a lo largo de todo el proceso, y no solo en el momento de plantear un resultado final.

1.3. Entornos tecnológicos orientados a la resolución de problemas

El diseño de entornos tecnológicos orientados a la enseñanza y aprendizaje de la resolución de problemas verbales ha sido un tema recurrente en la investigación del área de la Didáctica de las Matemáticas, especialmente centrado en la aritmética y el álgebra (Rakes, Valentine, McGatha, y Ronau, 2010). Un análisis de estos entornos tecnológicos orientados a la resolución de problemas nos permite observar que éstos han tomado el rol del profesor principalmente en dos vías: i) tutorizando el proceso de resolución; y ii) generando secuencias de problemas adaptados a las características del resolutor. En esta categoría se incluyen los llamados Sistemas Tutoriales Inteligentes (ITS, del inglés *Intelligent Tutoring Systems*), que son capaces tanto de supervisar las vías de resolución tomadas, como de ofrecer ayudas a demanda durante el proceso de resolución. Algunos ejemplos de entornos tecnológicos orientados a la resolución de problemas son: AnimalWatch (Beal, 2013); ANIMATE (Nathan, 1998); HERON (Reusser, 1993); HINTS (Arnau, Arevalillo-Herráez, Puig, y González-Calero, 2013); MathCAL (Chang, Sung, y Lin, 2006); o Ms. Lindquist (Heffernan y Koedinger, 2000). Además, existen algunos entornos específicamente orientados a la resolución de problemas mediante programación visual (Chao, 2016).

Aunque el desarrollo de estas herramientas ha sufrido sus altibajos en los últimos años (Drijvers et al., 2010; McArthur y Lewis, 1998), queremos poner el foco de atención en aquellos entornos orientados a la resolución de problemas que, tomando en consideración “el 10% de Papert”, permitan desarrollar nuevos tipos de conocimiento y nuevas prácticas relacionadas con el desarrollo de las habilidades en resolución de problemas. Especialmente, ofreciendo información sobre el aprendizaje del estudiante a partir de información obtenida durante el proceso de resolución y no solo a partir del resultado final.

1.4. Entornos tecnológicos basados en programación en bloques

En los trabajos seminales sobre construccionismo (Papert, 1980, 1991) Seymour Papert plantea que el diseño de programas informáticos pueden facilitar el aprendizaje de conceptos y procedimientos matemáticos. En el pasado, esta idea fundamentó el origen del lenguaje de programación *Logo* (Clements, 1986), y en el presente del lenguaje de programación en bloques *Scratch*, en el que los estudiantes pueden arrastrar y enlazar instrucciones ya programadas, como piezas de un puzzle, para crear programas complejos, juegos o simulaciones interactivas. Así, los entornos de programación por bloques ofrecen transparencia y facilidad de uso para poder ser utilizados directamente por los estudiantes sin requerir conocimientos previos en programación (Weintrop y Wilensky, 2015).

Por otra parte, los lenguajes de programación basados en bloques ofrecen entornos privilegiados para la aparición de habilidades de resolución de problemas mediante el uso de estrategias de descomposición, diseño de algoritmos, abstracción y razonamiento lógico. Así, las habilidades y conocimientos matemáticos puestos en juego en estos entornos de programación en bloques están estrechamente relacionados con la llamada *heurística*, como campo de estudio que hace referencia a la capacidad para generar soluciones originales en la resolución de un problema. En palabras de Pólya (1945, p. 102), “la heurística moderna trata de comprender el método que conduce a la solución de problemas, en particular las operaciones mentales típicamente útiles en este proceso”.

La llegada de la robótica a las aulas, como una extensión tangible de los entornos tecnológicos basados en programación en bloques en la que el robot actúa como herramienta activa intermediaria entre estudiante, profesor y materia escolar (Gaudiello y Zibetti, 2016), ha seguido favoreciendo los procesos de razonamiento matemático y las habilidades en resolución de problemas permitiendo, además, la integración de aprendizajes de áreas como las ciencias, las matemáticas, la tecnología y la ingeniería (Bers,

Seddighin, y Sullivan, 2013).

2. PROPÓSITO

En este trabajo se describen los fundamentos de diseño de un entorno tecnológico interactivo orientado a la enseñanza y aprendizaje de la resolución de problemas en niveles escolares elementales (Educación Infantil y primeros cursos de Educación Primaria). Este entorno, basado en programación en bloques, persigue los siguientes objetivos:

1. Proporcionar un entorno tecnológico capaz de proponer tareas matemáticas que resulten verdaderos problemas adaptados a primeros niveles escolares. La situación problemática se basará en contenido matemático elemental.
2. Diseñar un *software* orientado a la investigación en resolución de problemas basado en métricas de aprendizaje que ofrezca información sobre aspectos relativos al propio proceso de resolución de la tarea y no solo sobre el resultado final. Así, los datos recogidos por el *software* no solo facilitarán métricas finales de éxito/fracaso, sino que ayudarán a comprender procesos cognitivos relacionados con la monitorización y gestión de los diferentes planes de resolución ideados por el estudiante durante las distintas fases del proceso de resolución.
3. Poner a prueba, en un contexto real, el entorno tecnológico con el fin de explorar si permite observar estrategias y dificultades experimentadas por los estudiantes a la hora de enfrentarse a tareas propias de resolución de problemas.

3. FUNDAMENTOS DE DISEÑO

3.1. La resolución de problemas en primeras edades escolares

El trabajo de Kilpatrick (1985), *A Retrospective Account of the Past Twenty-five Years of Research on Teaching Mathematical Problem Solving*, ponía de manifiesto la naturaleza subjetiva de los problemas de matemáticas y cómo esta había sido aceptada hacía ya 40 años atrás por la comunidad de investigación en Educación Matemática. En palabras de Brownell (1942), una tarea puede ser considerada como problema cuando el resolutor tiene el suficiente conocimiento base como para poder entenderla pero es incapaz de afrontarla con técnicas o métodos directos. Además, la resolución de problemas de matemáticas obliga al resolutor a hacer uso de conocimiento matemático en el planteamiento de la solución. Así, esta definición hace que los problemas de matemáticas estén categorizados en un continuo entre enigmas (los cuales dejan al resolutor perplejo ante la tarea), ejercicios rutinarios (el resolutor conoce sobradamente los mecanismos para resolver la tarea) y verdaderos problemas (el resolutor entiende la tarea pero no tiene un mecanismo directo para dar con la solución de la misma).

En lo concerniente a la Educación Infantil y los primeros cursos de Educación Primaria, la introducción de la resolución de problemas supone en muchas ocasiones enfrentarse a un dilema didáctico. Este dilema es consecuencia del carácter psicológico ya expuesto sobre el concepto de problema y del limitado o escaso contenido matemático que se desarrolla en estos primeros niveles educativos. Como resultado, en muchas ocasiones los maestros y maestras tienen dificultades para plantear tareas matemáticas que resulten verdaderos problemas para sus estudiantes; es decir, el cuidado por no desanimar a sus estudiantes, les conduce raramente a plantear problemas y a abusar de puros ejercicios matemáticos. A pesar de ello, las directrices y estándares de aprendizaje incluyen la resolución de problemas como un contenido central que debe ser enseñado y aprendido, especialmente en los primeros niveles escolares (NCTM, 2000). De hecho, la resolución de problemas como actividad mental, se considera

un estándar de proceso que la investigación educativa ha relacionado con procesos cognitivos cruciales para el aprendizaje y comprensión de las estructuras y relaciones matemáticas (Weber y Leikin, 2016).

Por otra parte, las recomendaciones que se citan en *Principles and Standards for School Mathematics* (NCTM, 2000) insisten en la necesidad de que los estudiantes empleen y expliciten mecanismos de control y monitorización durante el proceso de resolución de problemas. Estas habilidades metacognitivas están incluidas entre las cinco categorías identificadas por Schoenfeld (1992) para la investigación en la enseñanza y aprendizaje de la resolución de problemas: el conocimiento base; las estrategias de resolución de problemas; monitorización y control; creencias y afectos; y prácticas. De este modo, la consciente monitorización y control ejecutivo, por parte de los estudiantes, sobre las estrategias de resolución empleadas durante la resolución de la tarea serían consideradas como operaciones cognitivas esenciales que se emplean para tomar decisiones sobre las posibles vías de resolución con el fin de tener éxito en la resolución de un problema.

3.2. La programación en bloques para evaluar las habilidades en resolución de problemas

El diseño de nuestro entorno tecnológico estará basado en la programación en bloques, ya que permite iniciar el aprendizaje de la resolución de problemas desde un enfoque tecnológico (Bers, Flannery, Kazakoff, y Sullivan, 2014; Chao, 2016; Grover, Pea, y Cooper, 2015; Shute, Sun, y Asbell-Clarke, 2017). El potencial de estas propuestas se apoya en que exige al estudiante reflexionar sobre algunos de los pasos de resolución de problemas tal y como los propone Pólya (1945); pero con el apoyo del entorno tecnológico en otros de los pasos. Así, el estudiante debe realizar un plan previo a la programación, con la exigencia adicional de usar una representación formal para el mismo (un lenguaje visual de programación por bloques); pero el entorno le ofrece la posibilidad de realizar una valoración del plan ideado a partir de la respuesta proporcionada.

Dado el potencial visual de la programación por bloques (que en edades tempranas permite prescindir de instrucciones escritas para el usuario) los entornos que hacen uso de ellos permiten plantear tareas que pueden ser entendidas como verdaderos problemas con contenido matemático apto para los niveles escolares iniciales. Algunos entornos tecnológicos existentes basados en programación en bloques serían *ScratchJr* o *Code.org* (en el apartado de *software*) y *Bee-bot*, *Cubetto* u *Ozobot* (en el apartado de robots), por ejemplo.

3.3. El interfaz gráfico del usuario y su funcionamiento

Los objetivos didácticos en base a los que se diseña este *software* son los que siguen:

- El contenido matemático puesto en juego está basado en la propia resolución de problemas y en el desarrollo de soluciones heurísticas.
- El entorno transmite la información al usuario en lenguaje natural, sin hacer uso de texto, con el fin de que pueda ser utilizado por estudiantes de primeros niveles escolares, con escasa o nula capacidad lectora.
- Los bloques de programación están constituidos, básicamente, por los movimientos que el objeto (abeja en nuestro caso) es capaz de realizar para recorrer un tablero.
- El entorno debe facilitar la aparición de elementos propios de la resolución de problemas ligados a las fases de Pólya (1945, previamente descritas).
- El sistema monitoriza la vía de resolución tomada por el usuario y es capaz de evaluarla en tiempo real. En el futuro se espera que el entorno sea capaz de dar

feedback inmediato al usuario y ofrecer ayudas a demanda en función de las decisiones tomadas por el estudiante.

La Figura 1 muestra el interfaz de usuario, con tres zonas diferenciadas:

1. El enunciado del problema (parte superior izquierda de la Figura 1).
2. Los bloques de programación elegibles correspondientes a los movimientos que la abeja es capaz de realizar (parte superior derecha de la Figura 1). Es importante hacer notar que los bloques de movimiento corresponden al propio sistema de referencia del objeto. Además, los giros (a derecha o izquierda) corresponden a giros de 90° sobre sí mismo (en sentido horario o anti-horario), sin que el objeto abeja se traslade. Se incluyen, además, bloques para borrar una instrucción del plan ideado o el plan completo. La abeja solo ejecutará un plan programado cuando el usuario seleccione la instrucción GO. El conjunto de bloques mostrados es modificable, pudiendo ofrecer más o menos instrucciones al usuario.
3. La caja de secuenciación (parte inferior de la Figura 1). En esta zona el usuario puede visualizar los bloques que está secuenciando y que, posteriormente, ejecutará la abeja cuando reciba la instrucción GO. Esta zona está dividida en tres espacios, correspondientes al número máximo de intentos que se le permite al usuario introducir para resolver el problema (fijado en tres actualmente). Cuando un plan es ejecutado con resultado no exitoso, éste queda visible y bloqueado para que el usuario pueda consultarlo y configurar un nuevo plan en el siguiente intento pudiendo consultar las instrucciones que anteriormente ha introducido (si no ha agotado el número de intentos estipulados).

La programación y el pilotaje inicial del entorno se realizó utilizando el software libre *Snap!*, una reimplementación de *Scratch* orientada a la introducción de la computación

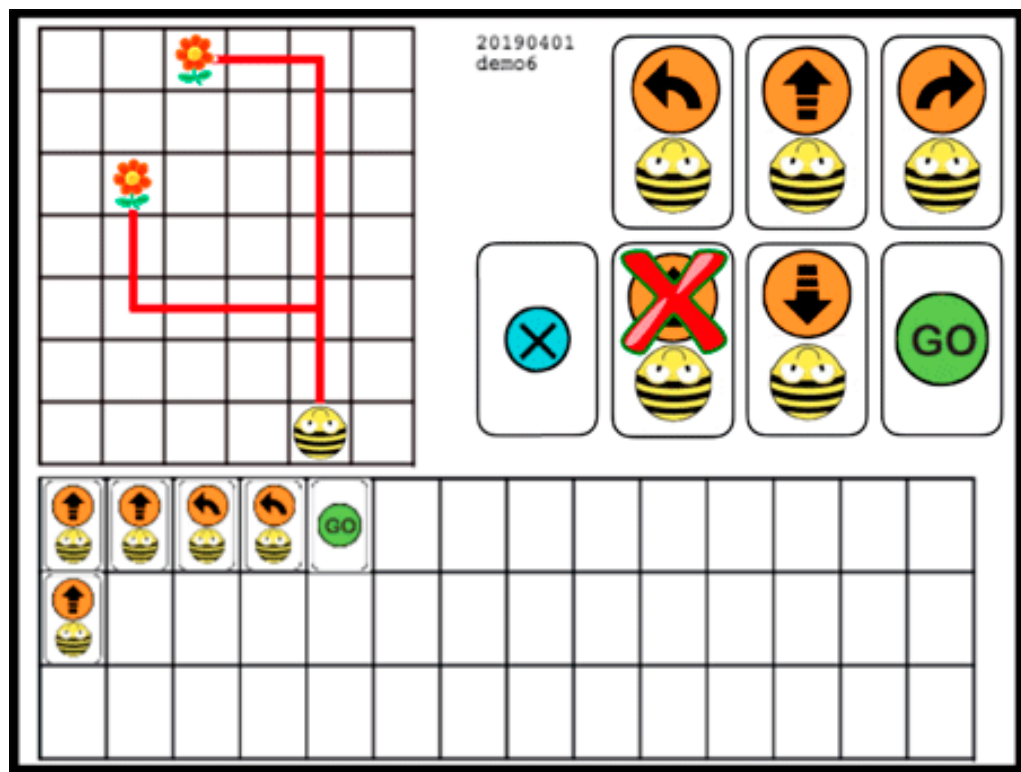


Figura 1. Vista general del interfaz de usuario

en niveles de secundaria, que permite la creación de subrutinas con instrucciones de niveles de programación más altos como *Javascript*. La elección de *Snap!* responde a la intención de acercar un lenguaje de programación por bloques sencillo y asequible a colectivos no familiarizados con este tipo de entornos, más comunes en otros ámbitos propios de las ciencias de la computación. Los problemas propuestos a los estudiantes pueden ser fácilmente añadidos mediante un archivo de configuración de programa que se carga con anterioridad y se describen en las secciones 3.3.1. y 3.3.2. El registro de datos se hace por medio de un archivo de texto en el que el sistema va almacenando información cada vez que el usuario selecciona una instrucción del panel de bloques. En la sección 3.3.3. se dan más detalles sobre la estructura de almacenamiento de datos.

3.3.1. La definición de los problemas

Con el fin de que el sistema pueda proporcionar datos sobre la gestión del proceso de resolución, el entorno ofrece al usuario un problema verbal enunciado de forma gráfica para el que se presentan explícitamente distintas vías de resolución. En el momento actual los problemas presentados consisten en la tarea de mover una abeja (basada en el diseño del robot *Bee-bot*) por una cuadrícula (fijada) desde una posición inicial hasta una de las flores mostradas siguiendo un camino marcado en rojo (como se aprecia en la Figura 1, izquierda). Con el fin de monitorizar y evaluar el proceso de resolución y las características de la toma de decisiones de los usuarios se proporcionan dos posibles trayectos que dan solución al problema. Para minimizar las instrucciones dadas al resolutor, dado que son estudiantes de primeros niveles escolares, los problemas presentados se enuncian todos de forma verbal al inicio de la sesión con la instrucción: *La abeja ha de llegar a una de las flores que aparecen en el tablero. Tú decides a cuál de ellas llevarla, siempre siguiendo los caminos marcados en rojo. Recuerda que tanto a la abeja como a nosotros nos da igual a qué flor elijas ir.* La solución al problema (también llamada plan en adelante) debe darse en el lenguaje que la abeja entiende, y que corresponde a bloques de instrucciones que aparecen en el propio interfaz gráfico.

3.3.2. Definición de las variables de tarea: los problemas administrados

Siguiendo las indicaciones de Kilpatrick (1978) para los estudios sobre resolución de problemas, se definen las siguientes variables asociadas a los problemas anteriormente descritos:

- El número de bloques (instrucciones) necesarios para resolver el trayecto dado (*n*).
- El número de giros involucrados necesarios para resolver el trayecto dado (*gir*).
- El cambio de orientación máximo (*ori*) de la abeja con respecto al sistema de referencia del usuario (que es fijo durante toda la tarea). Se tomará en cuenta siempre el mayor cambio de orientación realizado por la abeja durante todo el recorrido, en grados en sentido horario o antihorario.

Así por ejemplo, los problemas mostrados en la Figura 2 vendrían definidos por las variables mostradas en la Tabla 1 (en lo que sigue se considerará como trayecto 0 el menos complejo y 1 el más complejo).

Tabla 1. Variables de complejidad asociadas a los problemas de la Figura 2

Trayecto	Bloques	Giros	Ori	Bloques	Giros	Ori	Bloques	Giros	Ori	Bloques	Giros	Ori
0	5	0	0°	6	0	0	9	1	90°	11	1	-90°
1	4	1	90°	5	2	90°	9	2	90°	11	2	-90°

Estas variables de tarea son independientes y pueden controlarse previamente a la administración de los problemas. La definición de estas variables de tarea, además, per-

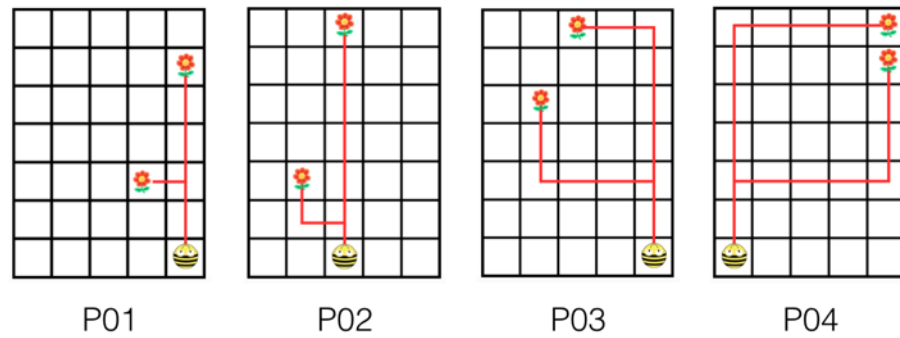


Figura 2. Ejemplos de problemas administrados durante el pilotaje del entorno

mite asignar una medida de complejidad para cada uno de los trayectos que conformen los problemas diseñados, pudiendo graduar la aparición de los mismos al usuario por su complejidad estructural.

3.3.3. Las variables de proceso y de producto: la estructura de recogida de datos

Con el fin de que los datos recogidos den cuenta de las habilidades puestas en juego durante el proceso de resolución se definen las siguientes variables de producto:

- El número de intentos iniciados por el usuario para resolver el problema (i). En la versión actual el número máximo de intentos está fijado en tres.
- La complejidad del trayecto elegido por el usuario ($elec$). Dado que actualmente la herramienta solo permite la resolución por las vías marcadas, esta variable toma valores 0, en el caso de que el usuario tome el trayecto menos complejo y 1 en el caso de que elija el más complejo.
- Puntuación del problema ($score_p$). Cada problema se puntúa con un valor entre 0 y 1, que va a venir dado por la tasa de acierto máxima obtenida en cada una de los intentos permitidos (i), es decir $score_p = \max(t_i)$, donde la tasa t_i es la que viene definida más abajo y el número de intentos se fija en $i = 3$, como máximo.

El sistema, para cada uno de los intentos, genera una línea de información en un archivo de texto plano cada vez que el usuario selecciona uno de los bloques de movimiento disponible en el interfaz (Figura 1, parte superior derecha). Esta información hace referencia a datos relacionados con las siguientes variables de proceso:

- Bloque introducido (B). La instrucción concreta que se ha seleccionado del panel.
- Tasa de acierto hasta el primer error en el plan elaborado por el estudiante ($score_{t_i}$, en relación al trayecto seleccionado y al intento i). Esta definición, que no es la propia del análisis de procesos de resolución de problemas, se considera así por las características del propio entorno tecnológico; pues el movimiento de la abeja se vuelve difícilmente interpretable o bien carente de sentido una vez ejecuta el primer bloque no coincidente con el trayecto correcto seleccionado. Para cada intento i se calcula como $score_{t_i} = B_c / n$, donde B_c es el número de bloques correctos presentes en el plan (no se contemplan los borrados o modificados) hasta el primer error y n es el número total de bloques del trayecto elegido (que es calculado para cada problema, como se puede ver en la Tabla 1). Con esta definición, la tasa de acierto para cada intento i toma valores $0 < score_{t_i} < 1$, siendo $t_i = 1$ el indicador de que el plan elaborado contiene todas las instrucciones necesarias para recorrer exitosamente el trayecto seleccionado.

- Número de bloque (b). El sistema guarda de forma correlativa el número de instrucción seleccionada, en referencia al histórico de instrucciones secuenciadas para cada problema. Las instrucciones que posteriormente son borradas también son almacenadas para poder analizar los planes que se han modificado o borrado.
- Tiempo de respuesta ($time$). Hace referencia al tiempo de respuesta entre cada una de las instrucciones que el usuario introduce en el sistema.
- Posición del objeto abeja en cada momento (x, y, O). Dado que el objeto se desplaza por una cuadrícula, a la vez que se van introduciendo bloques de instrucciones, el software almacena las coordenadas x e y , relativas al tablero, y la orientación (O) del objeto, pudiendo calcular de forma inmediata si las instrucciones resuelven o no la situación planteada y pudiendo, en un futuro facilitar *feedback* al momento.

Con todo, por cada una de las instrucciones introducidas por el usuario en el panel de bloques (Figura 1, derecha), el sistema almacena una fila en un archivo de diario con la siguiente información:

[id_usuario ; id_problema ; intento (i) ; *elec*; núm_bloque (b) ; x ; y ; O ;
 B ; *time*]

Con esta información almacenada, para cada intento i , puede determinarse la variables de proceso correspondiente a la tasa de aciertos (t_i) y la variable de producto correspondiente a la puntuación o tasa de éxito del problema ($score_p$).

4. EXPERIMENTO PILOTO

Con el fin de responder al tercer objetivo de este trabajo, poder testear el funcionamiento del entorno tecnológico, y continuar el trabajo iniciado en (Diago, Arnau, y González-Calero, 2018c), se diseñó un experimento piloto que pudiera poner de manifiesto las debilidades y fortalezas de la versión del software en desarrollo.

4.1. Participantes

El experimento se llevó a cabo con un total de 45 estudiantes, 23 (52,1% mujeres) correspondientes al último curso de Educación Infantil (5 años) y 22 (50% mujeres) correspondientes a primer curso de Educación Primaria (6 años).

4.2. Material y métodos

Haciendo uso de seis ordenadores con pantalla táctil se administró una colección de problemas en tandas a cada una de las muestras participantes (Figura 3). Los estudiantes, en grupos de seis, eran instruidos en el uso del entorno y posteriormente resolvían, de forma autónoma, las cuatro tareas de activación mostradas en la Figura 4.

Una vez superada esta fase, el entorno administraba uno detrás de otro los cuatro problemas mostrados en la Figura 1, que permiten dos posibles vías de resolución y cuyos parámetros estructurales de complejidad están descritos en la Tabla 1. El número máximo de intentos durante todo el experimento estaba fijado en tres. En total, cada tanda de seis estudiantes empleaba unos 20 minutos en completar el experimento.

4.3. Resultados iniciales y discusión

A continuación, describimos algunos resultados preliminares derivados del análisis de datos obtenidos en el experimento piloto. En todos ellos, se llevó a cabo un test Shapiro-Wilk para comprobar la normalidad de los datos analizados. Los resultados de estos tests no permitieron asegurar la normalidad de los datos, por los que se aplicaron tests no paramétricos con corrección de continuidad.



Figura 3. Estudiante haciendo uso del entorno tecnológico

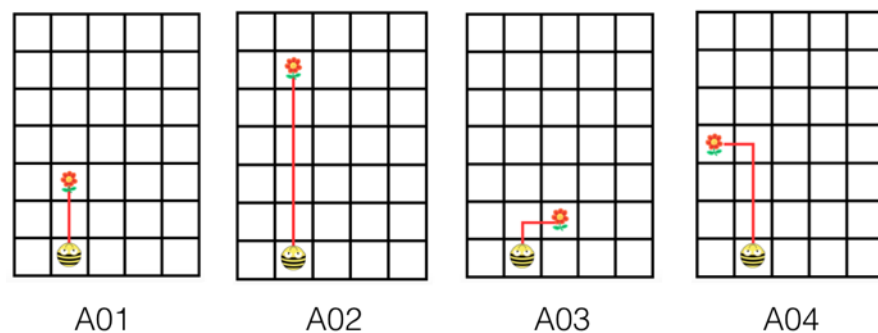


Figura 4. Tareas de activación administradas a los participantes en el experimento piloto

4.3.1. Sobre la influencia de los giros y el número de bloques a secuenciar

Las tareas de activación se han tomado como instrumento para medir la influencia de la aparición de instrucciones de giro en el éxito de la resolución de estas tareas. Así, sobre el total de estudiantes, la puntuación ($score_p$) obtenida en las tareas de activación con giro (A03 y A04, $Mdn = 0.67$) era significativamente menor que en las tareas de activación sin giro (A01 y A02, $Mdn = 1.00$), con $V = 92.5$, $p < .001$ y un tamaño del efecto medio ($r = 0.40$). Analizando las muestras separadas por edades (ver Figura 5), como era de esperar, las diferencias en el éxito entre las tareas con y sin giro muestran un mayor tamaño de efecto sobre el grupo de Educación Infantil ($p < .001$, $r = 0.61$) que sobre el de 1º de Educación Primaria ($p < .001$, $r = 0.52$). La explicación más plausible es que la gestión de los giros requiere del estudiante un manejo correcto de los cambios de sistema de referencia (abeja-estudiante) y de ciertas habilidades relacionadas con el pensamiento espacial, la rotación mental o la visualización que todavía no se encuentran totalmente desarrolladas en las edades en las que se ha llevado a cabo el experimento, siendo la población de 5 años la que muestra más carencias en este tipo de tareas espaciales. Tareas similares llevadas a cabo en contextos manipulativos (con el robot *Bee-bot*) han puesto de manifiesto las dificultades experimentadas por estudiantes de estos niveles educativos iniciales al enfrentarse a tareas que exigen abandonar la perspectiva egocéntrica típica en esta etapa (Diago, Arnau, y González-Calero, 2018a, 2018b; Hitt, Saboya, y Cortés, 2017)

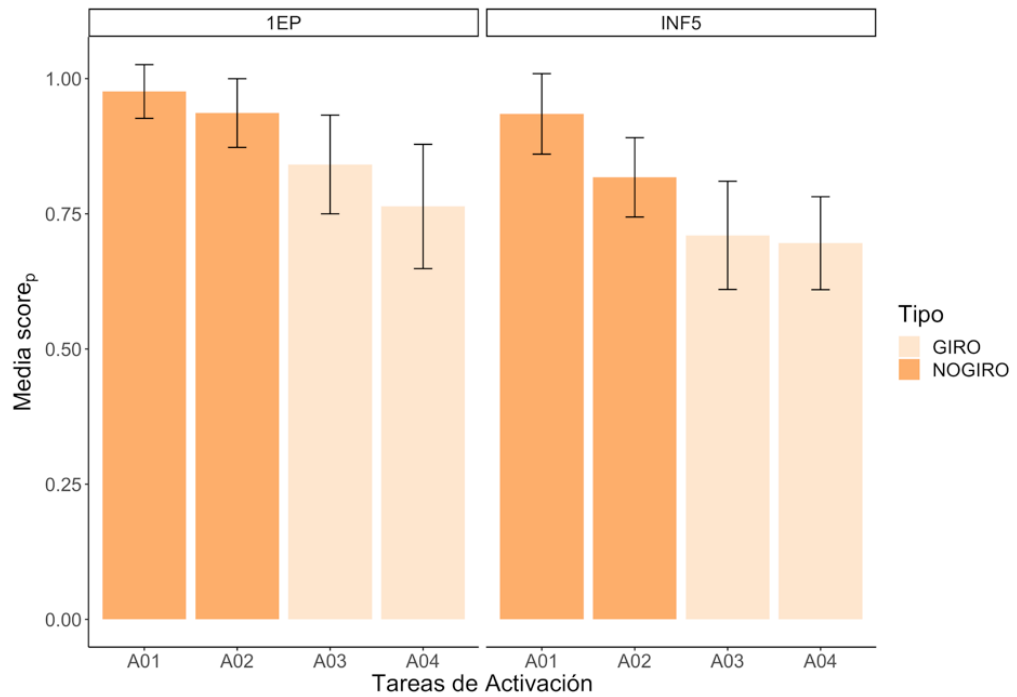


Figura 5. Media de la tasa de éxito ($score_p$) para las tareas de activación

También es destacable cómo el número de bloques a secuenciar en el plan de resolución afecta al éxito en la tarea. Así, el éxito en las tareas de activación que involucran menos pasos (A01 y A03, $Mdn = 1.00$) es significativamente menor al éxito de aquellas que involucran más pasos (A02 y A04, $Mdn = 0.80$) con $V = 1018$, $p = .038$, $r = 0.16$. Esta diferencia significativa tiene un mayor tamaño de efecto cuando se comparan el éxito entre las tareas que no involucran giro, A01 ($Mdn = 1.00$) y A02 ($Mdn = 1.00$) con $V = 182$, $p = .019$, $r = 0.25$. Este hecho es remarcable ya que el número de pasos a secuenciar involucra, como mucho, conjuntos de cinco instrucciones; cantidades sobradamente conocidas en los niveles en los que se llevó a cabo el experimento.

4.3.2. Sobre el papel del género en las tasas éxito de los problemas

Trabajos similares han obtenido un mejor desempeño en hombres con respecto a mujeres en actividades basadas en robótica y programación a través de tareas relacionadas con habilidades espaciales y de resolución de problemas (Coxon, 2012; González-Calero, Cózar, Villena, y Merino, 2018; Julià y Antolí, 2018). En nuestro caso, la puntuación media por intento ($score_{t_i}$) presentó diferencias a favor de los hombres ($Mdn = 0.67$) con respecto de las mujeres ($Mdn = 0.60$) en tres de los cuatro problemas administrados (P01 a P04, ver Figura 6); no obstante, estas diferencias resultaron ser no significativas ($W = 19124$, $p = .836$, $r = 0.02$). Tampoco fueron significativas las diferencias encontradas en las tareas de activación.

4.3.3. Sobre las elecciones en los trayectos

Trabajos como el de Diago et al. (2018c) sugieren una menor capacidad en estudiantes de 5 años con respecto a sus compañeros de 6 años a la hora de evaluar la complejidad con anterioridad a la elaboración de un plan de resolución en este tipo de tareas. En nuestro estudio piloto las elecciones de trayecto ($elec$) de los problemas P01 a P04 no se vieron significativamente afectadas por la edad de los estudiantes, $W = 4365$, $p = .169$, $r = 0.10$. En la Figura 7 podemos observar la distribución de los intentos iniciados por ambos grupos por cada uno de los trayectos, donde se aprecian intentos por trayectos

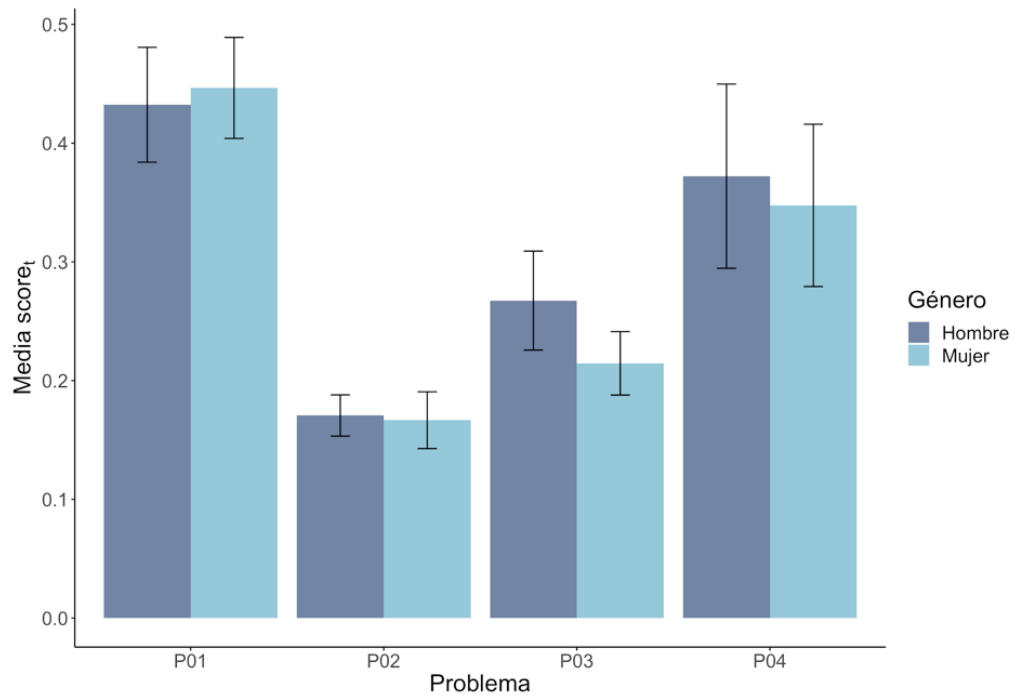


Figura 6. Tasa media de acierto por intento ($score_t$) por género para los problemas

complejos o sencillos indistintamente. En total, el número de intentos iniciados en el camino menos complejo (trayecto 0) fue de 304 contra los 85 que se iniciaron en el camino complejo (trayecto 1). Es especialmente destacable el caso de los estudiantes de infantil cuando abordan el último problema, pues todos toman la vía menos compleja.

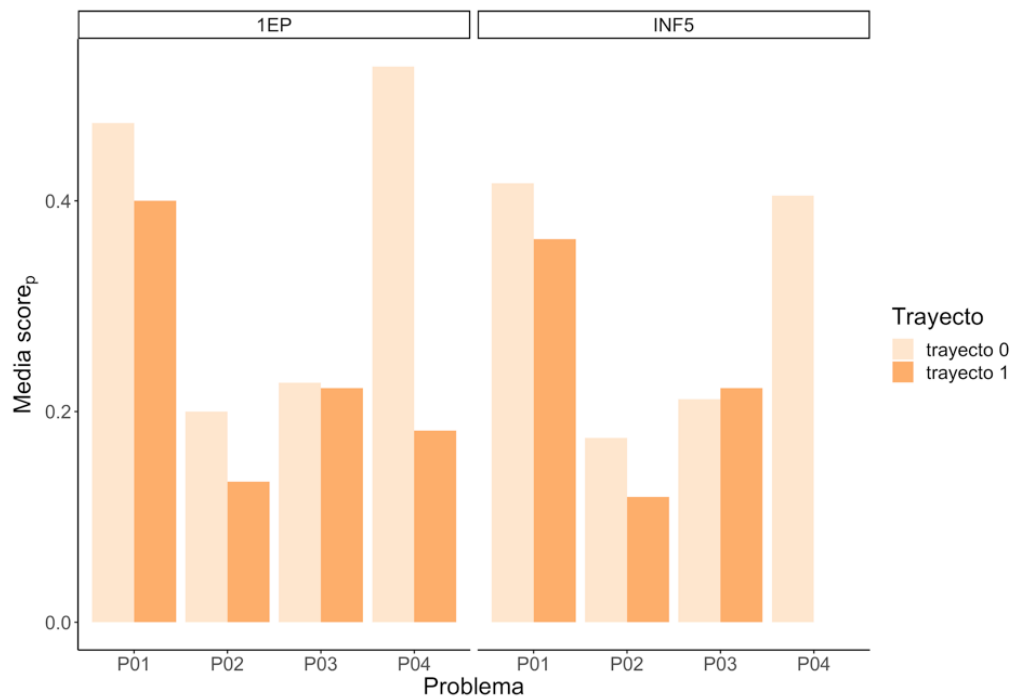


Figura 7. Media del éxito en la resolución del problema por problema ($score_p$) en función del grupo y trayecto elegido

La influencia de la elección del camino complejo (trayecto 1, $Mdn = 0.33$) o menos complejo (trayecto 0, $Mdn = 0.33$) en el éxito de la resolución del problema ($score_p$) resultó ser no significativa, con $W = 2811.5$, $p = .055$, $r = 0.24$ en el total de estudiantes. No obstante, esta diferencia en el éxito con respecto a la elección del trayecto 1 ($Mdn = 0.18$) y el trayecto 0 ($Mdn = 0.33$) sí que resultó ser significativa para el grupo de Educación Primaria con $W = 691.5$, $p = .008$, $r = 0.28$, con tamaño de efecto medio.

4.3.4. Sobre el desempeño en la resolución de los problemas

Como se ha descrito antes, para el grupo de 6 años la elección (*elec*) de uno u otro trayecto resultó afectar significativamente la tasa de acierto ($score_p$) en los problemas. En cuanto a la etapa educativa (ver Figura 8), resultó no haber diferencias significativas entre las tasas de éxito en los problemas de los estudiantes de Educación Infantil (5 años, $Mdn = 0.33$) y los de Educación Primaria (6 años, $Mdn = 0.33$), $W = 4317$, $p = .442$, $r = 0.23$. Sí que se encontraron diferencias significativas con respecto a la etapa educativa en el desempeño de los estudiantes en las tareas de activación (A01 a A04). En este caso la tasa de éxito de los estudiantes de 5 años ($Mdn = 0.80$) resultó significativamente inferior a la de los estudiantes de 6 años ($Mdn = 1.00$) con $W = 2999$, $p = .002$, $r = 0.24$.

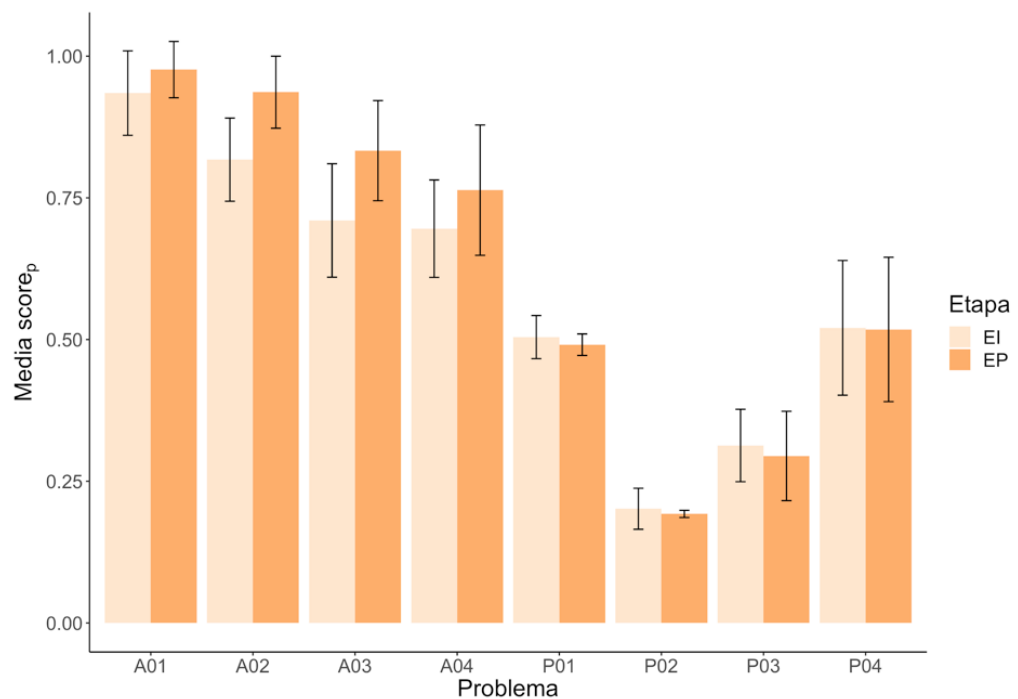


Figura 8. Media de la tasa de éxito ($score_p$) en la resolución del problema por etapa

Por último, reportamos la información relativa al número de intentos iniciados por cada grupo de estudiantes, pues de un total de 389 intentos, 227 correspondieron a estudiantes del último curso de Educación Infantil y 162 a estudiantes de primero de Educación Primaria. Los datos referentes al número de intentos por problema y grupo se adjuntan en la Figura 9. En esta figura queda patente una disminución en el número de segundos intentos (iniciados tras no resolver el problema en el primer intento, etiquetados como 2) para el grupo de estudiantes de 6 años con respecto a sus compañeros de 5 años. Lo mismo se observa para los terceros intentos (iniciados tras no resolver el problema en el segundo intento, etiquetados como 3). Es destacable que la diferencia entre las tasas de acierto obtenidas por los estudiantes de 6 años en el intento 1 (t_1 , $Mdn = 0.20$) con respecto a las obtenidas para el intento 2 (t_2 , $Mdn = 0.55$)

resulte significativa, con $W = 1533$, $p = .033$, $r = 0.24$. Para el resto de comparaciones obtenemos resultados no significativos, también para el grupo de 5 años. Este hecho indicaría una mejora significativa en la gestión del error por parte de los estudiantes de 6 años entre el primer y segundo intento. Una explicación plausible podría estar relacionada con un mayor control y manejo de los procesos de toma de decisiones en situaciones de resolución de problemas, así como mejores conocimientos previos, como se apunta en (Diago et al., 2018c), aunque los datos obtenidos en el presente estudio no han permitido un análisis más profundo sobre esta cuestión.

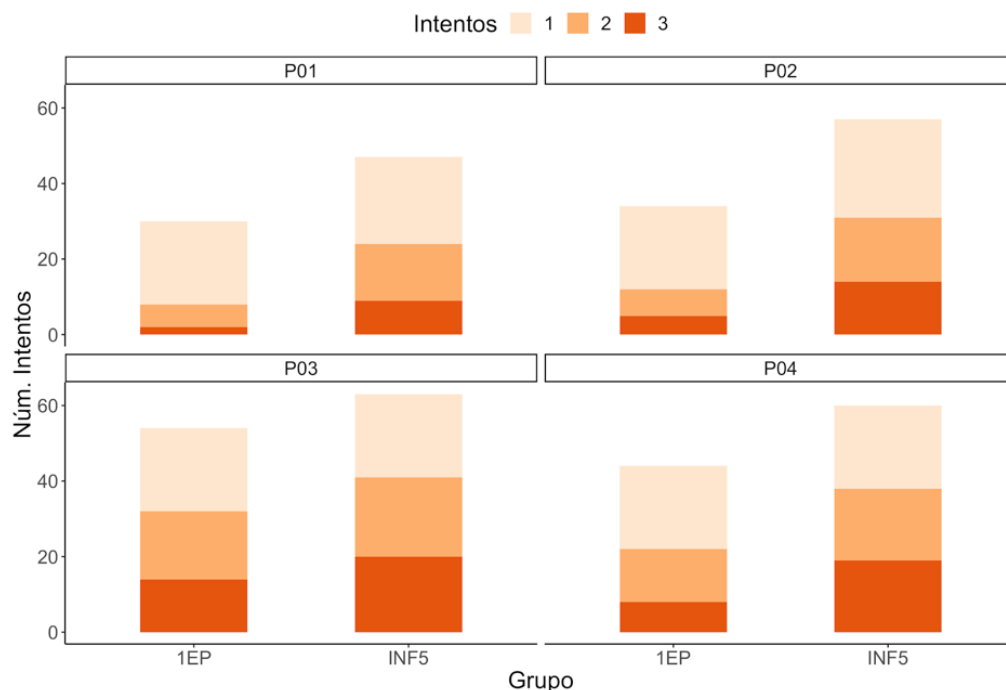


Figura 9. Cantidad de intentos iniciados por grupo de estudiantes para los problemas P01 a P04

5. CONSIDERACIONES FINALES

En este trabajo se han descrito los fundamentos didácticos propios de la resolución de problemas para el diseño de un entorno tecnológico orientado a la monitorización de información, tanto sobre las fases del proceso de resolución de problemas, como de las diferentes operaciones cognitivas asociadas. Este entorno tecnológico puede ser considerado como un ejemplo de aplicación del análisis de métricas de aprendizaje en educación, ya que ofrece información sobre las estrategias de resolución puestas en juego por los estudiantes y, a la vez, en un futuro podría facilitar ayudas a demanda mediante la evaluación de la vía de resolución tomada por el estudiante.

El análisis inicial de los datos del estudio piloto permiten vislumbrar resultados prometedores en relación tanto con variables de proceso como de producto. Entre las de producto encontramos la influencia del giro y del número de pasos en la elaboración de planes, con diferencias significativas en el éxito de la resolución en las tareas de activación entre los estudiantes de 5 y 6 años. También encontramos diferencias de género, en este caso no significativas, entre las tasas de éxito de los problemas de elección de trayecto. Entre los resultados de proceso destacamos cómo la elección de trayecto a la hora de elaborar un plan, y en particular la elección del trayecto sencillo, resulta significativa en relación al éxito en la resolución del problema para el caso de los

estudiantes de 6 años. Así mismo, para estos estudiantes se observa una disminución de los intentos iniciados posteriores al primero. Estos resultados, apuntarían hacia una mejor monitorización y evaluación de la complejidad en este tipo de problemas no trabajados en la escuela en los estudiantes de primer curso de Educación Primaria, como se ha obtenido en estudios para muestras menos numerosas (Diago et al., 2018c).

Somos conscientes de que el entorno tecnológico presentado se encuentra todavía en fase embrionaria. La nueva versión del *software*, en desarrollo, se ha programado en Java y ofrecerá más estabilidad en el funcionamiento que la versión en *Snap!* con la que se llevó a cabo el estudio piloto. Aun con las limitaciones técnicas de la versión inicial del entorno, el software es capaz de facilitar datos que podrán ayudar a investigar en aspectos relacionados con el proceso de resolución de problemas en niveles escolares iniciales como, por ejemplo, cómo afecta la presencia de giros, en qué grado el panel con el histórico de intentos es consultado y tomado en consideración en la elaboración de nuevos planes, la gestión y corrección de errores en los planes elaborados, y demás elementos propios de la resolución de problemas.

Tomando en consideración el “10% de Papert”, recalcamos la importancia del conocimiento didáctico en el diseño de soluciones tecnológicas basadas en *learning analytics*. Con esto en mente, los entornos tecnológicos construidos con esta intención no quedarán limitados a ofrecer medidas básicas que deriven en conclusiones pobres, sino que permitirán generar verdadera información sobre lo que está haciendo el estudiante en cada momento ofreciendo un conocimiento más específico sobre su proceso de aprendizaje. Así, nuestra aportación al 10% de Papert consiste en una nueva mirada a las métricas de aprendizaje como proceso de recogida de grandes datos que, con el conocimiento didáctico específico tomado en cuenta en el diseño de la herramienta, desemboque en una mejor identificación de las características del estudiante, relacionarlas con sus dificultades durante el aprendizaje, y que ello permita implementar secuencias de enseñanza adaptadas a la optimización de su aprendizaje.

REFERENCIAS

- Arnau, D., Arevalillo-Herráez, M., Puig, L., y González-Calero, J. (2013). Fundamentals of the design and the operation of an intelligent tutoring system for the learning of the arithmetical and algebraic way of solving word problems. *Computers and Education*, 63, 119–130. <https://doi.org/10.1016/j.compedu.2012.11.020>
- Beal, C. (2013). AnimalWatch: An intelligent tutoring system for algebra readiness. En R. Azevedo y V. Aleven (Eds.), *International Handbook of Metacognition and Learning Technologies*. Springer International Handbooks of Education (Vol. 28, pp. 337–348). New York: Springer. https://doi.org/10.1007/978-1-4419-5546-3_22
- Beatty, R., y Geiger, V. (2010). Technology, Communication, and Collaboration: Re-thinking Communities of Inquiry, Learning and Practice. En C. Hoyles y J. Lagrange (Eds.), *Mathematics Education and Technology-Rethinking the Terrain* (New ICMI Study Series ed., Vol. 13, pp. 251–284). Boston: Springer. <http://doi.org/10.1007/978-1-4419-0146-0>
- Bers, M. U., Flannery, L., Kazakoff, E. R., y Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers and Education*, 72, 145–157. <https://doi.org/10.1016/j.compedu.2013.10.020>
- Bers, M. U., Seddighin, S., y Sullivan, A. (2013). Ready for Robotics: Bringing Together the T and E of STEM in Early Childhood Teacher Education. *Journal of Technology and Teacher Education*, 21(3), 355–377.
- Bienkowski, M., Feng, M., y Means, B. (2012). *Enhancing Teaching and Learning Through Educational Data Mining and Learning Analytics: An Issue Brief* (U. D. of Education, Ed.). Descargado de <https://tech.ed.gov/wp-content/uploads/2014/03/edm-la-brief.pdf>
- Brownell, W. A. (1942). Problem Solving. En B. Henry (Ed.), *The forty-first yearbook of the National Society for the Study of Education: Part 2, The psychology of learning* (Vol. XII).

- Chicago: University of Chicago Press.
- Caballé, S., y Clarisó, R. (Eds.). (2016). *Formative Assessment, Learning Data Analytics and Gamification in ICT Education*. London: UK: Academic Press.
<https://doi.org/10.1016/C2015-0-00087-9>
- Chang, K. E., Sung, Y. T., y Lin, S. F. (2006). Computer-assisted learning for mathematical problem solving. *Computers and Education*, 46(2), 140–151.
<https://doi.org/10.1016/j.compedu.2004.08.002>
- Chao, P. Y. (2016). Exploring students' computational practice, design and performance of problem-solving through a visual programming environment. *Computers and Education*, 95, 202–2015. <https://doi.org/10.1016/j.compedu.2016.01.010>
- Clements, D. H. (1986). Logo and cognition: A theoretical foundation. *Computers in Human Behavior*, 2, 95–110.
- Coxon, S. V. (2012). The malleability of spatial ability under treatment of a FIRST LEGO league-based robotics simulation. *Journal for the Education of the Gifted*, 35(3), 291–316.
<https://doi.org/10.1177/0162353212451788>
- Diago, P. D., Arnau, D., y González-Calero, J. A. (2018a). Desarrollo del pensamiento computacional en Educación Infantil mediante la resolución de problemas en entornos tecnológicos. En R. Cózar y J. A. González-Calero (Eds.), *Tendencias y tecnologías emergentes en investigación e innovación educativa* (pp. 197–214). Barcelona: Editorial GRAÓ.
- Diago, P. D., Arnau, D., y González-Calero, J. A. (2018b). Elementos de resolución de problemas en primeras edades escolares con Bee-bot. *Edma 0-6: Educación Matemática en la Infancia*, 7(1), 12–41. Descargado de
<http://www.edma0-6.es/index.php/edma0-6/article/view/49>
- Diago, P. D., Arnau, D., y González-Calero, J. A. (2018c). La resolución de problemas matemáticos en primeras edades escolares con Bee-bot. *Matemáticas, Educación y Sociedad*, 1(2), 36–50.
- Drachsler, H., y Kalz, M. (2016). The MOOC and learning analytics innovation cycle (MOLAC): A reflective summary of ongoing research and its challenges. *Journal of Computer Assisted Learning*, 32(3), 281–290. <https://doi.org/10.1111/jcal.12135>
- Drijvers, P., Kieran, C., Mariotti, M. A., Ainley, J., Andresen, M., Cheung, Y., ... Meagher, M. (2010). Integrating Technology into Mathematics Education: Theoretical Perspectives. En C. Hoyles y J. Lagrange (Eds.), *Mathematics Education and Technology. Rethinking the Terrain* (Vol. 13, pp. 89–132). Boston: Springer.
https://doi.org/10.1007/978-1-4419-0146-0_7
- Freeman, A., Adams-Becker, S., Cummins, M., Davis, A., y Hall-Giesinger, C. (2017). *NMC/CoSN Horizon Report:2017 K-12 Edition*. Austin, Texas: The New Media Consortium. Descargado de <http://doi.org/978-0-9988650-3-4>
- Fuglestad, A., Healy, L., Kynigos, C., y Monaghan, J. (2010). Working with Teachers: Context and Culture. En C. Hoyles y J. Lagrange (Eds.), *Mathematics Education and Technology. Rethinking the Terrain* (pp. 293–310). New York: Springer.
<http://doi.org/10.1007/978-1-4419-0146-0>
- Gaudiello, I., y Zibetti, E. (2016). Learning Robotics: Users' Representation of Robots. En I. Gaudiello y E. Zibetti (Eds.), *Learning Robotics, with Robotics, by Robotics: Educational Robotics* (Vol. 3, pp. 1–41). John Wiley & Sons, Inc.
<https://doi.org/10.1002/9781119335740.ch1>
- González-Calero, J. A., Cózar, R., Villena, R., y Merino, J. M. (2018). The development of mental rotation abilities through robotics-based instruction: An experience mediated by gender. *British Journal of Educational Technology, Advance Online Publication*, 1–16.
<https://doi.org/10.1111/bjet.12726>
- Grover, S., Pea, R., y Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education*, 25(2), 199–237.
<https://doi.org/10.1080/08993408.2015.1033142>

- Heffernan, N. T., y Koedinger, K. R. (2000). Intelligent tutoring systems are missing the tutor: building a more strategic dialog-based tutor. En C. Rose y R. Freedman (Eds.), *Building dialogue systems for tutorial applications; papers of the 2000 AAAI fall symposium* (pp. 14–19). AAAI Press: Menlo Park. Descargado de <http://pact.cs.cmu.edu/koedinger/pubs/Heffernan%20&%20Koedinger%2000.pdf>
- Hitt, F., Saboya, M., y Cortés, C. (2017). Task design in a paper and pencil and technological environment to promote inclusive learning: An example with polygonal numbers. En F. Aldon, F. Hitt, L. Bazzini, y U. Gellert (Eds.), *Mathematics and Technology* (pp. 13–30). Cham, Switzerland: Springer International Publishing. Descargado de <https://doi.org/10.1007/978-3-319-51380-5>
- Hoyles, C., y Lagrange, J. (Eds.). (2010). *Mathematics Education and Technology-Rethinking the Terrain: The 17th ICMI Study*. New York: Springer. [10.1007/978-1-4419-0146-0](https://doi.org/10.1007/978-1-4419-0146-0)
- Julià, C., y Antolí, J. (2018). Enhancing spatial ability and mechanical reasoning through a STEM course. *International Journal of Technology and Design Education*, 28(4), 957–983. <https://doi.org/10.1007/s10798-017-9428-x>
- Kilpatrick, J. (1978). Variables and Methodologies in Research on Problem Solving. En L. L. Hatfield y D. A. Bradbard (Eds.), *Mathematical Problem Solving: Papers from a Research Workshop* (pp. 7–20). Columbus, Ohio: ERIC/SMEAC.
- Kilpatrick, J. (1985). A retrospective account of the past 25 years of research on teaching mathematical problem solving. En E. A. Silver (Ed.), *Teaching and Learning Mathematical Problem Solving. Multiple Research Perspectives* (pp. 1–15). Mahwah, NJ: Lawrence Erlbaum Associates. <https://doi.org/10.4324/9780203063545>
- Mah, D. (2016). Learning Analytics and Digital Badges: Potential Impact on Student Retention in Higher Education. *Technology, Knowledge and Learning*, 21(3), 285–305. <http://dx.doi.org/10.1007/s10758-016-9286-8>
- McArthur, D. J., y Lewis, M. W. (1998). *Untangling the Web: Applications of the internet and other information technologies to higher education*. Santa Monica, CA: RAND Corporation.
- Nathan, M. J. (1998). Knowledge and situational feedback in a learning environment for algebra story problem solving. *Interactive Learning Environments*, 5(1), 135–159. <https://doi.org/10.1080/1049482980050110>
- NCTM. (2000). National Council of Teachers of Mathematics. Reston, VA: National Council of Teachers of Mathematics.
- Papert, S. (1980). *Mindstorms: Children, Computers and Powerful Ideas*. New York: Basic Books Publishers.
- Papert, S. (1991). Situating constructionism. En S. Papert y I. Harel (Eds.), *Constructionism* (pp. 1–11). Norwood, NJ: Ablex. Descargado de <http://www.papert.org/articles/SituatingConstructionism.html>
- Papert, S. (1996). An exploration in the space of mathematics educations. *International Journal of Computers for Mathematical Learning*, 1(1), 95–123. Descargado de <http://www.papert.org/articles/AnExplorationintheSpaceofMathematicsEducations.html>
- Pólya, G. (1945). *How to Solve It. A New Aspect of Mathematical Method*. Princeton, NJ: Princeton University Press.
- Rakes, C. R., Valentine, J. C., McGatha, M. B., y Ronau, R. N. (2010). Methods of Instructional Improvement in Algebra. *Review of Educational Research*, 80(3), 372–400. Descargado de <https://www.jstor.org/stable/40927286>
- Reusser, K. (1993). Tutoring systems and pedagogical theory: representational tools for understanding, planning, and reflection in problem solving. En S. Lajoie y S. Derry (Eds.), *Computers as cognitive tools* (pp. 143–177). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Schoenfeld, A. H. (1992). Learning to think mathematically: Problem solving, metacognition, and sense making in mathematics. En D. A. Grouws (Ed.), *Handbook of research on mathematics teaching and learning. A Project of the National Council of Teachers of*

- Mathematics* (pp. 334–370). Reston, VA: National Council of Teachers of Mathematics.
- Shute, V. J., Sun, C., y Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142–158. <https://doi.org/10.1016/j.edurev.2017.09.003>
- Viberg, O., Hatakka, M., Bälter, O., y Mavroudi, A. (2018). The current landscape of learning analytics in higher education. *Computers in Human Behavior*, 89, 98–110. <https://doi.org/10.1016/j.chb.2018.07.027>
- Weber, K., y Leikin, R. (2016). Recent advances in research on problem solving and problem posing. En A. Gutiérrez, G. Leder, y P. Boero (Eds.), *The Second Handbook of Research on the Psychology of Mathematics Education* (pp. 353–382). Rotterdam: Sense Publishers. https://doi.org/10.1007/978-94-6300-561-6_10
- Weintrop, D., y Wilensky, U. (2015). To Block or not to Block, That is the Question: Students' Perceptions of Blocks-based Programming. *Proceedings of IDC 2015: The 14th International Conference on Interaction Design and Children* (pp. 199–208). Association for Computing Machinery, Inc. Descargado de <https://doi.org/10.1145/2771839.2771860>