

# Errores de traducción entre lenguajes de programación secuenciales en actividades con Cubetto

## Translation errors between sequential programming languages in Cubetto activities

Ainhoa Berciano Alcaraz  
Universidad del País Vasco/Euskal Herriko Unibertsitatea. Bizkaia, España  
ainhoa.berciano@ehu.es

María Astrid Cuida Gómez  
Universidad de Valladolid. Valladolid, España  
mariaastrid.cuida.gomez@uva.es

María Luisa Novo Martín  
Universidad de Valladolid. Valladolid, España  
marialuisa.novo@uva.es

### Resumen

El pensamiento computacional, entendido este como la capacidad humana de resolver problemas de distintos ámbitos valiéndose de herramientas de la ciencia de la computación, es una de las habilidades que deben desarrollarse desde edades tempranas. Dentro de los distintos planteamientos educativos para abordar esta tarea, destacan los robots educativos, los cuales son una herramienta valiosa en el aula. Así, en este artículo pretendemos analizar qué tipos de errores, dificultades y frecuencias en dichos errores cometen las maestras y los maestros en formación de Educación Infantil y Primaria cuando se les enfrenta a la resolución de tareas de traducción entre el lenguaje de programación basadas en el uso del robot Cubetto y el lenguaje verbal. Para tal fin, la metodología empleada ha sido de corte mixto, con una muestra de 32 participantes. Los resultados muestran que un porcentaje notable de las maestras y maestros en formación tiene dificultades asociadas a la comprensión del lenguaje de programación del robot Cubetto, a pesar de no ser consciente de dichas dificultades; donde parte de estas se debe a la singularidad de la tabla de programación del robot, lo que puede llevar a errores de transposición didáctica al aula de educación básica.

### Palabras clave

Pensamiento computacional, alfabetización computacional, sistemas de representación, errores de programación, robot Cubetto.

### Abstract

Computational thinking, understood as the human ability to solve problems in different fields using computer science tools, is one of the skills that should be developed from an early age. Among the different educational approaches to tackle this task, educational robots stand out as a valuable tool in the classroom. Thus, in this article we aim to analyse what types of errors, difficulties and frequencies of such errors are made by teachers in training in Infant and Primary Education when faced with the resolution of translation tasks between the programming language based on the use of the Cubetto robot and verbal language. For this purpose, the methodology used was mixed, with a sample of 32 participants. The results show that a significant percentage of trainee teachers have difficulties associated with understanding the programming language of the Cubetto robot, despite not being aware of these difficulties; where part of these difficulties is due to the uniqueness

of the robot's programming table, which can lead to errors of didactic transposition to the basic education classroom.

**Key words**

Computational thinking, computational literacy, representation systems, programming errors, Cubetto robot.

## 1. Introducción

Nuestra sociedad actual es cambiante, lo que supone que la escuela debe promover en el alumnado el desarrollo de habilidades, competencias y destrezas que les permitan adaptarse, dar respuestas eficientes y afrontar los retos de este entorno versátil. Investigaciones previas, como la de Bravo Sánchez y Forero Guzmán (2012), plantean que el uso de la robótica, concebida como una herramienta de aprendizaje, contribuye al desarrollo de estas habilidades.

Mucho antes de hacer un acercamiento a la robótica educativa, Papert (1980, p. 6) ya señaló dos ideas fundamentales: la primera es que es factible construir ordenadores para que los niños y las niñas aprendan a relacionarse con ellos de un modo natural; y la segunda, que aprender a comunicarse con un ordenador puede transformar otros aprendizajes; lo que lleva a que el ordenador puede ser entendido como un propulsor de conocimiento matemático y alfabético. Cuando los niños y las niñas logran acercarse a él y establecer una comunicación, se consigue que tanto la comunicación matemática como la comunicación alfabética dejen de ser algo raro y complicado para convertirse en algo natural y, por tanto, fácil.

Pero en la última década, gracias al avance de la tecnología, el pensamiento y alfabetización computacionales han cobrado relevancia en contextos educativos.

De hecho, en el Real Decreto 95/2022 (Ministerio de Educación y Formación Profesional, 2022a), de 1 de febrero, por el que se establece la ordenación y las enseñanzas mínimas de la Educación Infantil aparece por primera vez la alfabetización digital como capacidad a trabajar en edades tempranas:

Los niños y las niñas se encuentran inmersos en una sociedad en la que lo digital afecta a nuestra forma de comunicarnos, obtener información, aprender y relacionarnos. Es, por tanto, responsabilidad del centro educativo establecer pautas para el desarrollo de hábitos de uso saludables de las herramientas y tecnologías digitales, iniciándose así un proceso de alfabetización digital desde las primeras etapas (p.14597).

En lo que se refiere a la etapa de Educación Primaria, dentro del apartado “Tecnología y digitalización” del Real Decreto 157/2022 (Ministerio de Educación y Formación Profesional, 2022b), de 1 de marzo por el que se establece la ordenación y las enseñanzas mínimas de la Educación Primaria, se proponen:

2. Proyectos de diseño y pensamiento computacional.
  - Fases de los proyectos de diseño: prototipado, prueba y comunicación.
  - Materiales adecuados a la consecución de un proyecto de diseño.
  - Iniciación en la programación a través de recursos analógicos o digitales adaptados al nivel lector del alumnado (actividades desenchufadas, plataformas digitales de iniciación en la programación, robótica educativa...).
  - Estrategias básicas de trabajo en equipo (p. 24420).

Así, se necesita proporcionar al profesorado situaciones de formación para comprender lo que significa el pensamiento computacional, reconocer sus elementos y conseguir una buena disposición hacia su aprendizaje, trabajar habilidades imprescindibles para implementar la introducción del pensamiento computacional en la escuela (Yadav et al., 2017) y potenciar una pedagogía que establezca valores en ideas propias del Pensamiento Computacional (PC) para las primeras etapas educativas (Zapata et al., 2021). Si bien hay que tener en cuenta las carencias y errores en los que incurren maestras y maestros en formación al trabajar el concepto de PC, se reconocen actitudes que pueden resultar positivas para futuras prácticas docentes (González Martínez et al., 2018).

Teniendo en cuenta estas consideraciones, es necesario que los maestros y las maestras en formación estén preparados para ser competentes en todas estas cuestiones. Así, el objetivo de la presente investigación es analizar fundamentalmente los errores que comete el profesorado en formación de Educación Infantil y Educación Primaria al intentar resolver tareas de traducción de programación basadas en el uso del robot Cubetto; en concreto, pretendemos categorizar el tipo de errores, las frecuencias de los mismos, analizar más profusamente los posibles motivos que dan lugar a estos y ver si durante la práctica de la actividad, este alumnado es consciente de las dificultades que entraña la actividad.

## **2. Marco teórico**

En la actualidad, el PC se ha incorporado en los contextos educativos por su aporte en el desarrollo de competencias técnicas, sociales y emocionales (Bers, 2017). Se ha definido con un espectro de habilidades de razonamiento para formular y resolver problemas de forma algorítmica y desarrollar un sentido de fluidez tecnológica (García-Peñalvo y Mendes, 2018; Wing, 2006, 2011). Acorde a investigaciones previas (Bers et al., 2014; García-Valcárcel y Caballero-González, 2019), una de las capacidades incluidas en dicho pensamiento es la de poseer la destreza de resolver un problema por medio de la programación, esto es, ser capaz de programar, que a su vez podemos desgranar en tres dimensiones: la secuencia, entendida como la capacidad de secuenciar una serie de instrucciones en un cierto orden que lleven a la resolución del problema planteado computacionalmente; la correspondencia acción-instrucción, que es la capacidad de establecer una relación biunívoca entre las instrucciones dadas a un robot u objeto programado y la secuencia de pasos que realiza; y la depuración o capacidad de identificar y corregir los errores cometidos en la secuencia de instrucciones y mejorar los pasos dados en la programación. En consecuencia, la programación, además de ser una habilidad increíblemente valiosa, es una forma práctica de enseñar el pensamiento computacional basado en la investigación (Ladzowska y Patterson, 2013). Programar permite participar efectivamente en una sociedad colmada de objetos digitales (Bers, 2018; Kafai, 2016) y ofrece una nueva forma de expresar ideas (Bers, et. al., 2019; Papert, 1980).

En el proceso de enseñanza-aprendizaje de la programación ha emergido recientemente el concepto “código-alfabetización”, definido como un sistema simbólico socialmente situado que permite nuevos tipos de expresiones, así como la ampliación de formas de comunicación preexistentes (Vee, 2017). Se extiende, de esta manera, la definición de alfabetización a los lenguajes de programación, trazando paralelismos entre la lectura y

la escritura de código y la lectura y escritura de textos. Así, la código-alfabetización o codificación se entiende como un nuevo lenguaje que los niños y las niñas pueden aprender desde una edad temprana a través del juego y la creatividad (Bers, 2018), realizando desde las tareas más sencillas y divertidas hasta las más complejas, centrando el progreso del niño en la dificultad de las tareas y en su característica motivadora (Delacruz, 2020; DiSessa, 2001; Duin y Tham, 2019). De acuerdo con Resnick y Rusk (2020), con este proceso de enseñanza-aprendizaje de la lectoescritura con los lenguajes informáticos de programación “los estudiantes no sólo aprenden a programar, sino que programan para aprender” (p.121) y, al igual que sucede con la lectura y la escritura, la código-alfabetización, brinda oportunidades significativas y motivantes de aprendizaje (Resnick et al. 2009). Igualmente, con respecto a estos tres conceptos, pensamiento computacional, programación y codificación, es importante destacar que están estrechamente relacionados, pero son claramente distintos. Según González-González (2019, p.3),

Bers (2017) amplía el concepto de pensamiento computacional, como un proceso expresivo que permite nuevas formas de comunicar ideas. De esta forma, la programación se puede ver como una herramienta para enseñar el pensamiento computacional y podemos ver a la codificación como la escritura conectada con la tecnología.

A este respecto, gracias a la era de la tecnología, que comporta entornos tecnológicos y de programación que están cada vez más presentes en contextos de la vida cotidiana (móviles, tabletas, cámaras digitales, ordenadores, robots educativos, etc.), el número de países y regiones que han incluido en su plan de estudios la introducción tecnológica de la programación en edades tempranas se ha incrementado drásticamente (Bers et al., 2019). Por tanto, para cumplir con las exigencias de la sociedad actual, si se quiere involucrar a las niñas y los niños desde las primeras edades en todo el potencial del pensamiento computacional, se les debe ofrecer un entorno de programación accesible y motivador. Estas oportunidades de aprendizaje se pueden ofrecer a partir de siete ideas claves de la informática en la primera infancia: algoritmos, modularidad, estructuras de control, representación, *hardware/software*, proceso de diseño y depuración (Bers, 2017). Reconociendo al *hardware/software*, podemos establecer una graduación en la complejidad de la programación imperativa, atendiendo en el proceso de codificación si se requiere de la manipulación física (Martín et al., 2002), así, diferenciamos entre:

- La programación tangible, realizada mediante objetos con interfaces tangibles y con capacidad para ser programados; por ejemplo, robots educativos: Bee-Bot, Codioruga, Cubetto, Bluebot...
- La programación no tangible, realizada a través de *software* (ordenadores y tabletas).

En este sentido, García-Peñalvo y Mendes (2018) ponen de manifiesto la importancia de la selección del *hardware/software* en contextos de aula, donde los robots educativos cobran relevancia como ejemplos de programación tangible, debido a que en todo el mundo se está incluyendo la robótica educativa o la programación dentro del currículum.

De acuerdo con Ruiz-Velasco (2007) uno de los principales objetivos de la robótica educativa es formar entornos de aprendizaje ideales, en los que las acciones del alumnado son básicas, ya que planifican, ejecutan acciones reales, las comprueban, contrastan y cometen errores; por una parte, existe una experiencia directa, y posteriormente, el

ordenador/robot ejecuta lo planificado. Y, gracias a esta forma de actuar, se facilitarán al mismo tiempo, otros aprendizajes. Además, la robótica educativa contribuye a la construcción del conocimiento gracias a su versatilidad, suministrando interdisciplinariedad y consiguiendo que el alumnado sea agente activo de su propio aprendizaje (Segatto y Teixeira, 2021).

Para el caso del robot educativo Bee-Boot, Berciano et al. (2022) hacen hincapié en que este robot favorece el pensamiento computacional en edades tempranas, enfatizando también que se debe prestar atención a los errores derivados de las características de dicho robot, debido a que éste realiza giros sobre su propio eje de referencia, sin realizar ninguna traslación. Igualmente, para el caso de maestras y maestros de educación infantil en formación, Seckel et al. (2021) determinan que el error más frecuente cometido aparece en contextos de programación en el plano euclídeo, cuando se ponen de manifiesto instrucciones relacionadas con la orientación espacial (adelante, atrás, izquierda, derecha).

Con respecto al robot educativo Cubetto, Anzoátegui et al. (2017) destacan que los niños y las niñas de 4 a 5 años son capaces de realizar programaciones código a código y, aunque se producen errores, se consigue incrementar sus capacidades cognitivas. En esta misma línea, Faber et al. (2019) usan el Cubetto para proponer una serie de tareas en las que va creciendo el nivel de complejidad algorítmica. Exponen la ejemplificación de los comportamientos de los niños y las niñas cuando realizan las prácticas, y cómo van progresando en el desarrollo de su abstracción de forma paralela a la dificultad algorítmica de cada tarea.

En Marinus et al. (2018) usan el Cubetto para desarrollar una herramienta de evaluación de la capacidad de codificación para niños y niñas de 3 a 6 años, planteando analogías entre la facultad de codificar y el procesamiento cognitivo de expresiones en lenguaje natural.

Hasta aquí se han revisado investigaciones relacionadas con algunas de las ideas de Bers (2017), pero otras todavía están sin explorar. Así, nuestro estudio se centra en la importancia de la representación; esto es, la capacidad de entender y aprender a simbolizar los pasos de un programa, adecuados al entorno de programación utilizado. Restringiendo nuestro interés al Cubetto, es necesario comprender la codificación del robot para ser programado considerando, por un lado, el lenguaje simbólico que usa y, por otro, las características y restricciones que surgen de cómo está construido. Este hecho implica la capacidad de establecer una conexión entre los lenguajes de programación y el lenguaje natural, por medio de la escritura.

En esta línea, en un sentido más amplio, debemos aclarar que el término representación se refiere a una figura, imagen o idea que sustituye a la realidad. De acuerdo con Palmer (1978) la representación es algún tipo de modelo de la cosa que representa. Esta definición presupone la existencia de dos mundos relacionados, pero funcionalmente separados: el mundo representado y el mundo de la representación, que está conformado por los siguientes componentes, dando lugar a un sistema de representación: (1) el mundo o la cosa representada; (2) el mundo de la representación; (3) los aspectos del mundo o la cosa representada que están siendo modelados; (4) los elementos del mundo de la

representación que hacen de modelo; y (5) las correspondencias que existen entre el mundo o cosa representada y el mundo de la representación (p. 262).

En esta misma línea, Martí y Pozo (2000) afirman que cada sistema de representación (escritura, notación numérica, mapas, dibujo, notación musical, etc.) permite recuperar la información asociada a la representación interna del objeto de estudio de un modo diferenciado; pero es necesario trabajar las distintas representaciones para favorecer dicha comprensión. Por ello, se deben trabajar las transcripciones entre los sistemas de representación, lo que requiere conocerlos ampliamente (Pérez-Echeverría et al., 2010) y es importante que el alumnado realice procesos de traslación entre los modelos de representación, debido a que, normalmente, tiene dificultades en pasar de un sistema de representación a otro. Hecho ya contrastado por diversas investigaciones, entre ellas, Duval (1998), Fischbein (1977) y Lesh et al. (1987), donde coinciden en señalar que cuando un estudiante es capaz de trasladar una idea a diferentes sistemas de representación, es porque se entiende esa idea y eso le permite hacer un buen uso de sus representaciones a la hora de resolver un problema.

Volviendo al mundo de la programación, es importante entender qué y cómo debemos representar simbólicamente acorde al dispositivo que queramos programar, usando un lenguaje que él entienda. Este hecho conduce a la necesidad de crear un sistema de representación distinto para cada dispositivo y sus funcionalidades.

Así, tanto los lenguajes asociados a la programación tangible como los asociados a la programación no tangible no son ajenos a esta realidad. Por ejemplo, para el caso de la programación tangible, el Cubetto utiliza los colores para representar diferentes tipos de instrucciones (secuencias), como se muestra más adelante en la Figura 1, y cuando los bloques se juntan sobre el tablero, representan una secuencia de acciones que el robot debe realizar (algoritmo); mientras que, para el caso de la programación no tangible, Scratch también utiliza el color para representar diferentes tipos de instrucciones, las cuales están organizadas categóricamente según su funcionalidad. Así, a medida que las personas progresan en el uso de los lenguajes de programación, aprenden sobre otros tipos de datos y se dan cuenta de que al programar se pueden crear variables para almacenar distintos tipos de valores que representan datos; por lo que es primordial entender que los lenguajes de programación utilizan símbolos para representar acciones. Concebir estos lenguajes como contruidos y diseñados formalmente para dar instrucciones a una máquina, tiene fuertes implicaciones para la alfabetización temprana, la cual a su vez implica la comprensión de un sistema de representación (Bers, 2017).

En nuestro análisis actual, estamos interesadas en cómo el mismo mundo representado, siendo un conjunto particular de ideas y relaciones en el universo de la programación (es decir, líneas de programación), se representa a través de un sistema de codificación de bloques (condicionados por el Cubetto) y, se interpretan en un lenguaje natural, por medio de la escritura, poniendo en juego distintas demandas interpretativas e inferenciales en los futuros maestros y maestras de Educación Infantil y Educación Primaria.

### **3. Metodología**

Para poder dar respuesta a la pregunta de investigación indicada anteriormente, desde un paradigma interpretativo, se ha llevado a cabo una investigación de tipo mixto, con un corte predominantemente cualitativo. Hemos seleccionado dicho método ya que Cameron (2010) establece que las investigaciones con esta modalidad de diseños implican una valiosa combinación o conjunción entre los enfoques cuantitativo y cualitativo.

Según Díaz López (2015) los diseños de investigación mixtos, Mixed Methods Research (MMR), surgen con el objetivo de abrir nuevas vías de investigación ya que, en algunas ocasiones, los dos paradigmas tradicionales de investigación (enfoque cuantitativo y cualitativo) no satisfacen todos los problemas encontrados. Sobre todo, en temas educativos, se pueden usar las bondades de ambos paradigmas para conseguir mejores resultados. Este método representa un proceso ordenado, empírico y crítico de la investigación. Por un parte se realiza un análisis objetivo de la investigación cuantitativa para, posteriormente, evaluar la perspectiva subjetiva de la investigación cualitativa. Incluso así, numerosas investigaciones analizan la fiabilidad de los MMR, van más allá de las formas en las que la investigación cuantitativa y cualitativa se integran hacia una búsqueda de la combinación en la práctica. Una de las dificultades que se encuentran a menudo son las habilidades para examinar de forma integrada los datos (Bryman, 2006; Greene et al.,1989).

En este sentido, Núñez Moscoso (2017) afirma que los MMR significan una ruta legítima de investigación si satisfacen las siguientes características: buena formalización, reflexión en su desarrollo e importancia del contexto de trabajo.

Siguiendo estas ideas, nuestro estudio se clasifica dentro de una metodología de investigación basada en un MMR debido a que, por un lado, pretende analizar de modo cuantitativo los errores y dificultades que tiene el alumnado cuando se plantean actividades para pasar del lenguaje del Cubetto al lenguaje natural, consiguiendo una clasificación de dichos errores en categorías. Y, por otro lado, a partir de dichas categorías ya tenemos los datos necesarios para pasar a la parte cualitativa, eligiendo 9 casos en función de los tipos de errores cometidos. Esta metodología posibilita el fortalecimiento de los conocimientos teóricos y prácticos y supone una recopilación, exploración e interpretación de hechos cuantitativos y cualitativos que hemos seleccionado por ser indispensables para realizar nuestro trabajo.

La muestra ha estado compuesta por 32 estudiantes de cuarto curso de los Grados de Maestro/ a en Educación Primaria y Educación Infantil (87.5% mujeres, 12.5% hombres), con edad una edad comprendida entre los 22 y los 24 años.

#### **3.1. Diseño teórico e implementación de la secuencia didáctica en el aula**

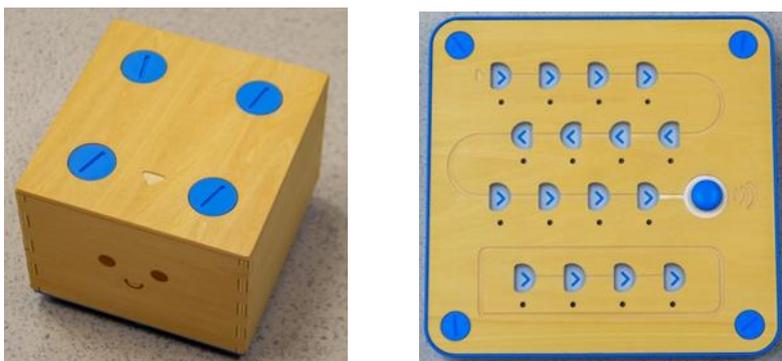
Para poder analizar los errores y dificultades que tiene el alumnado en la comprensión del lenguaje de programación cuando se plantean actividades de programación con el robot educativo Cubetto, se han diseñado distintas secuencias didácticas que establezcan una conexión y requieran de una traducción entre el lenguaje de programación propio del



secuenciación y la codificación, entre otras (Yang et al., 2022, Yu y Roque, 2019; Marinus et al., 2018). En particular, Cubetto es un prisma rectangular de madera, con 11 cm de largo, 10.5 cm de ancho y 8 cm de alto; que se desplaza en línea recta o gira sobre su eje de referencia sin desplazamiento alguno, según las indicaciones que se le indiquen con el tablero de control (ver Figura 2).

### Figura 2

*Cubetto (izquierda) y Tablero de control (derecha)*



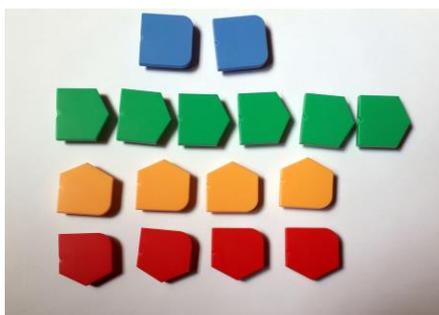
Dicho tablero, de madera de 25cm por 25cm, consta de 4 filas sobre las que se pueden colocar las piezas de programación: las 3 primeras para poner las fichas de programación y la cuarta por si se desea utilizar alguna opción de función, pudiendo generar bucles de programación. Tras apretar el botón redondo azul (al final de la tercera fila), se envía la secuencia al robot por Bluetooth y Cubetto comienza a desplazarse por “el mapa” y ejecutar la secuencia programada (da Silva y González, 2017).

La característica principal del tablero es que las fichas se ponen siguiendo una línea que muestra un sentido de continuidad en el movimiento, pero implica que la segunda fila de instrucciones se coloca de derecha a izquierda, esto es, en sentido inverso a las filas primera y tercera.

**Fichas de programación:** en total hay 16 piezas de diferentes formas y colores de 4 cm de lado (ver Figura 3), cada una representando una acción específica (6 verdes con la instrucción de adelante, 4 rojas de giro a la derecha, 4 amarillas de giro a la izquierda y 2 azules que representan una función y que permiten generar bucles en programación).

### Figura 3

*Fichas del Cubetto*



**Mapa del Cubetto** (ver Figura 1, derecha): cuadrícula cuadrada de 6 unidades de lado, decorada con diferentes dibujos. El sistema de coordenadas muestra una numeración de 1 a 6 en el eje vertical y de letras (A->F) en el eje horizontal, pudiendo determinar de modo unívoco cualquier posición en el plano con una tupla, o pareja de coordenadas atendiendo a sus ejes de referencia).

### 3.2. Recogida de datos y metodología de análisis

Para la recogida de los datos, cada estudiante ha realizado las actividades en folios independientes y dicha información ha sido la analizada de modo cuantitativo y cualitativo.

De acuerdo con Martínez Miguélez (2006), para dar fiabilidad a la metodología cualitativa es adecuado especificar el desarrollo de la teorización que contiene: a) la categorización en clases representativas del material que va apareciendo (justificaciones, hechos, datos) para comprender los elementos descubiertos; b) la distribución que organiza dichas categorías de acuerdo a su naturaleza y las interacciones entre ellas; y c) la teorización, que es un proceso creativo que integra de manera lógica los resultados y los contrasta con el marco referencial. En este caso se trata de entender lo que expresan nuestros estudiantes.

En esta línea, para la detección de errores, se realizó una primera clasificación de las categorías de análisis básicas, acorde a los errores típicos de programación asociados a un lenguaje de programación secuencial (omisión de una orden, traducción errónea de una orden, traducción incorrecta de un bucle y combinaciones de las anteriores). Estos errores bien pueden corresponder a la dimensión de secuencia o a la de acción-instrucción, dependiendo de su naturaleza (ver *Tabla 1*).

*Tabla 1.*  
*Estructura categórica inicial*

<b>Denominación de la categoría</b>	<b>Explicación de su interpretación</b>
Bien	La traducción es correcta
Falta una orden	Se ha omitido un comando
Una orden es errónea	Se ha traducido incorrectamente un comando
El bucle está mal traducido	Se ha traducido incorrectamente un bucle
Falta una orden y el bucle es incorrecto	Se ha omitido un comando y se ha traducido incorrectamente un bucle
Una orden y bucle incorrectos	Se ha traducido incorrectamente un comando y se ha traducido incorrectamente un bucle

Fuente: Elaboración propia.

A posteriori, tras el análisis cualitativo de las respuestas dadas por el alumnado de modo individual, se ha analizado la fiabilidad de los análisis individuales, con la estructura categórica derivada de los errores básicos de programación, calculando el índice de consistencia inter-observaciones independientes, la índice kappa, obteniéndose un valor  $k=.886$ ,  $p=.000$ . Este valor del índice garantiza la fiabilidad del sistema categórico de las observaciones; pero debido a la no concordancia plena ( $k=1$ ), ha dado lugar a un análisis

pormenorizado mediante una triangulación de la clasificación de las observaciones discrepantes, dando lugar a la necesidad de redefinir la estructura categórica acorde a la tabla 2, con la incorporación de una categoría específica relacionada con las características de la plantilla de programación del Cubetto y otras dos sobre los tipos de errores de programación básicos, dando lugar a la siguiente estructura de categorías de carácter excluyente. Posteriormente, se ha vuelto a calcular el índice Kappa,  $k=1$ ,  $p=.000$ , mostrando una concordancia total en la clasificación por parte de las observaciones independientes (ver Tabla 2).

*Tabla 2.*

*Estructura categórica final*

<b>Denominación de la categoría</b>	<b>Explicación de su interpretación</b>
Bien	La traducción es correcta
Falta una orden	Se ha omitido un comando
Una orden es errónea	Se ha traducido incorrectamente un comando
El bucle está mal traducido	Se ha traducido incorrectamente un bucle
Falta una orden y el bucle es incorrecto	
Una orden y bucle incorrectos	
Varias órdenes erróneas	
No interpreta el orden natural de la instrucción del Cubetto	La traducción no sigue el orden establecido por la plantilla del Cubetto.
Mal	La traducción está compuesta por múltiples errores lo que la hace incomprendible.

Fuente: Elaboración propia.

## **4. Resultados**

Para poder responder a la pregunta de investigación, desglosamos los resultados en dos partes. En una primera, mostramos el análisis cuantitativo de los datos, con el fin de destacar el grado de corrección de la traducción realizada por el alumnado y los errores más habituales.

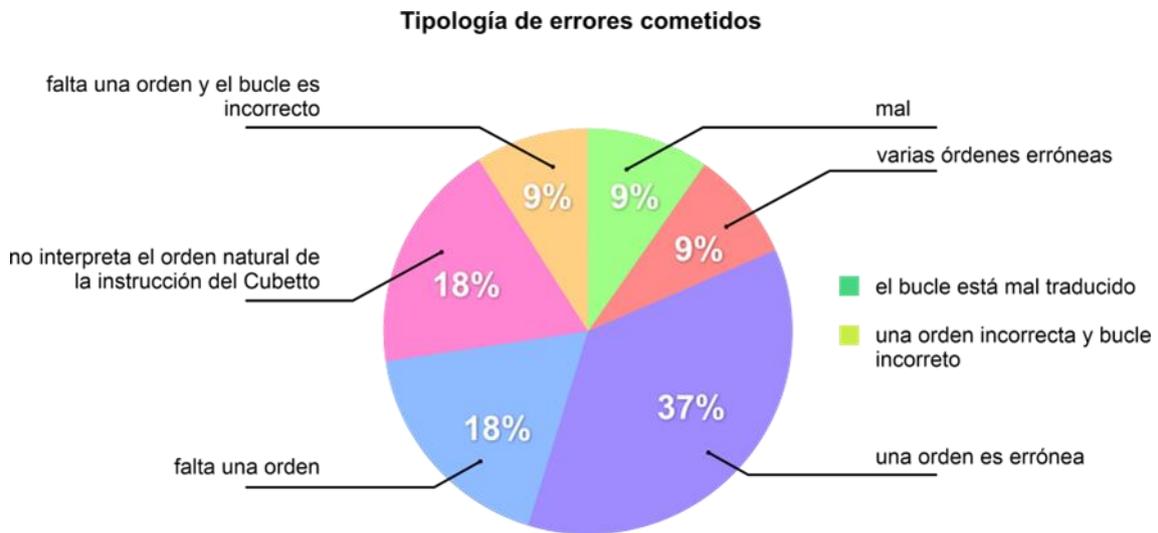
### **4.1. Análisis descriptivo**

Llama la atención el porcentaje notable de alumnado (34%) que se equivoca con la transcripción de las instrucciones dadas por Cubetto, cuando estas deben ser traducidas al lenguaje natural. En concreto, tenemos que un tercio del alumnado se equivoca en un ejercicio de este tipo; lo que pone de manifiesto que no es una actividad tan simple como a priori parecía.

Los errores cometidos se pueden clasificar atendiendo a la categoría establecida en el siguiente gráfico (Figura 4):

**Figura 4**

*Porcentajes del tipo de errores cometidos*

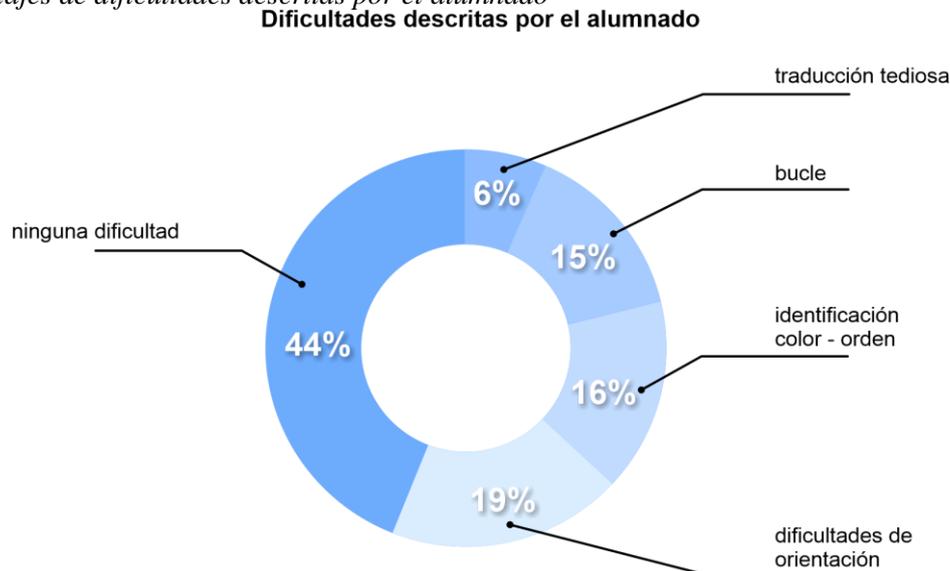


Destacan los errores relacionados con una mala traducción de una orden o la ausencia de una, pero igualmente llaman la atención aquellos errores ocasionados por una mala interpretación del orden natural de la instrucción en Cubetto (un 18% de los errores cometidos).

Igualmente, nos hemos interesado por saber cuánto alumnado ha percibido alguna dificultad en la realización de la tarea y qué tipo de dificultad ha percibido.

**Figura 5**

*Porcentajes de dificultades descritas por el alumnado*



Como se puede ver en la figura anterior (Figura 5), el 56% del alumnado manifiesta algún tipo de dificultad en la realización de la tarea, clasificándose dichas dificultades en las siguientes categorías:

1. Orientación en el plano (19%).

*“He encontrado mayor dificultad a la hora de realizar los giros teniendo en cuenta hacia dónde mira el Cubetto. Más lío al saber hacia dónde gira la izquierda o la derecha cambiando la orientación”.*

2. Identificación color-orden (16%).

*“La dificultad que he encontrado es realizar la correspondencia adecuada a la orden y al color”.*

3. Bucle (15%).

*“Cuando hay dos fichas seguidas azules la repetición”.*

4. Traducción tediosa (6%).

*“La dificultad que he tenido se debe a lo poco económico que resulta la transcripción”.*

## 4.2. Estudio de casos

En el proceso de la investigación cualitativa, una vez analizadas las categorías encontradas es necesario centrar nuestro interés en la interpretación de los datos. Para ello estudiaremos qué dificultades y casuísticas se muestran detrás de algunos de los errores anteriormente descritos. Para ello, analizamos 9 casos relacionados con las diferentes categorías encontradas, que nos permitan entender mejor las características intrínsecas a la traducción entre sistemas de la programación realizada.

### 4.2.1. Dificultades de programación asociadas a las características del tablero de programación del Cubetto

Comenzamos este estudio mostrando aquellas dificultades de programación asociadas a las características del tablero de programación del Cubetto, debido al alto porcentaje de errores detectados. Como hemos mencionado en el apartado anterior, esta característica es que “las fichas se ponen siguiendo una línea que muestra un sentido de continuidad en el movimiento, pero implica que la segunda fila de instrucciones se coloca de derecha a izquierda, esto es, en sentido inverso a las filas primera y tercera”.

Esta característica ha dado lugar a tres tipos de dificultad: una de traducción incorrecta al lenguaje de programación natural, que hemos definido como “no interpreta el orden natural de la instrucción del Cubetto”; otra de uso incorrecto del lenguaje natural, no así de la traducción en sí misma, que hemos denominado “escritura inversa” y una tercera asociada a no entender que la última fila se destina para describir una función (pieza azul), denominada “traducción fila a fila”.

#### 1. No interpreta el orden natural de la instrucción del Cubetto

Dentro de esta categoría encontramos varios casos, en los que se pone de manifiesto la característica principal del tablero, anteriormente descrita.

- En el caso del estudiante A (Figura 6), se observa que el estudiante traduce la instrucción de la fila segunda del tablero de izquierda a derecha, sin tener en

cuenta el sentido de continuidad del recorrido dibujado por la línea curva del tablero. En la siguiente imagen, se ve dicho error en la transcripción desde el comando “gira dos veces a la izquierda” en la primera línea. Esto es, la traducción al lenguaje natural viene marcada por el modo de escritura en este lenguaje (de izquierda a derecha y pasamos al siguiente renglón) y no por la característica del lenguaje de programación del robot educativo usado.

### Figura 6

#### *Transcripción del estudiante A*

Avanza, gira a la derecha, avanza, gira dos veces a la izquierda,  
avanza, gira dos veces derecha y vuelve a avanzar; avanza, gira dos  
veces a derecha y avanza, avanza dos veces, avanza, gira dos  
veces derecha y avanza, gira dos veces a la derecha.

Por otra parte, la dificultad que manifiesta el estudiante no está relacionada con este hecho, sino con que la transcripción le ha resultado poco económica. Entendemos que se refiere a que se necesitan muchas palabras para transcribir, de hecho, ha compactado órdenes para “economizar”.

- Otro caso que llama la atención en esta categoría de error es el del estudiante B (Figura 7), el cual no realiza la transcripción en el orden esperado, sino en orden inverso, comenzando por el final del “camino”. Igualmente, las dificultades que manifiesta están relacionadas con la traducción de la ficha azul: esto es, si cuando hay una ficha azul se trata de una orden, siendo cuatro movimientos, o si son cuatro órdenes.

### Figura 7

#### *Transcripción del estudiante B*

1. Derecha  
2. Derecha  
3. Recto, derecha, derecha, recto  
4. Recto  
5. Izquierda  
6. Recto, derecha, derecha, recto  
7. Recto, derecha, derecha, recto  
8. Recto  
9. Izquierda  
10. Recto  
11. Derecha  
12. Recto

## 2. Escritura inversa

Los estudiantes I y J (Figura 8 y Figura 9 respectivamente) escriben las órdenes de acuerdo con las filas del Cubetto, entendiendo la idea de continuidad de la programación de dicho robot, haciendo uso de una escritura inversa:

### Figura 8

*Transcripción del estudiante I*

Recto → derecha → recto → izquierdo → recto → recto → derecha → derecha  
recto ← izquierdo ← recto ← derecha ← derecha ← recto | ← | recto |  
↳ recto → derecha → derecha → recto | → derecha → derecha

### Figura 9

*Transcripción del estudiante J*

Adelante → Derecha → Adelante → Izquierda  
Adelante Derecha Derecha ← Adelante ← Adelante ← Derecha ← Derecha ← Adelante ← Adelante  
Izquierda  
↳ Adelante → Adelante → Derecha → Derecha → Adelante → Derecha → Derecha

Ambos casos no manifiestan ningún tipo de dificultad cuando se les pregunta por la resolución de la actividad, sin embargo, invierten el orden lógico de lectura en el lenguaje natural.

## 3. Traducción “fila a fila”

En este apartado nos encontramos con casos en los que la traducción de las órdenes al lenguaje natural se ha realizado de acuerdo a las filas que tiene el tablero del Cubetto sin tener en cuenta, por un lado, el cambio de sentido en las instrucciones de la segunda fila; y por otro, el significado de la ficha azul con la incorporación de su traducción en el resultado final del algoritmo en lenguaje natural; esto es, la dificultad se ha hallado en el momento de transcribir los momentos en los que había que tener en cuenta el bucle como es el caso del estudiante K (Figura 10).

## Figura 10

### *Transcripción del estudiante K*

Avanza, gira a la derecha, sigue avanzando, ahora gira a la izquierda y avanza.  
~~Repetir~~ Avanzamos de nuevo, giramos a la derecha y volvemos a girar a la derecha, por último avanzamos.  
Repetimos, volvemos a avanzar, giramos a la derecha, otra vez a la derecha y avanzamos.  
Giramos a la izquierda, avanzamos dos veces, giramos a la derecha, giramos otra vez a la derecha y avanzamos.  
Por último giramos dos veces a la derecha.

#### 4.2.2. *Dificultades asociadas a las características del Cubetto*

En esta sección, detallamos aquellas dificultades asociadas a las características del Cubetto, relacionadas con su sistema de referencia, y que el robot gire sobre su eje sin realizar ninguna traslación.

Encontramos un único caso que comete este error, en el que se observa que una de las acciones se expresa mal, ya que escribe “*giras y vas un paso hacia la derecha*”, en lugar de escribir simplemente “*girar a la derecha*”. Manifiesta que la única dificultad que tuvo fue la confusión de los colores con la acción que corresponde, no siendo consciente del error cometido.

#### 4.2.3. *Dificultades de programación asociadas a la dimensión de acción-instrucción en el lenguaje de programación secuencial*

En este apartado, encontramos errores más habituales, no vinculados a las características del Cubetto, sino a dificultades en la dimensión de acción-instrucción del pensamiento computacional. Así, los errores más habituales (como hemos mencionado anteriormente) se deben a una interpretación incorrecta de una orden o de un bucle u omisión de alguna de ellas.

##### 1. Falta una orden y el bucle es incorrecto

En algunos casos, se observan errores asociados no solo a la traducción incorrecta de la ficha azul, sino también a la omisión de distintas órdenes.

### Figura 11

*Transcripción del estudiante L*

Delante, derecha, delante, izquierda,  
delante, ~~repetición~~ derecha, derecha, delante,  
delante, derecha, derecha, delante, izquierda,  
delante, delante, izquierda, izquierda, delante,  
derecha, derecha.

En este caso (Figura 11), se ve cómo en el primer bucle le falta una instrucción para que la secuencia sea correcta, en concreto, omite la orden “delante”, el segundo bucle lo interpreta bien y en el tercer bucle confunde un giro hacia la izquierda con un giro hacia la derecha, interpretando incorrectamente el significado de la ficha roja (giro a la derecha) con la amarilla (giro a la izquierda), así las palabras que emplea son “delante, izquierda, izquierda, delante”, y tendría que haber puesto: “delante, derecha, derecha, delante”. Lo curioso es que afirma que no encuentra dificultades y con el plano completa bien la tarea.

### 2. Bucle incorrecto

Algunos casos muestran una falta de comprensión con el significado de la ficha azul, esto es, la función, o concatenación de varias órdenes en una misma ficha. Confunde la ficha de repetición con avanzar, no comprende el bucle. En la transcripción sigue las tres filas del Cubetto, pero omite la traducción de las fichas azules (Figura 12). La estudiante es consciente de su dificultad y manifiesta que le ha resultado complicado hacer la transcripción.

### Figura 12

*Transcripción del estudiante M*

- Cubetto avanza hacia la derecha. Una vez ha avanzado gira a la izquierda.
- Después avanza y avanza dos veces más. Gira a la izquierda.
- Luego avanza y repite otra vez hasta que finalmente, gira a la derecha dos veces.

### 3. Otros errores menores

Dentro del resto de errores, que podrían categorizarse como menores, encontramos la traducción de una orden erróneamente, bien sea por omisión de una o por traducción incorrecta puntual del significado de una pieza.

Así, la estudiante G (Figura 13) usa un lenguaje muy preciso, sin embargo, se equivoca, por despiste, en la traducción poniendo “avanza un paso hacia delante” en lugar de “girar a la derecha”.

### Figura 13

#### *Transcripción del estudiante G*

1. Avanza un paso hacia adelante.
2. Sin avanzar, en el sitio, gira a la derecha.
3. Avanza un paso hacia adelante.
4. Sin avanzar, en el sitio, gira a la izquierda.
5. Avanza un paso hacia adelante.
6. Avanza un paso hacia adelante; en el sitio, gira a la derecha; vuelve a realizar otro giro a la derecha; avanza un paso hacia adelante.
7. Repite los mismos pasos que en el punto anterior (6).
8. Sin avanzar, en el sitio, gira a la izquierda.
9. Avanza un paso hacia adelante.
10. Repite los pasos que has realizado en el punto 6.
11. Avanza un paso hacia adelante.
12. Avanza un paso hacia adelante.

Piensa que es sencillo escribir los pasos que hay que seguir. Sin embargo, considera complicado el momento de realizar los giros a la derecha o a la izquierda. Otra dificultad que ha encontrado es estar sentada y no poder representarlo de pie ya que facilitaría encontrar la solución.

## 5. Discusión

La innovación de la presente investigación se debe a proponer un primer acercamiento a los problemas que les puede suponer, a futuros maestros y futuras maestras, la introducción del pensamiento computacional usando robots educativos en el aula, cuando se enfrentan por primera vez al reto de realizar tareas docentes con el Cubetto.

En el análisis cuantitativo se comprueba que un 34 % del alumnado no es capaz de traducir correctamente las instrucciones dadas del Cubetto al lenguaje natural, datos que implican las dificultades que este colectivo tiene en tareas de programación con este robot educativo; dato hasta ahora no analizado por ninguna otra investigación.

Con el fin de entender el origen de este porcentaje tan alto de error detectado, a lo largo del documento hemos ido desgranando el tipo de error que comete el futuro profesorado de educación infantil y educación primaria cuando se enfrentan a estas tareas. Se ha constatado que, aunque el alumnado, en un porcentaje destacable, manifiesta no tener dificultades con las tareas solicitadas con el robot Cubetto, las características del robot inducen a errores de traducción asociadas a la falta de comprensión del modo de

programar con dicho *hardware*, por lo que lleva a errores importantes en la interpretación de los comandos mostrados del robot, referidos a la fase de acción-instrucción de la programación propiamente dicha. En este sentido, se han podido clasificar tres tipos de errores: a) aquellos asociados a no darse cuenta del cambio de sentido del tablero en el orden de las piezas; b) aquellos que han derivado en invertir la escritura en el lenguaje natural; c) aquellos que han llevado a traducir desde el final de la instrucción del robot, y no desde el comienzo.

Por otro lado, otro error, aunque menos habitual, tiene que ver con que el robot gira sobre sí mismo sin realizar ninguna traslación; aspecto importante a la hora de trabajar en el plano euclídeo. Este error o dificultad ya se pone de manifiesto por parte del alumnado participante en la investigación; y, aunque este resultado es completamente nuevo en el contexto del Cubetto; es cierto que, en investigaciones previas con el Bee-Bot como robot educativo seleccionado, ya se había constatado (Berciano et al., 2022, para el caso de niños y niñas de 3 años y Seckel, et al., 2021).

Todos estos hallazgos ponen de manifiesto la importancia de establecer correctamente una correspondencia biunívoca entre los dos sistemas de representación aquí expuestos (lenguaje natural-escritura/ lenguaje de programación del Cubetto). De hecho, atendiendo a la teoría de Palmer (1978), vemos que para parte del alumnado el lenguaje de programación del Cubetto tiene un carácter de índole absoluto, no dependiente del sistema de referencia relativo a su posición (relacionado con el apartado 2 de los sistemas de representación); lo que lleva a establecer un error de correspondencia entre ambos lenguajes.

## 6. Conclusiones

Finalmente, las conclusiones que resultan de esta investigación determinan las pautas a seguir: por un lado, el robot educativo seleccionado ha permitido establecer un puente entre el lenguaje de programación propio del Cubetto y el lenguaje natural, creando conexiones que favorezcan el aprendizaje y la representación y, contribuyendo a la introducción del pensamiento computacional en la formación inicial del profesorado de educación infantil y primaria; además, se ha visto la viabilidad del mismo en el aula, lo que nos lleva a plantear la necesidad de profundizar en el potencial de los robots educativos y su transferibilidad a contextos de aula con niños y niñas de edades tempranas para fomentar el pensamiento y alfabetización computacionales. Sin embargo, no podemos obviar las dificultades detectadas, lo que nos obliga a poner énfasis en que el futuro profesorado debe ser consciente de las características de dichos robots, esto es, de la interconexión entre el *hardware* y el *software* (Bers, 2017); ser capaz de analizar correctamente las características de cada uno de ellos para, posteriormente, determinar los posibles errores de aprendizaje que pueden ocasionar según a qué alumnado se dirija la propuesta didáctica diseñada. En esta línea, investigaciones previas (Segatto y Teixeira, 2021) ya avanzan que el profesorado, una vez instruido en actividades con el Cubetto, es capaz de transferir este conocimiento a actividades encaminadas a sus aulas, fomentando la motivación en su alumnado; pero queda pendiente y, por tanto, resulta necesario, indagar en la conexión entre los distintos sistemas de representación en ámbitos de programación con estos robots (como los aquí descritos), el lenguaje de programación y

la escritura (junto con los modelos manipulables, en este caso los robots educativos) y su repercusión en el desarrollo del pensamiento computacional; de modo paralelo a investigaciones previas en ámbitos como la matemática, donde autores como Lesh et al. (1987), ya establecieron el número necesario de sistemas de representación interdependientes en el proceso de aprendizaje de la matemática (escritura, modelos manipulables, dibujos y diagramas, lenguaje verbal y símbolos escritos).

Presentación del artículo: 30 de diciembre de 2022

Fecha de aprobación: 19 de mayo de 2023

Fecha de publicación: 31 de julio de 2023

Berciano, A., Cuida, M.A., y Novo, M.L. (2023). Errores de traducción entre lenguajes de programación secuenciales en actividades con Cubetto. *RED. Revista de Educación a Distancia*, 23(76). <http://dx.doi.org/10.6018/red.552581>

### Financiación

Este trabajo no ha recibido ninguna subvención específica de los organismos de financiación en los sectores públicos, comerciales o sin fines de lucro.

### Referencias

- Anzoátegui, L. G. C., Jarrín, M. D. C. S. & Pereira, M. I. A. R. (2017). Cubetto para pre-escolares: programación informática código a código. *Atas do XIX Simpósio Internacional de Informática Educativa e VIII Encontro do CIED–III Encontro Internacional*, 114-118.
- Berciano, A., Salgado, M. y Jiménez-Gestal, C. (2022). Alfabetización computacional en educación infantil: Dificultades y beneficios en el aula de 3 años. *Revista Electrónica Educare*, 26(2), 1-21. <https://doi.org/10.15359/ree.26-2.15>
- Bers, M. U. (2017). *Coding as a playground: Programming and computational thinking in the early childhood classroom*. Routledge. <https://doi.org/10.4324/9781315398945>
- Bers, M. U. (2018). Coding and computational thinking in early childhood: The impact of ScratchJr in Europe. *European Journal of STEM Education*, 3(3):08. <https://doi.org/10.20897/ejsteme/3868>
- Bers, M. U., Flannery, L., Kazakoff, E. R. & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, 72, 145–157. <https://doi.org/10.1016/j.compedu.2013.10.020>
- Bers, M. U., González-González, C. & Armas-Torres, M. B. (2019). Coding as a playground: Promoting positive learning experiences in childhood classrooms. *Computers & Education*, 138, 130-145. <https://doi.org/10.1016/j.compedu.2019.04.013>
- Bravo Sánchez, F. Á. & Forero Guzmán, A. (2012). La robótica como un recurso para facilitar el aprendizaje y desarrollo de competencias generales. *Teoría de la Educación. Educación y Cultura en la Sociedad de la Información*, 13(2), 120-136. <https://doi.org/10.14201/eks.9002>

- Bryman, A. (2006). Integrating quantitative and qualitative research: how is it done? *Qualitative research*, 6(1), 97-113. <https://doi.org/10.1177/1468794106058877>
- Cameron, R. (2010). Mixed methods in VET research: Usage and quality. *International Journal of Training Research*, 8(1), 25-39. <https://doi.org/10.5172/ijtr.8.1.25>
- Da Silva, M. & González, C. (2017). PequeBot: Propuesta de un Sistema Ludificado de Robótica Educativa para la Educación Infantil. *Actas del V Congreso Internacional de Videojuegos y Educación (CIVE'17)* ISBN 978-84-697-3849-8 Universidad de la Laguna. Recuperado de: <https://riull.ull.es/xmlui/handle/915/6677>
- Delacruz, S. (2020). Starting from scratch (Jr.): Integrating code literacy in the primary grades. *The Reading Teacher*, 73(6), 805-812. <https://doi.org/10.1002/trtr.1909>
- Díaz López, S. M. (2015). Los métodos mixtos de investigación: Presupuestos Generales y Aportes a la Evaluación Educativa. *Revista Portuguesa De Pedagogía*, 1(1) 7-23. [https://doi.org/10.14195/1647-8614\\_48-1\\_1](https://doi.org/10.14195/1647-8614_48-1_1)
- DiSessa, A. A. (2001). *Changing minds: Computers, learning, and literacy*. MIT Press. <https://doi.org/10.7551/mitpress/1786.001.0001>
- Duin, A. H. & Tham, J. C. K. (2019). Cultivating code literacy: Course redesign through advisory board engagement. *Communication Design Quarterly Review*, 6(3), 44-58. <https://doi.org/10.1145/3309578.3309583>
- Duval, R. (1998). Registros de representación semiótica y funcionamiento cognitivo del pensamiento. In F. Hitt (Ed.), *Investigaciones en Matemática Educativa II* (pp. 173-201). México: Cinvestav.
- Faber, H., Koning, J., Wierdsma, M., Steenbeek, H. & Barendsen, E. (2019, November). Observing abstraction in young children solving algorithmic tasks. In *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives* (95-106). Springer. [https://doi.org/10.1007/978-3-030-33759-9\\_8](https://doi.org/10.1007/978-3-030-33759-9_8)
- Fischbein, E. (1977). *Image and Concept in Learning Mathematics*. Dordrecht, Holanda: Reidel Publishing Company. <https://doi.org/10.1007/BF00241022>
- García-Peñalvo, F. J. & Mendes, A. J. (2018). Exploring the computational thinking effects in pre-university education. *Computers in Human Behavior*, 80, 407-411. <https://doi.org/10.1016/j.chb.2017.12.005>
- García-Valcárcel, A. & Caballero-González, Y. (2019). Robótica para desarrollar el pensamiento computacional en Educación Infantil. *Comunicar: Revista científica iberoamericana de comunicación y educación*, 27(59), 63-72. <https://doi.org/10.3916/C59-2019-06>
- González-González, C. S. (2019). Estado del arte en la enseñanza del pensamiento computacional y la programación en la etapa infantil. *Education in the Knowledge Society (EKS)*. [https://doi.org/10.14201/eks2019\\_20\\_a17](https://doi.org/10.14201/eks2019_20_a17)
- González Martínez, J., Estebanell Minguell, M. & Peracaula Bosch, M. (2018). ¿Robots o programación? El concepto de Pensamiento Computacional y los futuros maestros. *Education in the Knowledge Society (EKS)*, 19(2), 29-45. <https://doi.org/10.14201/eks20181922945>

- Greene, J. C., Caracelli, V. J. & Graham, W. F. (1989). Toward a conceptual framework for mixed-method evaluation designs. *Educational evaluation and policy analysis*, 11(3), 255-274. <https://doi.org/10.3102/01623737011003255>
- Kafai, Y. B. (2016). From computational thinking to computational participation in K-12 education. *Communications of the ACM*, 59(8), 26-27. <https://doi.org/10.1145/2955114>
- Ladzowska, E. & Patterson, D (November 26, 2013). Students of All Majors Should Study Computer Science. *Chronicle of Higher Education*. <https://bit.ly/3Oj55j4>
- Lesh, R., Post, T. & Behr, M. (1987). Representations and translation among representations in mathematics learning and problem solving. In C. Janvier (eds.) *Problems of representations in the teaching and learning of mathematics* (pp. 33-40). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Marinus, E., Powell, Z., Thornton, R., McArthur, G. & Crain, S. (2018, August). Unraveling the cognition of coding in 3-to-6-years old: The development of an assessment tool and the relation between coding ability and cognitive compiling of syntax in natural language. In *Proceedings of the 2018 ACM Conference on International Computing Education Research* (133-141). <https://doi.org/10.1145/3230977.3230984>
- Martí, E. & Pozo, J. I. (2000). Más allá de las representaciones mentales: la adquisición de los sistemas externos de representación. *Infancia y aprendizaje*, 23(90), 11-30. <https://doi.org/10.1174/021037000760087946>
- Martín, G., Toledo, G. y Cerverón, V. (2002). *Fundamentos de Informática y Programación*. Universidad de Valencia
- Martínez Miguélez, M. (2006). Validez y confiabilidad en la metodología cualitativa. *Paradigma*, 27(2), 07-33.
- Ministerio de Educación y Formación Profesional (2022a). Real Decreto 95/2022, de 1 de febrero, por el que se establece la ordenación y las enseñanzas mínimas de la Educación Infantil. *Boletín Oficial del Estado*, 2 de febrero de 2022, 28, 14561-14595.
- Ministerio de Educación y Formación Profesional (2022b). Real Decreto 157/2022, de 1 de marzo, por el que se establecen la ordenación y las enseñanzas mínimas de la Educación Primaria. *Boletín Oficial del Estado*, 2 de marzo de 2022, 52, 24386-24504.
- Núñez Moscoso, J. (2017). Los métodos mixtos en la investigación en educación: hacia un uso reflexivo. *Cadernos de pesquisa*, 47(163), 632-649. <https://doi.org/10.1590/198053143763>
- Palmer, S. (1978). Fundamental aspects of cognitive representation. In Rosch, E. and Lloyd, B (eds.) *Cognition and Categorization*, Lawrence Elbaum Associates. pp. 259-303.
- Papert, S. (1980). *Mindstorms: Children, computers and powerful ideas*. Harvester Press.
- Pérez-Echeverría, M. P., Martí, E. & Pozo, J. I. (2010). Los sistemas externos de representación como herramientas de la mente. *Cultura y Educación*, 22(2), 133-147. <https://doi.org/10.1174/113564010791304519>

- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J. Silverman, B. & Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60-67. <https://doi.org/10.1145/1592761.1592779>
- Resnick, M. & Rusk, N. (2020). Coding at a crossroads. *Communications of the ACM*, 63(11), 120-127.
- Ruiz-Velasco, E. (2007). *Educatrónica. Innovación en el aprendizaje de las ciencias y la tecnología*. Ediciones Díaz de Santos S.A.
- Seckel, M. J., Vásquez, C., Samuel, M. & Breda, A. (2021). Errors of programming and ownership of the robot concept made by trainee kindergarten teachers during an induction training. *Education and Information Technologies*, 27(3), 2955-2975. <https://doi.org/10.1007/s10639-021-10708-8>
- Segatto, R. & Teixeira, A. C. (2021). Utilização do Robô Cubetto em Um Processo De Formação Docente Para Professores Da Educação Básica Na Área Da Robótica Educacional. *Ensino de Ciências e Tecnologia em Revista-ENCITEC*, 11(1), 219-236. <https://doi.org/10.31512/encitec.v11i1.390>
- Vee, A. (2017). *Coding literacy: How computer programming is changing writing*. MIT Press. <https://doi.org/10.7551/mitpress/10655.001.0001>
- Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33–35. <https://doi.org/10.1145/1118178.1118215>
- Wing, J. (2011). *Research notebook: Computational thinking—what and why?* *The link magazine*. Pittsburgh: Spring. Carnegie Mellon University. Recuperado de: <https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>
- Yadav, A., Gretter, S., Good, J. & Mclean, T. (2017). Computational Thinking in Teacher Education. In P. J. Rich & C. B. Hodges (eds.), *Emerging Research, Practice, and Policy on Computational Thinking*, 205-220. Springer. <https://doi.org/10.1007/978-3-319-52691-1>
- Yang, W., Ng, D. T. K. & Gao, H. (2022). Robot programming versus block play in early childhood education: Effects on computational thinking, sequencing ability, and self-regulation. *British Journal of Educational Technology*, 53, 1817-1841. <https://doi.org/10.1111/bjet.13215>
- Yu, J. & Roque, R. (2019). A review of computational toys and kits for Young children. *International Journal of Child-Computer Interaction*, 21, 17-36. <https://doi.org/10.1016/j.ijcci.2019.04.001>
- Zapata, J. M., Jameson, E., Zapata-Ros, M. & Merrill, D. (2021). El Principio de Activación en el Pensamiento Computacional, las Matemáticas y el STEM: Presentación del número especial. *Revista de Educación a Distancia (RED)*, 21(68). <https://doi.org/10.6018/red.498531>