La enseñanza-aprendizaje del Pensamiento Computacional en edades tempranas: una revisión del estado del arte

Preprint · February 2018

DOI: 10.13140/RG.2.2.36740.63368

1 author:

Carina Soledad González González
Universidad de La Laguna
517 PUBLICATIONS **4,967** CITATIONS

SEE PROFILE

González-González C.S. (2018). La enseñanza-aprendizaje del Pensamiento Computacional en edades tempranas: una revisión del estado del arte. En Libro "Pensamiento computacional". Zapata-Ros M. y Villalba Condor K. O. (Coordinadores). Editorial Universidad Católica de Santa María de Arequipa, Perú. (Preprint)

La enseñanza-aprendizaje del Pensamiento Computacional en edades tempranas: una revisión del estado del arte

Carina Soledad González González

Universidad de La Laguna

cjgonza@ull.edu.es

Resumen

Aprender a programar es la nueva alfabetización del siglo XXI. El pensamiento computacional, estrechamente relacionado con la programación, requiere pensar y resolver problemas con diferentes niveles de abstracción y es independiente de los dispositivos de hardware. Aunque en los últimos 10 años se han creado numerosas herramientas para enseñar programación y se han desarrollado numerosas iniciativas en forma de talleres y cursos, entre los que se incluyen países como Singapur, Australia, Reino Unido, entre otros, o en algunas comunidades autónomas españolas, tales como Madrid o País Vasco, existen muy pocos estudios relacionados con la incorporación efectiva en el currículo escolar, el impacto en docentes y estudiantes y sobre metodologías que permitan diseñar actividades educativas en las aulas. Por otra parte, existen estudios que demuestran que los niños menores de 7 años (4-7) pueden aprender conceptos de pensamiento computacional. En este capítulo se analizan las principales iniciativas relacionadas con el pensamiento computacional en las escuelas, el uso de herramientas específicas, tales como los kits de robótica o entornos de programación educativa, y principales estrategias de enseñanza-aprendizaje utilizadas en educación infantil.

1. Introducción

La etapa de educación inicial brinda una oportunidad a los docentes de sentar las bases de una formación integral de calidad mediante la utilización de herramientas innovadoras y la utilización de las tecnologías. En tal sentido, la robótica educativa en la educación infantil se convierte en una herramienta que facilita la adquisición de conocimientos a los niños y niñas de modo lúdico, basándose en los principios de interactividad, las interrelaciones sociales, el trabajo colaborativo, la creatividad, el aprendizaje constructivista y construccionista y el enfoque didáctico centrado en el estudiante, permitiéndoles a su vez la adquisición de destrezas digitales y del desarrollo del pensamiento lógico y computacional de manera subyacente.

La teoría del constructivismo creada por J. Piaget y el construccionismo creada por S. Papert, se basan en explicar cómo el conocimiento en los individuos es adquirido y desarrollado. Ambos autores se fundamentan en el hecho de que el verdadero aprendizaje va mucho más del simple hecho de recibir información o de adherirse a las ideas o valores de otras personas, es expresar nuestras ideas al mundo o encontrar nuestra propia voz e intercambiar nuestras ideas con otras personas (Ackermann, 2001). La filosofía educativa construccionista de Papert emerge del constructivismo de J. Piaget pero añade que la construcción de un nuevo aprendizaje es más eficiente cuando los estudiantes se comprometen en la elaboración, por sus propios medios, de un objeto tangible con alguna representación significativa para estos. Es lo que llama Aprender haciendo (Alimisis et al, 2007). Adicionalmente, señala que las ideas son transformadas cuando son expresadas por diferentes medios, en contexto en particular y en la mente de diferentes personas. Por ello, la robótica educativa se presenta como una herramienta propicia para el aprendizaje mediante la filosofía construccionista ya que permite trasladar la experiencia obtenida

mediante la interacción de la herramienta con el entorno en un determinado contexto, en ideas que transforman las percepciones y conocimientos previos del niño, dando origen al aprendizaje por construcción a través de la experiencia.

Por otra parte, el STEM (en inglés: Science, Technology, Engineering and Mathematics) es el acrónimo empleado para referirse a los conocimientos en las áreas de las ciencias, la tecnología, la ingeniería y las matemáticas, al que recientemente se le ha agregado una A correspondiente al concepto de las artes para finalmente convertirse en STEAM. Cada vez cobra más fuerza la necesidad de incluir estos conocimientos desde los más tempranos niveles educativos, debido entre otras cosas a la necesidad de que los niños conozcan y comprendan conceptos del mundo altamente tecnificado y sistematizado que les rodea, además de convertirse en uno de los objetivos formativos de la agenda educativa de la Unión Europea, apuntando algunos estudios a la importancia de exponer a los niños y niñas de manera temprana a estos conocimientos a fin de evitar la formación de estereotipos y otros obstáculos para su incorporación a posteriori en éstos campos (Elkin, Sullivan & Bers, 2014).

Una revisión de literatura 2003-2009 pidió formas más innovadoras para unir la alfabetización, la tecnología y el aprendizaje, ya que los textos digitales y la tecnología se entrelazan con las habilidades de alfabetización temprana (Burnett, 2010; Van Kleeck, A., & Schuele, 2010). Por tanto, se hace necesario ver la forma de abordar de forma innovadora y transversal la enseñanza de la tecnología y la programación en edades tempranas.

Por ello, en el presente trabajo, se ha realizado una revisión de las principales iniciativas relacionadas con el pensamiento computacional en las escuelas, el uso de herramientas específicas, tales como los kits de robótica o entornos de programación educativa, y principales estrategias de enseñanza-aprendizaje utilizadas en educación infantil.

2. Conceptos fundamentales: codificación, programación y pensamiento computacional

Los términos codificación, programación y pensamiento computacional se usan indistintamente para discutir el desarrollo de habilidades digitales, indispensables para el desarrollo de la ciudadanía del siglo XXI (Bender, Urrea & Zapata-Ros, 2015), pero no hay un claro consenso sobre los mismos, por lo cual se hace necesario clarificar qué significan éstos términos para nuestro trabajo (ECDL Foundation, 2015):

- *Programación:* es el proceso de desarrollar e implementar instrucciones de forma que se permita a un ordenador ejecutar una tarea, resolver un problema y permitir la interacción con humanos.
- *Pensamiento computacional:* es la aproximación hacia la resolución de problemas mediante el uso de estrategias de descomposición, diseño de algoritmos y abstracción, así como razonamiento lógico. El pensamiento computacional implica formular problemas de una manera que permite el uso de un ordenador para resolverlos; organizando y analizando lógicamente datos, representando datos a través de abstracciones, automatizando soluciones a través de algoritmos.
- *Codificación*: Se refiere a la creación de un código en un lenguaje de programación. Permite la intercomunicación entre humanos y máquinas. Asigna un código a "algo". En informática, los términos programación y codificación generalmente se usan indistintamente.

Una vez clarificados los diferentes términos, nos centraremos en el concepto de pensamiento computacional. Aunque recientemente el término "pensamiento

computacional" ha cobrado interés para el mundo académico, sus inicios se remontan a los años 60's, con S. Paper y su enfoque constructivista del lenguaje de programación LOGO, que permitía a los estudiantes crear sus propios procesos de resolución de problemas. Wing (2006), recupera el concepto de pensamiento computacional y lo define como una mezcla entre diferentes formas de pensamiento para la resolución de problemas (ingeniería, matemático, científico) a través de la abstracción formal y del enfoque del mundo real y cotidiano. Wing define al pensamiento computacional como "*la resolución de problemas, el diseño de los sistemas, y la comprensión de la conducta humana haciendo uso de los conceptos fundamentales de la informática*".

El interés por éste término no para de crecer, así como el número de iniciativas para promover su efectiva introducción en las escuelas. Así, desde el Computer Science Teachers Association (CSTA) (https://www.csteachers.org/page/CompThinking) y el International Society for Technology in Education (ISTE) (https://www.iste.org/explore/articleDetail?articleid=152&category=Solutions&article=Computational-thinking-for-all) se han creado un conjunto de herramientas y recursos para que el pensamiento computacional pueda ser desarrollado en las escuelas.

Asimismo, la codificación (coding) es vista como un nuevo tipo de alfabetización. Según Resnik (2014): "*Coding (or computer programming) is a new type of literacy. Just as writing helps you organize your thinking and express your ideas, the same is true for coding. In the past, coding was seen as too difficult for most people. But we think coding should be for everyone, just like writing…With ScratchJr, children aren't just learning to code, they are coding to learn."*

Del mismo modo que en siglos pasados era necesario que los ciudadanos aprendan a escribir, no solo para leer para ser productores de información y conocimiento en vez de solo consumidores, en el siglo XXI, en la sociedad digital de información, es necesario

que los ciudadanos aprendan “codificar” para ser productores digitales de información, no solo consumidores digitales de la misma.

Bers (2017) habla sobre la “codificación” y el “pensamiento computacional”, diciendo que estas ideas no son lo mismo, pero están relacionadas. El pensamiento computacional es la capacidad de usar los conceptos de la informática para formular y resolver problemas. El pensamiento computacional implica un conjunto más amplio de habilidades (por ejemplo, análisis de problemas, pensamiento algorítmico, ...). Usualmente, involucra los conceptos centrales de abstracción, algoritmo, automatización, descomposición, depuración y generalización. Asimismo, puede entenderse como un vínculo directo y como un componente de la "competencia digital". El pensamiento computacional representa un tipo de pensamiento analítico que comparte muchas similitudes con el pensamiento matemático (por ejemplo, resolución de problemas), el pensamiento de ingeniería (diseño y evaluación de procesos) y el pensamiento científico (análisis sistemático). Además, Bers (2017) amplía el concepto de pensamiento computacional, como un proceso expresivo que permite nuevas formas de comunicar ideas. De esta forma, la “codificación” se puede ver como una herramienta para enseñar el pensamiento computacional y podemos ver a la programación como “la escritura conectada con tecnología”. La programación es escribir el código (representación simbólica en un lenguaje informático).

Por tanto, la “codificación” o programación, es la nueva alfabetización, y es necesario comenzar a integrar la alfabetización informática en edades tempranas, especialmente, a través de las tecnologías que soporten el aprendizaje basado en juegos, porque involucran a los niños para que sean creadores, diseñadores, solucionadores de problemas, creadores, artistas ... en resumen, y de esta forma, los niños aprenden a ser productores digitales.

Según Bers (2017), *"La ‘‘codificación’’ promueve experiencias apropiadas para el desarrollo como resolución de problemas, imaginación, desafíos cognitivos, interacciones sociales, desarrollo de habilidades motrices, exploración emocional ... y puede integrarse en diferentes áreas curriculares para promover la alfabetización, matemática, ciencia, ingeniería y las artes a través de un enfoque basado en proyectos"*.

Además, Bers (2017) amplía la noción de pensamiento computacional, definiéndolo como un proceso expresivo y presentando 7 poderosas ideas de pensamiento computacional: 1) algoritmos, 2) modularidad, 3) estructuras de control, 4) representación, 5) hardware / software, 6) el proceso de diseño, y 7) la depuración.

De esta forma, Bers (2017) presenta la "codificación" como una nueva forma para que los niños expresen y compartan sus ideas. En este sentido, puede integrarse en casi cualquier actividad de la clase, con o sin tecnología, como una nueva alfabetización y una nueva forma de pensar, integrada con otras partes del plan de estudios.

Por otra parte, en el contexto de la Agenda Digital europea, la codificación se considera explícitamente como una habilidad clave del siglo XXI: "*La codificación es la alfabetización de hoy y ayuda a practicar habilidades del siglo XXI, como la resolución de problemas, el trabajo en equipo y el pensamiento analítico*" (Bocconi et al, 2016). En la Comisión Europea (2015), considera esencial la adquisición de competencias digitales, incluida la codificación, para sostener el desarrollo económico y competitividad. En la misma línea, la Nueva Agenda de Habilidades invita explícitamente a los Estados Miembros a desarrollar la "codificación / informática" en la educación (Bocconi et al, 2016).

A continuación, se presentan algunas iniciativas de introducción del pensamiento computacional y la enseñanza de la codificación/programación a nivel formal e informal.

3. Iniciativas de introducción del pensamiento computacional a nivel formal e informal

Existen diversas iniciativas en países que han incorporado la enseñanza de la programación a nivel formal en sus currículums escolares, tales como Estonia, Suiza, Finlandia, EEUU, Israel, Singapur o Reino Unido entre otros.

En España existe una normativa a nivel nacional, que puede ser reformulada y concretada en cada Comunidad Autónoma. La situación actual de la enseñanza de la programación/codificación en nuestro sistema educativo es la siguiente (Román-González, 2016):

- A nivel nacional, la codificación/programación forma parte de la asignatura optativa "Tecnologías de la Información y la Comunicación I" de la Etapa de Bachillerato.
- A nivel autonómico, existen distintas Comunidades Autónomas que incluyen en su currículum la enseñanza de la programación/codificación: Cataluña (segundo ciclo de la ESO); Madrid (1º, 2º y 3º de la ESO y Educación Primaria a nivel optativo, fuera del horario escolar), Castilla y León (optativa de 3º y 4º de la ESO), Navarra (4º y 5º de Primaria incluye contenidos obligatorios transversales al área de matemáticas).

Según Román-González (2016) la enseñanza de la programación se está incorporado principalmente en la etapa de Educación Secundaria (13 países), aunque se está incrementando el número de países que lo incorporan también en la etapa de Primaria (10

países), si bien no hay consenso en cuanto a su forma de incorporación: como asignatura obligatoria, optativa o transversal a otras áreas.

El sistema de educación español establece unas directrices para los organismos autonómicos que deben ser seguidas, y que en relación a la enseñanza de la programación, se está introduciendo de forma progresiva y a diferentes ritmos.

Por ejemplo, en la Comunidad autónoma de Canarias, el Área de Tecnología Educativa (Medusa) de la Consejería de Educación y Universidades del Gobierno de Canarias, trabaja de forma coordinada con las asesorías TIC de los centros de formación del profesorado. Actualmente, en el plan de trabajo conjunto, se contempla un área de trabajo sobre Pensamiento Computacional. En cuanto a su presencia en el currículo, desde esta área se elaboraron orientaciones generales sobre pensamiento computacional, las cuales han sido incluidas en un "*documento para la descripción del grado de desarrollo y adquisición de las competencias*".

Como iniciativas que desarrollan el pensamiento computacional en el ámbito no formal podemos encontrar a las siguientes (Llorens-Largo, García-Peñalvo, Molero-Prieto & Vendrell-Vidal, 2017; Fundación Telefónica, 2017; Segredo, Miranda y León, 2017; Bocconi et al, 2016; Google, FECYT y everis, 2016; Espino y González, 2015):

- *A nivel nacional*: Code.Educalab (http://code.educalab.es), Programamos (https://programamos.es/), Genios (https://educagenios.com/), entre otras.
- *A nivel internacional:* Code Week (http://codeweek.eu/), Code.org (https://code.org/), Made with code (https://www.madewithcode.com/), Computing at School (http://www.computingatschool.org.uk/), Coding pirates (https://codingpirates.dk/), Hour of Code (https://hourofcode.com/uk/es), Bebras (http://bebras.org/), Girls in Tech (https://girlsintech.org/), CoderDojo

(https://coderdojo.com/), CT en Google for Education (https://edu.google.com/resources/programs/exploring-computational-thinking/), TACCLE 3 - Codificación (García-Peñalvo, 2016), Computer Science for All (Ciencias de la computación para todos, 2016), entre otras.

Recientemente, el Ministerio de Educación, Cultura y Deporte ha publicado un informe sobre la situación en España de la introducción del pensamiento computacional en las aulas (MECD, 2018). Además, la Sociedad Científica de Informática de España (SCIE) ha realizado un manifiesto "consciente de la importancia creciente para las nuevas generaciones de una formación universal en conocimientos básicos de Informática, manifiesta la necesidad de incluir en el sistema educativo español la materia "Informática", de carácter obligatorio desde Educación Primaria hasta Bachillerato" (SCIE, 2018). Como podemos observar, son escasas las iniciativas a nivel formal para la Educación Infantil y su preocupación en la introducción de este tipo de enseñanzas, siendo el caso de Singapur el más destacado (Sullivan & Bers, 2017). La experiencia pionera de Singapur forma parte de una estrategia educativa nacional, ya que incorporan el pensamiento computacional y la enseñanza de la programación en todo el conjunto de su sistema educativo dentro un programa denominado "Smart Nation Programme Office (SNPO) (https://www.smartnation.sg/)" y específicamente en edades tempranas a través de un programa denominado "Playmaker Programme" (Sullivan & Bers, 2017).

Como ejemplos de iniciativas de introducción de la enseñanza de la programación en la educación infantil a nivel nacional podemos mencionar al Centro público Antonio Machado de Collado de Villalba (Madrid) (Reina y Reina, 2014), quienes han desarrollado diversos proyectos, tales como la programación direccional mediante programación tangible con Bee Bot, la programación a través de tabletas usando Apps (Kodable, Bee Bot App, Daisy the Dinosaur) y la programación de robots con Lego Wedo.

Asimismo, en Canarias se ha llevado a cabo un Estudio sobre la enseñanza de la robótica a nivel infantil utilizando KIBO Robots durante el año 2017, realizando la formación del profesorado y la introducción de forma transversal de la enseñanza de la programación en tres colegios de la isla de Tenerife, colaboración con el grupo ITED de la Universidad de La Laguna y el grupo DEV TECH de Tufts University (EEUU) (Bers, González, Armas & Torres, 2018).

4. Herramientas para la enseñanza de la programación para la educación infantil

Actualmente existen numerosas herramientas para la enseñanza de la programación para primaria y secundaria, tales como: LOGO, Scratch, Alice, App Inventor, Greenfoot, Pencilcode, Agentcheets and Angetcubes,…Sin embargo, son pocos los entornos y herramientas existentes para la enseñanza de la programación a nivel de educación infantil. Entre ellas, podemos destacar las siguientes (da Silva y González, 2017):

- **Robot Turtles** (http://www.robotturtles.com/): es un juego de tablero para que los niños y niñas, a partir de 4 años, aprendan los fundamentos de la programación (Fig. 1). En este juego, uno de los jugadores se convierte en el "motor de tortugas" y el resto son "maestros de tortugas". Los maestros deben conseguir que su tortuga avance por el tablero y los diferentes laberintos hasta llegar a la gema de su color.

Fig. 1. Robot Turtles. Fuente: http://www.robotturtles.com/

- **Hello Ruby** (http://www.helloruby.com/): es un cuento infantil a partir del cual se puede enseñar a los niños y niñas a programar, donde cada lenguaje de programación es un personaje. Incluye actividades e ideas adicionales en su página web para aprender a programar.

Fig. 2. Hello Ruby. Fuente: http://www.helloruby.com/

- **ScratchJr** (https://www.scratchjr.org/): es un lenguaje de programación visual diseñado por el MIT en colaboración con el DEV TECH de Tufts University, para la enseñanza de la codificación para niños y niñas de 5 a 7 años de edad (Fig. 3). Permite crear y razonar a niños y niñas que no saben leer aún. Está disponible de forma libre para iOS, Android y Chromebook.

Fig. 3. Interfaz de Scratch Jr. Fuente: https://www.scratchjr.org/learn/interface

- **Kodable** (https://www.kodable.com/): es una herramienta educativa para enseñar a programar a niños y niñas a partir de los 5 años de edad (Fig. 4). Cuenta con diversos niveles de dificultad y con distintos contenidos que permiten practicar secuencias, bucles, variables, condicionales, operaciones algorítmicas, resolución de problemas, habilidades comunicativas, pensamiento crítico, etc., así como otros contenidos relacionados con el pensamiento computacional.

Fig. 4. Kodable. Fuente: https://www.kodable.com/

- **Cargobot** (https://twolivesleft.com/CargoBot/): es una app para el iPad que, a través de un lenguaje llamado Codea, permite aprender a programar como si fuera un juego (parecido al tetris) (Fig. 5).

Fig. 5. Cargobot. Fuente: https://twolivesleft.com/CargoBot/

- **LightbotJr** (http://lightbot.com/): es una app que permite aprender a programar a través de la resolución de puzles (Fig. 6). En este juego los niños y niñas deberán

darle instrucciones lógicas al robot para que se mueva a través de un camino hasta el enchufe que permitirá encender la luz de la habitación.

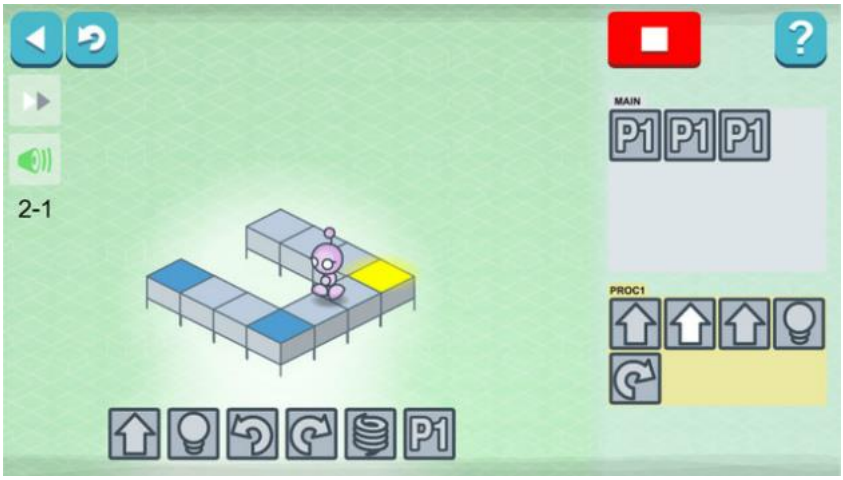

Fig. 6. LightbotJr. Fuente: http://lightbot.com/

A continuación, se mencionan algunos de los juguetes robóticos más empleados o considerados más adecuados para ser utilizados en las aulas de educación infantil.

- **KIBO Robots**: KIBO ha sido desarrollado por Kinderlabs Robotics en la Universidad de Tufts y es un robot con sensores de sonidos, luces y distancia, y un lector de código de barras que es el medio por el que le son ingresadas todas las instrucciones de programación. KIBO (Fig. 7) viene acompañado de una serie de bloques de madera en los que se encuentran representadas mediante imágenes y colores, diferentes instrucciones de programación que incluyen *loops* (ciclos de repeticiones) y declaraciones condicionales, acompañadas de un código de barras. Con estos bloques, los niños y niñas van conformando la secuencia de pasos que conforman su algoritmo y cuyos códigos de barras serán posteriormente capturados uno a uno mediante el lector del que para tal fin dispone el robot. KIBO posee una apariencia que luce como si no hubiese sido concluido su diseño físico, lo que invita a los niños a completarlo mediante el uso de materiales que permitan su

decoración y personalización, involucrándolos e inspirándolos en la creación de proyectos con un significado propio en el que poder desarrollar destrezas creativas y artísticas.

Fig. 7. KIBO Robot. Fuente: http://kinderlabrobotics.com

- **BEE BOT y BLUE-BOT:** Sin duda el Bee-Boot (Fig.8) es uno de los dispositivos más utilizados y estudiados actualmente en las aulas de educación infantil para la iniciación de los estudiantes en los conceptos de robótica, así como para la adquisición de competencias en diferentes áreas del conocimiento, como la lecto-escritura, matemáticas, arte, entre otras. El robot es operado mediante las teclas que posee en la carcasa, las cuales emplean básicamente los comandos de paso atrás, paso adelante (15 cm.), giro a la derecha, giro a la izquierda (ambos de 90º), pausa y un botón para ejecutar la secuencia de pasos a las que se ha dado entrada mediante el uso de los botones anteriores, además de otro que permite iniciar el dispositivo (clear) para introducir una nueva secuencia de comandos, el costo del BEE-BOT se encuentra alrededor de los 80€ por unidad.

Fig.8. Bee Bot. Fuente: https://www.bee-bot.us/

El Blue-Bot (Fig. 9) es prácticamente igual que el Bee-Boot, con la prestación añadida de que adicionalmente puede ser programado mediante una aplicación para dispositivos móviles (Android e IOS) con una interfaz gráfica bastante fácil de manipular para los niños. Su precio es de 140 € por unidad.

Fig.9. Blue-Bot. Fuente: https://www.bee-bot.us/bluebot.html

El fabricante de Blue-Bot ha sacado recientemente al mercado un lector táctil de programación en el que los niños colocan fichas individuales con una instrucción determinada, e ir así construyendo el algoritmo. Una vez finalizado éste y presionada la tecla 'Go', el set de instrucciones será ejecutado por el robot. Este dispositivo es vendido por separado. En Reino Unido su precio se encuentra en 96 £ . Adicionalmente, ambos robots operan sobre una alfombrilla cuadriculada, cuyo precio oscila sobre sobre los 35€ (precio página web Ro-botica) la unidad (dependiendo del contenido de aprendizaje de la misma).

- **Roamer:** es un robot educativo desarrollado por Valiant Technology con la finalidad de facilitar los procesos de enseñanza desde el primer ciclo de educación infantil hasta 6to grado de primaria cuya principal característica es que puede ser adaptado a diferentes edades y niveles de habilidades y conocimientos, mediante el cambio del módulo de teclado en la parte superior del dispositivo (Fig.10).

Fig. 10. ROAMER. Fuente: http://www.roamer-educational-robot.com/

Este es el módulo de teclado utilizado para edades entre 5 y 7 años. En él se puede observar que se manejan patrones de velocidad, distancia y ángulo de giro, introduciendo además el concepto de repetición mediante los números que se muestran en el teclado. Su precio en Reino Unido según muestra la página web de Valiant, es de 95£ antes de impuestos. Es de destacar que en la página web de Valiant, se encuentra disponible una plataforma Moodle con formación sobre el uso de Roamer, una librería con actividades, una revista con proyectos e ideas educativas para su implantación en el aula, un apartado dedicado a la investigación en el uso del dispositivo, y un foro que ofrece la posibilidad de formar parte de una comunidad de usuarios a fin de compartir experiencias de uso de la herramienta. El valor añadido que presenta este robot es el potencial que posee para el desarrollo de conceptos matemáticos en la niñez (Misirli & Komis, 2014). Por otra parte, la creación de un pseudo-lenguaje mediante la representación gráfica de los comandos del robot Roamer es una estrategia de enseñanza apropiada para la visualización de los procedimientos de programación.

- **PRO BOT:** es un robot basado en el diseño de la TORTUGA del lenguaje LOGO de Papert, y reviste algo más de complejidad que el BEE BOT y BLUE BOT (Fig.

11). Posee sensores de luz, sonido y contacto y puede ser operado mediante los botones que dispone sobre su carcasa, o mediante conexión USB al ordenador y el software PROBOTIX (bajo licencia). A diferencia del BEE BOT y BLUE BOT, el PRO BOT requiere que le sea introducida la distancia a recorrer que puede ser desde 1m hasta 5 m (hacia adelante o hacia atrás), así como el ángulo de giro que puede ir desde 1º hasta 500º. Se introduce el concepto de *loops* o número de veces que puede ser repetida una acción, y los programas elaborados pueden ser almacenados para posteriormente ser ejecutados como un procedimiento (llamados de una acción). En el centro dispone de un agujero en el que puede ser insertado un rotulador para poder hacer dibujos durante la ejecución de una secuencia de programación, con lo que son muchas las acciones extras que pueden ser ejecutadas con este robot. Su precio es de 135 € la unidad, sin la licencia del software PROBOTIX.

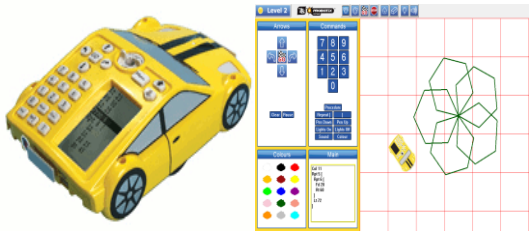

Fig. 11. PROBOT. Fuente: http://ro-botica.com/es/Producto/PRO-BOT/

- CUBETTO: es un robot diseñado por Primo Toys en el que la secuencia de comandos que conforman la programación del robot, es realizada mediante la inserción de bloques en diferentes slots de un tablero que representa la interfaz con el robot (Fig. 12.). Los bloques poseen diferentes formas y colores, representando cada uno de ellos una acción específica: adelante, giro derecha, giro izquierda y función. El tablero se encuentra dividido en dos partes, una parte superior con doce

slots que guían la secuencia y el lugar en el que los niños deben ir colocar los bloques de instrucciones, y una parte inferior con 4 slots, en el que de igual modo mediante el uso de los diferentes bloques los niños podrán introducir una secuencia de programación pero que en este caso podrá ser invocada desde los slots superior como una función o procedimiento (para ello el bloque de función). Una vez introducidas las piezas deseadas, el envío del programa se realiza vía bluetooth al pequeño y sencillo robot representado por una cajita de madera construido con Arduino. Cabe destacar que CUBETTO cuenta con el visto bueno del método Montessori. Su precio es de 225 US$ la unidad.

Fig. 12. CUBETTO. Fuente: https://www.primotoys.com/

- **CODE A PILLAR**: creado por FISHER PRICE para introducir los conceptos de programación en niños de pre-escolar a fin de promover las destrezas de pensamiento computacional y de solución de problemas (Fig. 13). El juguete viene con 8 segmentos que se conectan entre sí mediante un puerto USB, representando cada uno de ellos un determinado comando o instrucción (adelante, derecha, izquierda, giro, sonido). El orden en el que son conectados los bloques determinará la secuencia de programación que ejecutará el robot (a diferentes entradas, diferentes resultados) una vez presionado el botón “start”.

Fig. 13. CODE A PILLAR. Fuente: http://www.fisher-price.com/en_US/codeapillar/index.html

- TANGIBOT: es un proyecto que ha sido creado por un equipo de investigadores de la Universitat Politècnica de València (UPV), dirigido a niños de entre 3 y 5 años, en el que un robot LegoTM Mindstorms® Ev3 al que se le ha incorporado un lector RFID en su chasis y un teléfono móvil (Fig. 14). Por otra parte, en objetos como lápices de goma, pelotas y otros tangibles, se han incorporado etiquetas RFID (codificadas todas ellas para que el robot responda con diferentes tipos de comportamiento) que al ser detectadas por el lector son enviadas al móvil para ser procesadas y enviadas en forma de comandos al robot.

Fig. 14. Niños interactuando con el robot TANGIBOT. Fuente: UPV.

- **ROOT**: ha sido desarrollado por "The Wyss Institute" de la Universidad de Harvard con la finalidad de incentivar el aprendizaje de los principios de

programación en las escuelas (Fig. 15). El robot cuenta con una amplia cantidad de prestaciones entre las que destacaría la particularidad de que puede adherirse a superficies en vertical como lo son las pizarras blancas metálicas, pudiendo así el conjunto de la clase visualizar su comportamiento sobre dicha superficie, a la vez de los trazados que puede realizar mediante la incorporación de un rotulador a su carcaza. Puede responder a estímulos del entorno como son los luminosos, y a puntos de contacto en su superficie, es capaz de reconocer hasta 32 colores de líneas a seguir, y puede ser programado para reproducir diferentes notas musicales. Su programación es realizada mediante una aplicación para dispositivos móviles denominada SQUARE, la cual puede ser operada mediante bloques gráficos, o mediante sentencias de programación en formato textual, permitiendo así adaptarse a s desde niveles educativos iniciales, hasta aquellos con edades y capacidades de abstracción superior, permitiendo evolucionar con los estudiantes a medida que van creciendo y convirtiéndose de este modo en una herramienta adaptable a diferentes niveles educativos. Su precio es de $199 y estará disponible en junio de 2018.

Fig.15. Robot ROOT y SQUARE. Fuente: http://www.rootrobot.io/

5. Estrategias pedagógicas para la enseñanza de la programación en la etapa de infantil

Existen distintas recomendaciones para poder introducir la enseñanza de la programación en edades tempranas. Martin, Toledo y Cerverón (2002) diferencian entre la programación tangible, realizada mediante objetos con interfaces tangibles y con capacidad para ser programados (e.j. robots) y la programación no tangible, realizada a través de software (ordenadores y tabletas). La programación tangible es la recomendada para tempranas, aunque a partir de los 4-5 años ya puede introducirse la programación gráfica táctil a través de tabletas y ordenadores (Bers & Horn, 2010).

Tabla 1. Elementos de STEM/STEAM que pueden integrarse en el currículum de infantil a través de la robótica y la programación:

STEM/STEAM	**Elementos a trabajar en el curriculum**	**Referencias**
Tecnología (Technology)/ Ingeniería (Engineering)	Metodologías de diseño de proyectos de ingeniería Conceptos de robótica y programación	Bers et al. 2002; Cejka et al. 2006; Sullivan et al. 2013; Sullivan and Bers 2015.
Matemáticas (Mathematics)	Números Secuencia Estimación Conteo Tamaño / Formas Resolución de problemas Metacognición Razonamiento	Bers 2008; Clements and Meredith 1992, Clements et al, 2011; Highfield et al. 2008; Kazakoff et al. 2013; Resnick et al. 1998; Kazakoff et al. 2013; Clements and Gullo 1984; Navarro et al, 2012; Resnick et al, 2007.
Ciencia (Science)	Exploración de los sentidos Causa y efecto Observación estructurada	Bers 2008; Kazakoff et al. 2013.

Artes (Arts)	Desarrollo de la motricidad fina Coordinación visoespacial Colaboración y trabajo en grupo Creatividad	Hamner and Cross, 2013; Lee et al. 2013.

Algunos autores recomiendan distintas actividades para niños entre 3 y 6 años para introducir el pensamiento computacional (CSTA e ISTE, 2011; Pane & Myers, 2001). Por ejemplo:

- A partir de los 3-4 años: actividades de producción y ejecución de instrucciones, principalmente vinculadas al propio cuerpo y acción y trabajo con objetos manipulables (programación tangible).
- Entre los 4-5 años: programación tangible manipulativa, incorporación de programación a través de interfaces naturales táctiles (interacciones de tipo arrastrar-soltar comandos con representación visual (instrucciones gráficas).
- Entre los 5-6 años: programación tangible y táctil, posibilidad de introducción de comandos con algunas palabras (instrucciones simples).

Asimismo, recomiendan mantener un enfoque globalizador, con metodologías de aprendizaje activo, aprendizaje colaborativo, aprendizaje basado en juegos, aprendizaje basado en proyectos (Nacher, Garcia-Sanjuan & Jaen, 2015; Eck et al, 2013; Janka, 2008). Y, destacan que en la evaluación de trabaje la habilidad de aprender a aprender y la autoevaluación, la reflexión y la generalización (Vilalta García, 2016).

Por otra parte, existen distintas metodologías relacionadas con la enseñanza de la tecnología. Entre ellas destacamos: a) el Modelo TPACK (en inglés: Technological, Pedagogical, Content and Knowledge) que es una extensión de la expresión Pedagogical Content Knowledge de Shulman (1986) (PCK) (Koehler & Mishra, 2009), b) el TIM (en inglés: Arizona Technology Integration Matrix) (Hornak, 2011), y c) PTD (en inglés: Positive Technological Development), que fue desarrollado especialmente para la enseñanza de la tecnología en edades tempranas por la profesora Marina Umaschi Bers (Bers, 2008; Bers, 2012).

El marco pedagógico del PTD guía el desarrollo, la implementación y la evaluación de programas educativos que utilizan las nuevas tecnologías para promover el aprendizaje como un aspecto del desarrollo positivo de los niños y niñas. El marco pedagógico PTD es una extensión natural de la alfabetización informática y los movimientos tecnológicos que han influido en el mundo de la educación, añadiendo a los elementos cognitivos componentes psicosociales y éticos. Desde un punto de vista teórico, el marco pedagógico PTD se basa en un enfoque interdisciplinario que integra ideas de los campos de la comunicación mediada por ordenador, el aprendizaje colaborativo apoyado por ordenadores y la teoría construcionista del aprendizaje desarrollada por Seymour Papert (1980).

El PTD propone seis comportamientos positivos (seis C) que deben ser apoyados por actividades educativas que utilizan nuevas tecnologías, como, por ejemplo, la robótica. Estos seis comportamientos son: la creación, la creatividad, la comunicación, la colaboración, la construcción de la comunidad y las opciones de conducta.

A continuación, se ejemplifican estos 6 comportamientos con diferentes actividades utilizando el robot KIBO:

1. Creación de contenidos: El proceso de diseño de ingeniería de la construcción y el pensamiento computacional involucrados en la programación fomentan la competencia en la alfabetización informática y la fluidez tecnológica. El uso de los “Diarios de Diseño de Ingeniería” permite que los propios niños y niñas, así como para los docentes y la familia, puedan documentar su propio pensamiento, sus trayectorias de aprendizaje y la evolución del proyecto desarrollado con el robot KIBO en el tiempo.

2. Creatividad: realizando y programando proyectos personalmente significativos, resolviendo problemas de forma creativa y lúdica, integrando diferentes medios como robótica, motores, sensores, materiales reciclables, creando arte a través de un lenguaje de programación tangible con KIBO.

3. Colaboración: involucrando a las niñas y niños en un ambiente de aprendizaje que promueva el trabajo en equipo, compartir recursos y preocuparse unos de otros mientras trabajan con sus robots KIBO. Al principio de cada jornada de trabajo, cada niño/a recibe, junto con su diario de diseño, una impresión personalizada con su fotografía en el centro de la página y las fotografías y nombres de todos los demás niño/as de la clase dispuestos en un círculo. A lo largo del día, con el mensaje del docente, cada niño/a dibuja una línea de su propia foto hacia las fotos de los niño/as con los que ha colaborado. La colaboración se define aquí como obtener ayuda o ayudar con un proyecto, programar juntos, prestar o pedir prestado materiales o trabajar juntos en una tarea común. Al final de la semana, los niño/as deben escribir o dibujan "tarjetas de agradecimiento" a los niño/as con los que han colaborado más.

4. Comunicación: a través de mecanismos que promuevan un sentido de conexión entre pares o con adultos. Por ejemplo, a través del uso de los círculos

tecnológicos, cuando los niños/as dejan de trabajar, ponen sus proyectos sobre la mesa o el piso y comparten su proceso de aprendizaje. Los círculos tecnológicos representan una buena oportunidad para resolver problemas como comunidad.

5. Construcción de la comunidad, promoviendo la contribución de ideas en la comunidad de aprendizaje. Los proyectos finales realizados por los niños y niñas son compartidos con la comunidad a través de una demostración o una exposición abierta. Estas demostraciones o exposiciones brindan oportunidades para que los niños y niñas compartan y celebren el proceso y los productos tangibles de su aprendizaje con la familia y los amigos. A los niños y niñas se le da la oportunidad no sólo de ejecutar su robot, y al mismo tiempo, pueden desempeñar el papel de docente al explicar a su familia cómo lo construyeron, programaron y trabajaron a través de problemas.

6. Opciones de conducta, que proporcionan a los niños y niñas la oportunidad de experimentar con preguntas de "qué pasa si" y las consecuencias potenciales de cada opción. Estas preguntas permiten examinar los valores y explorar los rasgos de carácter de los niños y niñas al mismo tiempo que se trabaja con la robótica.

Asimismo, siguiendo el método propuesto por Bers (2012) se pueden integrar curricularmente la enseñanza de la tecnología y la ingeniería a través de distintas actividades organizadas en torno a ideas poderosas de los ordenadores, la programación y robótica, tales como: la programación de ordenadores, el proceso de diseño de ingeniería, secuencias de comandos y flujo de control, bucles, sensores, parámetros y condicionales. Además, se pueden combinar y trabajar de forma transversal con lenguas, matemáticas, ciencias sociales, ciencias y las artes por medio de diferentes actividades educativas, tales como círculos tecnológicos, diarios de tecnología, diseño del proceso de ingeniería, etc.

6. Conclusiones

En este capítulo se han presentado diferentes conceptos relacionados con la codificación, programación y pensamiento computacional, que aparecen en la literatura y sobre los cuales no hay consenso. Se han clarificado éstos conceptos para luego centrar la discusión cómo abordar desde un punto de vista pedagógico y tecnológico la enseñanza de los mismos en la etapa de educación infantil.

Asimismo, se han presentado una revisión sobre diferentes iniciativas a nivel formal e información de introducción curricular de la enseñanza de la codificación, programación y pensamiento computacional nacionales e internacionales. Vemos que son escasas las experiencias de introducción de estos conceptos y habilidades en edades tempranas. Asimismo, son escasos los estudios sobre los resultados a largo plazo en el aprendizaje de las habilidades relacionadas al pensamiento computacional así como principios pedagógicos y metodologías que guíen al profesorado en el diseño de experiencias de aprendizaje basadas en tecnologías para la etapa de educación infantil. Por ello, este trabajo aborda una revisión de las principales tecnologías y herramientas especialmente adecuadas para estas edades, así como los principios metodológicos para poder introducirlas de forma efectiva.

Finalmente, se presentan diversas recomendaciones sobre estrategias y marcos pedagógicos para la enseñanza de la programación, la robótica y la ingeniería en la educación infantil, destacándose el marco de Desarrollo Positivo de la Tecnología (PTD) como marco de referencia para la introducción curricular efectiva y transversal de los conceptos fundamentales de tecnología e ingeniería en edades tempranas.

Bibliografía

Ackermann, E. (2001). Piaget's constructivism, Papert's constructionism: What's the difference. Future of learning group publication, 5(3), 438.

Alimisis, D., Moro, M., Arlegui, J., Pina, A., Frangou, S., & Papanikolaou, K. (2007). Robotics & constructivism in education: The TERECoP project. In EuroLogo (Vol. 40, pp. 19-24).

Bender, W., Urrea, C., & Zapata-Ros, M. (2015). Pensamiento Computacional [Número monográfico]. *RED, Revista de Educación a Distancia, 46*. Recuperado de http://www.um.es/ead/red/46/

Bers, M. (2008). Blocks to robots: Learning with technology in the early childhood classroom. New York, NY: Teachers College Press.

Bers, M. U. (2012). Designing digital experiences for positive youth development: From playpen to playground. Cary, NC: Oxford

Bers, M. U. (2017). Coding as a playground: Programming and computational thinking in the early childhood classroom. Routledge.

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.

Bers, M., & Horn, M. (2010). Tangible programming in early childhood: Revisiting developmental assumptions through new technologies. In I. R. Berson & M. J.Berson (Eds.), High-tech tots: Childhood in a digital world (pp. 49–70). Greenwich, CT: Information Age Publishing

Bers, M.U. & Resnick, M. (2015). The Official ScratchJr Book. San Francisco, CA: No Starch Press. INTRODUCTION & CHAPTER 1 (skim activities as needed)

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K. (2016). Developing computational thinking in compulsory education – Implications for policy and practice; EUR 28295 EN; doi:10.2791/792158

Burnett, C. (2010). Technology and literacy in early childhood educational settings: A review of research. Journal of Early Childhood Literacy, 10(3), 247-270.

Cejka, E., Rogers, C., & Portsmore, M. (2006). Kindergarten robotics: Using robotics to motivate math,science, and engineering literacy in elementary school. International Journal of Engineering Education, 22(4), 711–722.

Clements, D. H., Sarama, J., Farran, D., Lipsey, M., Hofer, K. G., & Bilbrey, C. (2011). An examination of the Building Blocks math curriculum: Results of a longitudinal scale-up study. Society for Research on Educational Effectiveness. Retrieved from: https://eric.ed.gov/?id=ED518182

Clements, D. H., & Gullo, D. F. (1984). Effects of computer programming on young children's cognition. Journal of Educational Psychology, 76(6), 1051–1058. doi:10.1037/0022-0663.76.6.1051.

Clements, D. H., & Meredith, J. S. (1992). Research on logo: Effects and efficacy. Retrieved from: http://el.media.mit.edu/logo-foundation/pubs/papers/research_logo.html.

Digital Agenda Scoreboard, "Digital Inclusion and Skills", 2014, https://ec.europa.eu/digital-agenda/en/news/scoreboard-2014-digital-inclusion-and-skills-eu-2014

ECDL Foundation (2015). Computing and Digital Literacy - Call for a Holistic Approach. Recuperado de: https://ec.europa.eu/epale/en/resource-centre/content/computing-and-digital-literacy-call-holistic-approach

Eck, J., Hirschmugl-Gaisch, S., Hofmann, A., Kandlhofer, M., Rubenzer, S., & Steinbauer, G. (2013). Innovative concepts in educational robotics: Robotics projects for kindergartens in Austria. In Austrian Robotics Workshop 2013 (Vol. 14, p. 12).

Elkin, M., Sullivan, A., & Bers, M. U. (2014). Implementing a robotics curriculum in an early childhood Montessori classroom. Journal of Information Technology Education: Innovations in Practice, 13, 153-169.

European Schoolnet (2015). Computing our future. Computer programming and coding: priorities, school curricula and initiatives across Europe [Informe técnico]. Recuperado de: http://www.eun.org/c/document_library/get_file?uuid=3596b121-941c-4296-a760-0f4e4795d6fa&groupId=43887

García-Peñalvo F. (2016). TACCLE 3 – Coding. En Proceedings of XVIII Simposio Internacional de Informática Educativa SIIE 2016, pp. 187-189.

Google, Fundación Española para la Ciencia y la Tecnología (FECYT) y everis (2016). Informe "EDUCACIÓN EN CIENCIAS DE LA COMPUTACIÓN EN ESPAÑA 2015". Recuperado de: https://www.fecyt.es/es/publicacion/educacion-de-las-ciencias-de-la-computacion-en-espana

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. Educational Researcher, 42(1), 38-43.

Highfield, K., Mulligan, J., & Hedberg, J. (2008). Early mathematics learning through exploration with programmable toys. In Proceedings of the joint meeting of PME 32 and PME-NA (pp. 169–176).

Hornack M.A. (2011). Technology Integration Matrix.: Recuperado de: http://teche.pbworks.com/f/hornackassignment4-4_22_2011.pdf

Janka, P. (2008). Using a programmable toy at preschool age: why and how. In Teaching with robotics: didactic approaches and experiences. Workshop of International Conference on Simulation, Modeling and Programming Autonomous Robots (pp. 112-121).

Kazakoff, R. E., Sullivan, A., & Bers, U. M. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. Early Childhood Education, 41 , 245–255.

Koehler, M., & Mishra, P. (2009). What is technological pedagogical content knowledge (TPACK)?. *Contemporary issues in technology and teacher education*, *9*(1), 60-70.

Koschmann, T. D. (1996). Paradigm shifts and instructional technology: An introduction. In T. D. Koschmann (Ed.), CSCL: Theory and practice of an emerging paradigm (pp. 1-25). NJ: Lawrence Erlbaum.

Lee, K., Sullivan, A., & Bers, M. U. (2013). Collaboration by design: Using robotics to foster social interaction in Kindergarten. Computers in the Schools, 30(3), 271–281.

Llorens Largo, F., García-Peñalvo, F. J., Molero Prieto, X., & Vendrell Vidal, E. (2017). La enseñanza de la informática, la programación y el pensamiento computacional

en los estudios preuniversitarios. Teoría de la Educación. Educación y Cultura en la Sociedad de la Información, 18(2).

Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014, June). Computational thinking in K-9 education. In Proceedings of the working group reports of the 2014 on innovation & technology in computer science education conference (pp. 1-29). ACM.

Martin G., Toledo G. y Cerverón V. (2002). Fundamentos de Informática y Programación. Recuperado de: http://robotica.uv.es/Libro/Indice.html

Ministerio de Educación, Cultura y Deporte de España (2018). Programación, robótica y pensamiento computacional en el aula. Situación en España. Retrieved from:http://code.educalab.es/wp-content/uploads/2017/09/Pensamiento-Computacional-Fase-1-Informe-sobre-la-situaci%C3%B3n-en-Espa%C3%B1a.pdf

Miranda-Pinto, M. S. (2016). Desafíos de programación y robótica en Educadión Preescolar: proyecto Kids Media Lab. In Tecnología, innovación e investigación en los procesos de enseñanza-aprendizaje (pp. 1848-1855). Octaedro Editorial.

Misirli, A., & Komis, V. (2014). Robotics and programming concepts in early childhood education: A conceptual framework for designing educational scenarios. In Research on e-Learning and ICT in Education (pp. 99-118). Springer New York.

Nacher, V., Garcia-Sanjuan, F., & Jaen, J. (2015). Game Technologies for Kindergarten Instruction: Experiences and Future Challenges. In Proceedings of the 2nd Congreso de la Sociedad Española para las Ciencias del Videojuego (pp. 58-67).

Navarro, J. I., Aguilar, M., Marchena, E., Ruiz, G., Menacho, I., & Van Luit, J. E. (2012). Longitudinal study of low and high achievers in early mathematics. British Journal of Educational Psychology, 82(1), 28-41.

Pane, J. F., & Myers, B. A. (2001). Studying the language and structure in non-programmers' solutions to programming problems. *International Journal of Human-Computer Studies*, *54*(2), 237-264.

Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas. Basic Books, Inc.

Pensamiento Computacional (2017). Fundación Telefónica. Recuperado de: https://www.fundaciontelefonica.com/arte_cultura/publicaciones-listado/pagina-item-publicaciones/itempubli/618/

Portelance, D.J., Strawhacker, A., & Bers, M.U. (2015). Constructing the ScratchJr programming language in the early childhood classroom. International Journal of Technology and Design Education. pp. 1-16. Online First.

Reina M. y Reina S. (2014). INFANTIC/TAC. Proyecto de alfabetización digital de alumn@s, familias y docentes. Recuperado de: http://olmedarein7.wixsite.com/proyectotic

Resnick, M., Martin, F., Berg, R., Borovoy, R., Colella, V., Kramer, K., et al. (1998). Digital manipulatives. In Proceedings of the CHI '98 conference, Los Angeles, April 1998.

Resnick, M. (2007). All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten. In Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition (pp. 1-6). ACM.

Resnick, M. (2017). Lifelong Kindergarten: Cultivating Creativity Through Projects, Passion, Peers, and Play. MIT Press.

Román-González M. (2016). Codigoalfabetización y pensamiento computacional en educación primaria y secundaria: validación de un instrumento y evaluación de programas. Tesis doctoral. UNED.

Sociedad Científica de España (2018). MANIFIESTO SOBRE LA ENSEÑANZA PREUNIVERSITARIA DE LA INFORMÁTICA. Manifiesto elaborado en colaboración con la Conferencia de Directores y Decanos de Ingeniería Informática (CODDI). Disponible en: http://www.scie.es/wp-content/uploads/2018/02/manifiesto-ensenanza-no-universitaria-v2.pdf

Segredo E., Miranda G, León C. (2017). Hacia la educación del futuro: El pensamiento computacional como mecanismo de aprendizaje generativo. Education in the Knowledge Society (EKS). 18(2): 33-58. Disponible en: http://revistas.usal.es/index.php/revistatesi/article/view/16809

Sullivan, A., & Bers, M. U. (2017). Dancing robots: integrating art, music, and robotics in Singapore's early childhood centers. International Journal of Technology and Design Education. DOI 10.1007/s10798-017-9397-0.

Sullivan, A., & Bers, M. U. (2016). Robotics in the early childhood classroom: learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. International Journal of Technology and Design Education, 26(1), 3-20.

Sullivan, A., Elkin, M., & Bers, M. U. (2015). KIBO robot demo: engaging young children in programming and engineering. In Proceedings of the 14th International Conference on Interaction Design and Children (pp. 418-421). ACM.

Van Kleeck, A., & Schuele, C. M. (2010). Historical perspectives on literacy in early childhood. American Journal of Speech-Language Pathology, 19(4), 341-355.

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. Education and Information Technologies, 20(4), 715-728.

Wing, J. (2006). Computational Thinking. Communications of the Acm 49 (3):33-35 (2006)