

Universidad de Zaragoza: Facultad de Educación

Máster en Aprendizaje a lo largo de la vida: Iniciación a la Investigación

**PENSAMIENTO COMPUTACIONAL
EN EDUCACIÓN INFANTIL Y
PRIMARIA: UNA REVISIÓN
SISTEMÁTICA**

Trabajo Fin de Máster

Curso 2020-2021

Autor:

Cristian Ruiz Reinales

Director:

Juan Carlos Bustamante

ÍNDICE

Resumen	7
Abstract	7
1 – Introducción: Justificación de la investigación	9
2 - Marco teórico	13
2.1 Pensamiento Computacional: definición del constructo	13
2.1.1 Definiciones previas: conceptos básicos asociados	13
2.1.2 Pensamiento computacional: evolución conceptual	14
2.1.3 Componentes del Pensamiento Computacional	17
2.2 Evolución de su aplicación en la educación	22
2.2.1 Los sistemas educativos más avanzados en la vanguardia de currículos innovadores	22
2.2.2 Situación curricular en España	23
2.2.3 Producción científica en España	24
3 - Marco metodológico	27
3.1 Concreción del problema de investigación	27
3.2 Objetivos	28
3.2.1 Objetivo General	28
3.2.2 Objetivos específicos	29
3.3 Método	29
3.3.1 La revisión sistemática como técnica rigurosa de investigación documental	29
3.3.2 PRISMA	31
3.3.3 Procedimiento	38
3.3.3.1 Estrategia de búsqueda y fuentes de datos	38
3.3.3.2 Criterios de inclusión	40
3.3.3.3 Criterios de exclusión	41
3.3.3.4 Informe de elegibilidad	42
3.3.3.5 Extracción de datos	43
4 - Resultados	47
4.1 Análisis y clasificación de las diferentes herramientas de evaluación del impacto en el aula	48
4.2 Análisis de las evidencias empíricas de las intervenciones	53
4.3 Análisis de los entornos de aprendizaje (formal/informal)	56

4.5 Análisis de otros constructos relacionados con el pensamiento computacional	59
5 - Discusión	65
6 - Limitaciones	69
7 - Conclusiones e implicaciones prácticas	71
8 - Referencias bibliográficas	73
Anexo I. Artículos seleccionados para la revisión sistemática	83

Resumen

Con el creciente interés en los últimos años del Pensamiento Computacional en educación, es necesario encontrar planteamientos que puedan evaluar las habilidades relacionadas con este constructo. Esta revisión sistemática ha analizado cómo ha sido evaluado en los últimos cinco años, y fueron seleccionados 21 artículos en educación Infantil y Primaria que analizaron las siguientes perspectivas: cómo fue medido el impacto de las intervenciones, evidencias empíricas que aportan, contexto educativo y la relación con otros constructos de interés.

Los resultados indicaron que (a) son necesarias más intervenciones en educación Infantil y contextos informales, (b) para evaluar se apostó por los test, (c) se evaluaron principalmente prácticas computacionales en entornos visuales, (d) escasas experiencias con transferencia de conocimiento del pensamiento computacional. Se detectaron brechas y se han sugerido posibles líneas de investigación en el futuro. Se espera que los datos aportados puedan ser de ayuda a investigadores, diseñadores de currículos educativos y docentes en general.

Palabras clave: Pensamiento Computacional, evaluación, educación primaria, educación infantil, revisión

Abstract

With the growing interest in Computational Thinking in education in recent years, it is necessary to find approaches that can evaluate the skills related to this construct. This systematic review has analyzed how it has been evaluated in the last five years, and 21 articles on Early Childhood and Primary education were selected that analyzed the following perspectives: how the impact of the interventions was measured, the empirical evidence they provide, the educational context and the relationship with other constructs of interest.

The results indicated that (a) more interventions are necessary in Early Childhood education and informal contexts, (b) to evaluate, tests were preferred option, (c) computational practices in visual environments were mainly evaluated, (d) few experiences with knowledge transfer of computational thinking. Gaps were detected and possible lines of investigation in the future have been suggested. It is hoped that the data provided may be of help to researchers, designers of educational curricula and teachers in general.

Keywords: Computational Thinking, assessment, primary education, early childhood education, review

1 – Introducción: Justificación de la investigación

En el momento histórico en el que nos encontramos viviendo, la sociedad y sus diferentes cauces de sistemas de producción demandan profesionales cualificados en nuevas tecnologías como uno de los perfiles más importantes. Dentro de todos estos perfiles que son necesarios cubrir, destaca la alta demanda no cubierta de profesionales como ingenieros de software, desarrolladores de apps, etc, que hayan recibido una formación de calidad en sus correspondientes etapas universitarias especializadas o de Formación Profesional. El sistema educativo, consciente de esta realidad, está comenzando a apostar por propuestas verticales para insertar este tipo de contenidos en los respectivos currículos nacionales (González, 2016), donde uno de los objetivos será forjar estudiantes que no sean simples consumidores de nuevas tecnologías, sino que sean capaces de utilizarlas para crear y ser protagonistas de una sociedad digital (Lambert & Guiffre, 2009; Gardner & Feng, 2010; Clark, Rogers, Spradling y Pais, 2013). Estas propuestas, aunque pudiera pensarse que son novedosas, tienen referentes en las teorías clásicas del aprendizaje como el construccionismo, donde Papert afirmaba que “los niños deben programar la computadora en lugar de ser programados por ella” (Papert, 1980 a través de Blikstein, 2013). Rushkoff (2010) incluso pone en la cúspide del poder comunicativo a aquellos que sepan programar apps y no solo consumirlas, de la misma forma que cuando apareció el lenguaje lo importante no era solo saber escuchar, sino sobre todo saber hablar; fue más importante escribir que leer, más importante manejar la imprenta que leer los periódicos. En este contexto, ser capaz de entender y manipular el lenguaje de los ordenadores se está manifestando como una habilidad ineludible, una nueva forma de alfabetización, que nos habilita para participar de manera plena y efectiva en esta realidad digital que nos rodea, donde debemos no solamente saber cómo utilizar los programas que otros han desarrollado, sino también a aprender a crearlos: “o bien crearás el software, o sino tú serás el software. Es realmente simple, o programas, o serás programado” (Gardner y Davis, 2013).

Estas competencias no deberían ser omitidas en el proceso de enseñanza y aprendizaje y hoy en día deberían tener la misma presencia en las aulas que la lectura, la escritura o las matemáticas (Stošić y Bogdanović, 2013). De hecho, Pea y Kurland (1984) destacan la relación entre las habilidades matemáticas y la programación informática puesto que a la hora de programar se requiere de una continua utilización de la memoria y de una capacidad de concentración que posibilite controlar las diferentes variables que intervienen en un problema.

El pensamiento computacional está relacionado con una forma específica de pensar, de organizar ideas y representaciones. Es una habilidad que fomenta el análisis y la relación de ideas para la organización y la representación lógica de procedimientos (Acevedo, 2017), donde no solo será importante efectuar estudios que permitan diseñar, desarrollar y evaluar la integración de actividades educativas de pensamiento computacional, sino que también se manifiesta como muy importante tener al alcance herramientas validadas que nos permitan evaluar el impacto de nuestras intervenciones, poniendo el foco especialmente en la evaluación de los aspectos cognitivos vinculados con la programación informática y la comprensión de las computadoras (Holmboe, McIver y George, 2001, a través de González, 2006).

A la hora de realizar una intervención educativa, no solamente es prioritario diseñarla adecuadamente para alcanzar unos objetivos específicos establecidos previamente, sino también tener al alcance un método que nos permita poder evaluar el impacto que has conseguido en dicha intervención. Este trabajo de investigación pondrá el foco especialmente en detectar intervenciones educativas donde la inclusión del pensamiento computacional haya tenido un impacto notable y medible en las etapas infantil y primaria, y donde además se puedan establecer relaciones con otros constructos significativos propios de estas etapas educativas. Desde que Jeanette Wing (2006) introdujera el concepto de pensamiento computacional, tal y como veremos en apartados posteriores, han sido muchos los intentos de realizar aproximaciones a las definiciones conceptuales del constructo, establecer relaciones con otros constructos, así como formas diferentes de medir el impacto que estas intervenciones educativas suponen para los estudiantes la adquisición de estas habilidades. Cuando docentes o investigadores consultan sobre la literatura académica sobre un tema en concreto, es necesario la lectura de muchos artículos puesto que la práctica nos demuestra que las conclusiones que pueden ser aportadas desde alguno de ellos, pueden ser rebatidas en otro, y rara vez la lectura de un artículo nos dará un visión definitiva y certera. Por lo tanto, en la búsqueda de la mejor evidencia posible, acudir a una revisión sistemática cuya literatura gire en torno al área de interés puede aportarnos la rigurosidad y especificidad que buscamos. Una revisión sistemática bien diseñada y ejecutada puede aportar mayor evidencia académica y científica, permitiendo realizar una selección acotada de unos artículos de los cuales se espera que puedan clarificar el estado de la cuestión sobre el tema que sea objeto de estudio.

Con todo ello surge la necesidad de realizar una revisión sistemática de la producción científica, donde se analizarán experiencias reales o estudios que tengan como objetivo potenciar el uso del pensamiento computacional en ámbitos formales o informales,

manteniendo actualizado el estado de la cuestión de estas materias, e intentando aportar nuevo conocimiento que pueda impulsar futuras investigaciones enfocadas al ámbito educativo.

2 - Marco teórico

2.1 Pensamiento Computacional: definición del constructo

2.1.1 Definiciones previas: conceptos básicos asociados

En un momento donde vivimos rodeados de lo digital, surgen nuevas formas de inteligencias (inteligencia digital) y la aparición de nuevas habilidades como es la **codigoalfabetización** (*codeliteracy*) (Zapata-Ros, 2015). Según el autor, podemos definir la codigoalfabetización como el proceso de enseñanza-aprendizaje de la lectoescritura con los diferentes lenguajes de programación, donde lo importante no será el lenguaje de programación en sí mismo, sino la capacidad de realizar estas creaciones independientemente del propio lenguaje de programación. Una persona estará códigoalfabetizada cuando es capaz de entender y crear con un lenguaje que los dispositivos programables entiendan. De la misma forma, las personas que desarrollan y evolucionan esta capacidad se dice que piensan computacionalmente, que al fin y al cabo no es más que un proceso cognitivo que nos permite resolver problemas, y que finalmente será expresado de una forma códigoalfabetizada.

González ahonda en la concreción y diferenciación de conceptos importantes en la codigoalfabetización como lo son algoritmo y programa. Un **algoritmo** es una secuencia ordenada de instrucciones u operaciones cuya ejecución en ese correcto orden nos va a dar lugar a la solución deseada para un problema. La construcción de estos algoritmos se produce en nuestra mente tras un espacio de tiempo variable de reflexión personal (Moschovakis, 2001). Para facilitar la creación de estos algoritmos y estandarizar propuestas universales independientemente de los lenguajes usados por los humanos, surgieron herramientas que nos van a ayudar en la construcción visual de estos algoritmos como pueden ser los **diagramas de flujo**: unos símbolos o dibujos que van a representar las operaciones básicas de cualquier algoritmo: secuencia, condición, repetición e iteración (Barrer, 2013).

El paso para convertir un algoritmo en un programa es el arte de codificar. La **codificación** tiene que ver con crear un código fuente en un determinado lenguaje de programación partiendo de un algoritmo previamente creado (González-González, 2019). La codificación nos permite la comunicación entre los humanos que crean esos algoritmos y el lenguaje que entiendan las máquinas. Para que los dispositivos puedan ser programados y realizar las funciones que deseamos que hagan, necesitamos convertir ese algoritmo que

resuelve el problema a un “programa” con un lenguaje que sí pueda ser entendido por esta máquina, para que pueda ser procesado y finalmente ejecutado (Zapata-Ros, 2015). A esta variedad de lenguajes se les denomina lenguajes de programación, de los cuales podemos encontrar cientos de ellos con diferentes propósitos (Chatley, Donaldson y Mycroft 2019). Chatley argumenta que cada lenguaje de programación, al tener sus propias reglas de sintaxis y sus propios conjuntos de instrucciones, evolucionarán en un futuro a otros miles de lenguajes de programación que surgirán a partir de estos.

Bers (2017) argumenta que mientras el pensamiento computacional tiene que ver con habilidades del pensamiento para resolver problemas, la “codificación” se puede ver como una herramienta para enseñar el pensamiento computacional. La codificación es considerada según la Agenda Digital europea como una habilidad clave ya que ayuda a poner en práctica habilidades del siglo XXI tales como la resolución de problemas, el pensamiento analítico y el trabajo en equipo (Bocconi, Chiocciariello, Dettori, Ferrari y Engelhardt, 2016).

Así pues, podemos concluir que cualquier programa estará asociado a un algoritmo que habrá resuelto el problema subyacente, de la misma forma que no todos los algoritmos podrán ser expresados como un programa. Además, es importante resaltar que un mismo programa podrá ser escrito por diferentes personas, con diferentes lenguajes de programación utilizando diferente código. Todas estas definiciones y precisiones terminológicas se espera que contribuyan a una lectura más ágil y precisa de las páginas que siguen a continuación, donde nos adentramos de lleno en las diferentes formas que existen para definir el pensamiento computacional.

2.1.2 Pensamiento computacional: evolución conceptual

La primera referencia formal de pensamiento computacional la podemos encontrar en el artículo de Wing (2006), donde lo define como la habilidad que implica resolver problemas, diseñar sistemas, y entender el comportamiento humano a partir de los conceptos fundamentales de la informática, y que incluye una gama de herramientas mentales que reflejan la amplitud del campo de la informática. La autora argumenta no sólo lo que es el pensamiento computacional, sino también lo que no es:

- a. “Conceptualizing, not programming” (conceptualizar, no programar). _Pensar como un programador de dispositivos va mucho más allá de estar capacitado para programar un

ordenador, ya que ello requiere tener la capacidad de pensar en múltiples capas de abstracción.

- b. “A way that humans, not computers, think” (‘una manera en que los humanos piensan, no las computadoras’). El pensamiento computacional es una habilidad que las personas usamos para resolver problemas, no para simular el pensamiento de un ordenador. Las máquinas están a nuestra disposición y su forma de realizar las cosas es predecible. Sin embargo, las personas, tenemos la capacidad de ser creativos, inteligentes y espontáneos. Por lo tanto, somos los humanos los que creamos estas máquinas para que nos ayuden, utilizando para ello nuestra inteligencia para acometer y resolver problemas que seguramente seríamos incapaces de poder realizar antes de inventar las máquinas.
- c. “Fundamental, not rote skill” (habilidad básica, no puramente mecánica). Se considera una habilidad básica a aquella que cualquier ser humano tiene que poseer para poder desenvolverse en esta sociedad actual.
- d. “Complements and combines mathematical and engineering thinking” (se complementa y se combina con el pensamiento matemático e ingeniero). El pensamiento computacional tiene una relación en su origen con el pensamiento matemático, como el resto de ciencias. De la misma forma, tienen una relación con el pensamiento desarrollado en estudios de ingeniería puesto que lo se crean son construcciones de sistemas informáticos para interactuar con nuestro mundo físico.
- e. “Ideas, not artifacts” (ideas, no artefactos). El pensamiento computacional no sólo está relacionado con las creaciones hardware o software que el ser humano sea capaz de diseñar, sino que también es una habilidad que podemos usar siempre para resolver problemas tan cotidianos como preparar un plato en la cocina o para gestionar mejor nuestra agenda personal.
- f. “For everyone, everywhere” (para cualquiera, en cualquier parte). El pensamiento computacional será una realidad cuando lo tengamos tan integrado en nuestras formas de abordar tareas, que lo más lógico sea que desaparezca como término y filosofía explícitos.

La Doctora Wing actualizará su propia definición argumentando que el pensamiento computacional incluye los procesos de pensamiento implicados en la formulación de problemas

y de sus soluciones, de tal modo que éstos estén representados de una manera que pueda ser abordada efectivamente por un agente-procesador de información (Wing, 2008).

A partir de este momento, se suceden en el tiempo diferentes aportaciones, todas ellas enfocadas a enriquecer los currículos educativos. Fruto del trabajo colaborativo de la “Computer Science Teachers Association” (CSTA, 2011) y la “International Society for Technology in Education” (ISTE) de los Estados Unidos surge su propia aportación: un enfoque para resolver un problema concreto que ayuda a la inclusión de tecnologías digitales con ideas humanas. Todo ello no reemplaza el énfasis en creatividad, razonamiento o pensamiento crítico, pero refuerza esas habilidades al tiempo que realza formas de organizar el problema de manera que el computador pueda ayudar (CSTA & ISTE, 2011).

En 2012 la Royal Society (Reino Unido) crea su primera definición al respecto, donde se argumenta que el pensamiento computacional es el proceso de reconocimiento de los aspectos computables en el mundo que nos rodea, y de aplicar las herramientas y técnicas de las Ciencias de la Computación para comprender y razonar sobre sistemas y procesos, tanto naturales como artificiales (Royal Society, 2012).

Otra aportación interesante ha sido la que realizaron Grover y Pea (2013), quienes proponen los principales conceptos que ellos piensan que han generado el mayor consenso, y que, por lo tanto, deberían estar presentes en cualquier currículo educativo:

- a. Abstracción y generalización de patrones (incluyendo modelos y simulaciones)
- b. Procesamiento sistemático de la información
- c. Sistemas de símbolos y representación
- d. Noción algorítmica de control de flujo
- e. Descomposición estructurada de problemas
- f. Pensamiento iterativo, recursivo y paralelo
- g. Lógica condicional
- h. Limitadores de eficiencia y rendimiento
- i. Depuración y detección sistemática de errores

Kafai y Burke (2014) amplían las definiciones anteriores con un concepto innovador, definiéndolo como un tipo de pensamiento basado en procesos ejecutados por una persona o

una máquina utilizando métodos y modelos que permiten resolver problemas, así como diseñar sistemas que por sí solos no podrían hacerlo.

El equipo de desarrollo de Scratch (Lamb y Johnson, 2011) el software educativo más utilizado en el mundo (Zhang y Nouri, 2019), aportó su visión definiendo el pensamiento computacional como un conjunto de conceptos, prácticas y perspectivas que está fundamentado en el ámbito de la informática. Para ellos, aprender a programar y compartir sus propias creaciones provoca en los estudiantes que se desarrollen como pensadores computacionales, aprendiendo conceptos básicos a la vez que son capaces de desarrollar estrategias de resolución de problemas, diseño y formas de colaboración (ScratchEd Team, 2015). En la misma línea, es visto como una metodología que implementa conceptos básicos de la computación que ayudan a resolver cualquier clase de problemas, forjar estrategias y ejecutar tareas de tal forma que nos permita afrontar los problemas con eficacia y posibilidades de éxito (Olabe, Basogain y Basogain, 2015).

Para finalizar este apartado de definiciones, hilaremos la primera de las definiciones realizada por Wing (2006, 2008) con la postura de Bers (2017), la cual destaca que aunque la resolución de problemas tiene su importancia dentro de la definición más operacional del pensamiento computacional, le otorga especial relevancia al hecho de que el principal potencial es la posibilidad de expresar y crear ideas mientras programamos, argumentando que la programación, al igual que la escritura, es una forma de expresarse. Así, si con el lenguaje somos capaces de concretar múltiples y variadas representaciones, con los lenguajes de programación somos capaces también de expresarnos y crear productos (del Mar Sánchez-Vera, 2019).

2.1.3 Componentes del Pensamiento Computacional

El pensamiento computacional es un término que sirve para aglutinar una serie de habilidades del pensamiento, imprescindibles para cualquier estudiante del siglo XXI (Kong y Abelson, 2019). El Reino Unido, a través de su currículum en Ciencias de la Computación (DFE, U. 2013), detalla que el pensamiento computacional lo forman 6 conceptos (lógica, algoritmos, descomposición, patrones, abstracción, y evaluación sistemática) y 5 aproximaciones (experimentación, creación, depuración, perseverancia y colaboración). Los conceptos explicados serían los siguientes:

- a. Lógica: el razonamiento lógico nos ayuda a explicar por qué algo sucede. Por esta razón, podemos utilizar el razonamiento lógico para determinar exactamente lo que queremos que haga un algoritmo de forma exacta.
- b. Algoritmos: un algoritmo está escrito para ser entendido por humanos y es una secuencia de instrucciones o conjunto de reglas para solucionar un problema.
- c. Descomposición: la capacidad que tenemos para poder fraccionar un problema en partes más pequeñas a través de las cuales nos ayuden a resolver problemas complejos y a gestionar proyectos grandes.
- d. Patrones: estamos rodeados de patrones. Ser capaces de identificar patrones nos permitirá hacer predicciones, crear reglas y resolver problemas más generales
- e. Abstracción: la abstracción tiene que ver con simplificar las cosas; identificando qué es importante sin preocuparnos demasiado por lo anecdótico o irrelevante. La abstracción nos permite manejar la complejidad.
- f. Evaluación sistemática: la evaluación es sistemática y rigurosa; tiene que ver con juzgar la calidad, la efectividad y la eficiencia de las soluciones, sistemas, productos y procesos. La evaluación comprueba que las soluciones aportadas resuelven el problema que nos planteábamos inicialmente no solo de una forma exitosa, sino además de forma más optimizada. Esto tiene mucho que ver con enseñar a nuestros alumnos que el error no tiene que generar frustración, sino que la programación nos da la maravillosa oportunidad de solucionar un error al estar en nuestras manos, en nuestra comprensión.

Y a continuación las aproximaciones:

- a. Experimentación ('Tinkering'): significa probar haciendo. En los más pequeños es la forma más natural de probar los juegos, de una forma espontánea mediante la exploración y el descubrimiento. Para los más mayores tiene que ver más con el concepto de ensayo-error-mejora.
- b. Creación ('Creating'): Programar es un proceso creativo que implica tanto originalidad como la generación de un producto final valioso.
- c. Depuración ('Debugging'): los errores en un algoritmo, programa o código y el proceso de encontrarlos y arreglarlos se denomina 'debugging'. Algunos 'bugs' son errores lógicos, otros son errores sintácticos.

- d. Perseverancia ('Persevering'): programar puede ser una tarea ardua y compleja en la que también vamos a necesitar de nuestra predisposición a perseverar en momentos frustrantes.
- e. Colaboración ('Collaborating'): colaborar significa trabajar con otros para asegurar un mejor resultado. Para la creación de un producto final, en el mundo real no lo crea una sola persona, sino que son varias personas, incluso con diferentes perfiles, las que van creando su propia parte del código para cerrar la solución final. En un caso más sencillo, también se programa por parejas validando la expresión coloquial "cuatro ojos ven más que dos".

Estados Unidos fue otro de los países pioneros en crear un currículo de Computer Science. A través de su Fundación Nacional para la Ciencia de los Estados Unidos ("National Science Foundation") pone en marcha una serie de acciones formativas para la comunidad educativa, como por ejemplo la iniciativa "CS Principles" (Astrachan y Briggs, 2012), cuyo objetivo es fijar y transmitir las bases de las Ciencias de la Computación al alumnado de Bachillerato y primeros años de universidad. Se destacan las siguientes ideas principales relacionadas con procesos cognitivos y operacionales previa a la que fue su primera definición del pensamiento computacional:

- a. Formular problemas de un modo que se haga posible utilizar un ordenador y otras máquinas en su resolución
- b. Automatizar soluciones a través del pensamiento algorítmico (una serie de pasos discretos y ordenados)
- c. Organizar lógicamente y analizar datos
- d. Identificar, analizar e implementar posibles soluciones con el objetivo de lograr la combinación más efectiva y eficiente de pasos y recursos
- e. Representar datos a través de abstracciones tales como modelos y simulaciones
- f. Generalizar y transferir este proceso de solución de problemas a una amplia variedad de situaciones

Otra aproximación distinta es la que realizó el grupo Google for Education (2014), detallando cuales son las 4 fases de procesos cognitivos que trabajan como una rueda circular:

- a. Descomposición de un problema o tarea en pasos discretos

- b. Reconocimiento de patrones (regularidades)
- c. Generalización de dichos patrones y abstracción (descubrir las leyes o principios que causan dichos patrones)
- d. Diseño algorítmico (desarrollar instrucciones precisas para resolver el problema y sus análogos)

En el mismo artículo se argumenta que el pensamiento computacional implica una serie de habilidades, incluyendo:

- a. Formular problemas de una manera que nos permita usar un ordenador y otras herramientas para ayudar a resolverlos
- b. Organizar y analizar de forma lógica un grupo de datos.
- c. Representación de datos a través de abstracciones como modelos y simulaciones
- d. Soluciones automatizadas a través del pensamiento algorítmico (una serie de pasos ordenados)
- e. Identificar, analizar e implementar posibles soluciones con el objetivo de lograr la combinación más eficiente y efectiva de pasos y recursos
- f. Generalizar y transferir este proceso de resolución de problemas a una amplia variedad de problemas

Lo realmente novedoso en su propuesta es que declaran que estas habilidades están respaldadas y mejoradas por una serie de disposiciones o actitudes que incluyen:

- a. Confianza al desenvolverse bien en la complejidad
- b. Persistencia al trabajar con problemas difíciles
- c. Ser más tolerante con respecto a la ambigüedad
- d. Capacidad de hacer frente a problemas abiertos (sin una solución concreta y evidente)
- e. Capacidad de comunicarse y trabajar con otros para llegar a una meta-solución común

Y aún van más allá asegurando que los conceptos de pensamiento computacional no son solo los procesos mentales (por ejemplo, abstracción, diseño de algoritmos, descomposición, reconocimiento de patrones, etc.) sino que también producen resultados tangibles (por ejemplo, automatización, representación de datos, generalización de patrones, etc.) asociados con la resolución de problemas en computación. Estos se definen de la siguiente manera:

- a. Abstracción: identificación y extracción de información relevante para definir las ideas principales
- b. Diseño de algoritmos: creación de una serie ordenada de instrucciones para resolver problemas similares o para realizar una tarea
- c. Automatización: Tener computadoras o máquinas para hacer tareas repetitivas.
- d. Análisis de datos: dar sentido a los datos mediante la búsqueda de patrones o el desarrollo de ideas
- e. Recopilación de datos: Recopilación de información
- f. Representación de datos: representación y organización de datos en gráficos, cuadros, palabras o imágenes apropiados
- g. Descomposición: desglosar datos, procesos o problemas en partes más pequeñas y manejables
- h. Paralelización: procesamiento simultáneo de tareas más pequeñas de una tarea más grande para alcanzar de manera más eficiente un objetivo común
- i. Generalización de patrones: creación de modelos, reglas, principios o teorías de patrones observados para probar los resultados previstos
- j. Reconocimiento de patrones: observación de patrones, tendencias y regularidades en los datos
- k. Simulación: Desarrollando un modelo para imitar procesos del mundo real

Bers (2017) amplía los elementos relacionados con el pensamiento computacional, concretándolo como un proceso expresivo organizado en siete habilidades de pensamiento computacional: algoritmos, modularidad, estructuras de control, representación, hardware / software, el proceso de diseño, y la depuración.

Para finalizar, partiendo de la propuesta original de Wing (2006), Olabe, Basogain y Basogain (2015) resumen las principales habilidades del pensamiento asociados al pensamiento computacional:

- a. Reformular un problema a uno parecido que sepamos resolver por reducción, encuadrarlo, transformar, simular
- b. Pensar recursivamente

- c. Procesar en paralelo
- d. Interpretar código como datos y datos como código
- e. Generalizar análisis dimensional
- f. Reconocer ventajas y desventajas del solapamiento
- g. Reconocer coste y potencia de tratamiento indirecto y llamada a proceso
- h. Juzgar un programa por simplicidad de diseño
- i. Utilizar abstracción y descomposición en un problema complejo o diseño de sistemas complejos
- j. Elegir una correcta representación o modelo para hacer tratable el problema
- k. Seguridad en utilizarlo, modificarlo en un problema complejo sin conocer cada detalle
- l. Modularizar ante múltiples usuarios
- m. Prevención, protección, recuperarse de un escenario de peor caso posible
- n. Utilizar razonamiento heurístico para encontrar la solución
- o. Planificar y aprender en presencia de incertidumbre
- p. Buscar, buscar y buscar más
- q. Utilizar muchos datos para acelerar la computación
- r. Límite tiempo/espacio y memoria/potencia de procesado

Este apartado ha servido como base teórica para comprender los procesos de pensamiento computacional con respecto a la resolución de problemas y procesos cognitivos asociados. Dada esta realidad, no es sorprendente que haya aparecido un interés en muchos países por introducir el pensamiento computacional como un conjunto de habilidades de resolución de problemas que los nuevos estudiantes deberían adquirir.

2.2 Evolución de su aplicación en la educación

2.2.1 Los sistemas educativos más avanzados en la vanguardia de currículos innovadores

El pensamiento computacional es considerado como el núcleo de las asignaturas STEM (Science, Technology, Engineering, & Mathematics) (Kong y Abelson, 2019). Uno de los gobiernos pioneros en elevar las asignaturas “Computer Science” a la altura del resto de

ciencias fue el gobierno británico (Crick y Sentance, 2011), el cual introdujo un novedoso currículo vertical en el año 2013, con un gran programa de formación para los profesores de todo el país, y con un plataforma donde los docentes del país están al tanto de las novedades de este currículo así como también tienen un espacio virtual donde comparten materiales para todas las edades y de todas las materias.

En estos momentos hasta 13 países europeos ya han integrado la programación en sus respectivos currículos, si bien es cierto que con diferentes enfoques y en diversos niveles (Bocconi et al., 2016). Fuera de nuestras fronteras europeas, el gobierno federal de Estados Unidos, también ha lanzado un programa denominado “Computer Science for all”, en el cual aspiran a que todos los estudiantes de este país, en todas las etapas obligatorias educativas, aprendan Computer Science y de esta forma puedan adquirir las habilidades relacionados con el PC para poder ser creadores en mundo digitalmente globalizado (Smith, 2016). Incluso ciudades como San Francisco, Chicago o New York han anunciado programas para proporcionar el acceso a la programación para todos los alumnos en edad escolar.

También merece la pena comentar especialmente programas un poco más exóticos para nuestras latitudes como pueden ser un programa que es tendencia mundial como lo es el “Code For Fun” de Singapur (Gal, 2019) o el nuevo plan de estudios en Nigeria (Hsu, Irie y Ching, 2019), donde el pensamiento computacional figura dentro de las asignaturas con un valor importante.

2.2.2 Situación curricular en España

El INTEF (Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado) ha realizado junto con expertos de diferentes comunidades, un marco de trabajo denominado “*Ponencia de pensamiento computacional y robótica educativa*” (MEFP, I. CNIIE, 2018), la cual finalizó en Octubre de 2018 con una propuesta normativa concreta al Ministerio de Educación, y donde podemos encontrar un análisis sobre la actual Ley de Educación, la cual no menciona el pensamiento computacional en las etapas de Infantil y Primaria, haciendo sólo referencias a uso de las TIC en varias asignaturas tanto troncales (Ciencias de la naturaleza, Ciencias sociales, Lengua castellana y literatura, Matemáticas y Primera lengua extranjera) como específicas (Educación artística, Educación física y Valores sociales y cívicos). También podemos encontrar una reflexión preocupante, donde aseguran que, aunque algunas Comunidades Autónomas han incluido asignaturas de programación, son

de libre elección, por lo que un alumno puede terminar la educación secundaria obligatoria sin haber cursado esta materia.

Es el mismo caso de asignaturas optativas existentes en la ley actual donde se puede encontrar al pensamiento computacional tal como establece el Real Decreto 1105/2014, de 26 de diciembre de 2015 (Estado, B. O., 2015)., tanto en Tecnología 4º ESO, como en Tecnologías de la Información y de la Comunicación, una asignatura específica-optativa tanto para la opción de enseñanzas académicas como para la opción de enseñanzas aplicadas. En Bachillerato existen dos asignaturas específicas-optativas, Tecnología industrial y Tecnologías de la Información y la Comunicación, tanto en 1º como en 2º curso, que incluyen contenidos de estas habilidades.

La Comunidad de Madrid es la que destaca por encima de todas las demás. Tal como se recoge en el Decreto 48/2015, de 14 de mayo del Consejo de Gobierno, por el que se establece para la Comunidad de Madrid el currículo de la Educación Secundaria Obligatoria del Estado (de Madrid, C., 2015). se creó la asignatura de configuración autonómica “Tecnología, Programación y Robótica”, obligatoria para 1º, 2º y 3º de ESO. La materia se articula en torno a cinco ejes: Programación y pensamiento computacional, Robótica y la conexión con el mundo real, Tecnología y el desarrollo del aprendizaje basado en proyectos, Internet y su uso seguro y responsable, y Técnicas de diseño e impresión 3D.

2.2.3 Producción científica en España

La bibliografía sobre artículos de investigación respecto al pensamiento computacional en nuestro país es escasa, más aún si nos ceñimos a cuestiones clave de esta investigación cómo son la evaluación de las intervenciones prestando especialmente atención al “qué” se está evaluando, así como al “cómo”.

Marcos Román González es el autor más prolífico, habiendo realizado una tesis doctoral sobre códigoalfabetización y pensamiento computacional en educación primaria y secundaria, donde se realiza una evaluación de programa y se detalla la validación de un instrumento de medición del pensamiento computacional (González, 2016). Partiendo de los resultados de su tesis doctoral, posteriormente publica varios artículos relacionados. Uno de ellos es un artículo cuya meta principal es divulgar cómo su Computational Thinking Test (CTt) ha sido diseñado y desarrollado, así como sus estadísticas descriptivas y confiabilidad derivadas de una medición realizada sobre una muestra superior a mil estudiantes españoles.

Otro de sus objetivos es investigar sobre la correlación entre otras capacidades cognitivas y el pensamiento computacional (Román-González, Pérez-González y Jiménez-Fernández, 2017).

Román-González publica más artículos junto con Jesús Moreno, Juan Carlos Pérez-González, Gregorio Robles y Juan Daniel Rodríguez. Fruto de esta colaboración se desarrolló una herramienta para evaluar proyectos realizados con Scratch en modo portfolio (Moreno-León, Robles y Román-González, 2015) que posteriormente fue validada como software óptimo para medir la complejidad (Moreno-León, Robles y Román-González, 2016, Abril) y donde se correlacionó positivamente su corrección automática de proyectos hechos con Scratch con la corrección hecha por humanos (Moreno-León, Román-González, Hartevelt, y Robles, 2017, Mayo); otro artículo donde se intentó indagar en qué etapa educativa podría ser la mejor para incluir el aprendizaje a la programación (Moreno-León, Robles, y Román-González, 2016, Junio); un artículo orientado a analizar la importancia de aportar evidencia empírica para la introducción del pensamiento computacional (Moreno-León, Robles, y Román-González, 2017); una comparativa entre diferentes herramientas mencionadas en este trabajo para medir el pensamiento computacional desde diferentes perspectivas (Moreno-León, Robles y Román-González, 2017, Julio); un artículo donde se realiza un análisis de la importancia de la inclusión de actividades unplugged (sin tecnología) en el currículo educativo junto a más autores (Brackmann et al., 2017, Noviembre); un artículo donde se concluye que el futuro de la investigación podría estar en expandir esta red nomológica explorando cómo es el pensamiento computacional relacionado con otras variables cognitivas y computacionales, tales como memoria de trabajo, funciones ejecutivas o programación específica (Román-González, Pérez-González, Moreno-León y Robles, 2018); un artículo donde destacan el pensamiento computacional como un skill universal (Moreno-León, Román-González y Robles, 2018, Abril); un artículo donde el debate se centra sobre si el talento computacional puede ser detectado (Román-González, Pérez-González, Moreno-León y Robles, 2018, Junio) y artículos donde se empieza a indagar en aspectos como el aprendizaje del pensamiento computacional a través de machine learning (García, León, Román-González y Robles, 2019, Noviembre) y el desarrollo de una herramienta para introducir la inteligencia artificial en las aulas para trabajar estas habilidades del pensamiento (García, Moreno-León, Román-González y Robles, 2020).

Otro de los autores más prolíficos es Miguel Zapata-Ros, centrado en artículos que profundizan en los conceptos más teóricos del pensamiento computacional como su relación con el pensamiento bayasano, la probabilidad y el “data science” (Zapata-Ros, 2020, Noviembre), aquellos en los que habla del fenómeno que supone en nuestras aulas en cuanto a

la introducción de estas materias llegando a considerar como una competencia clave (Zapata-Ros, 2015; Zapata-Ros, 2018; Zapata-Ros, 2020, Diciembre), la importancia de utilizar actividades unplugged (sin tecnología) para enseñar estos conceptos en la etapa de Infantil (Zapata-Ros, 2019) e incluso estableciendo relación entre uno de los conceptos clave del pensamiento computacional como son los patrones con el aprendizaje de lenguas (Pérez-Paredes y Zapata-Ros, 2018).

Juan Manuel Doderó, experto en computación, realidad aumentada y lenguajes de programación, también ha publicado literatura relacionada con el pensamiento computacional: un análisis de la relación entre el pensamiento computacional y el concepto de abstracción matemática en educación primaria (Chaabi, Azmani y Doderó, 2019, Octubre) y una propuesta de formación a profesores en pensamiento computacional para el desarrollo de aplicaciones móviles en el curriculum (Doderó, Mota y Ruiz-Rube, 2017, Octubre).

También podemos encontrar literatura relacionada con pensamiento computacional a Xabier Basogain Olabe, que entre sus numerosas publicaciones encontramos algunas destinadas a crear materiales o didácticas sobre la introducción del pensamiento computacional en educación (Olabe et al., 2015; Rico y Olabe, 2018, Enero; Lugo y Olabe, 2018, Febrero; Lugo, Olabe y Niño, 2018, Marzo; Basogain, Olabe y Olabe y Parco, 2020).

3 - Marco metodológico

3.1 Concreción del problema de investigación

El pensamiento computacional está emergiendo como una de las competencias clave del siglo XXI (Voogt, Fisser, Good, Mishra y Yadav, 2015). La adquisición de esta competencia requiere de un aprendizaje profundo en procesos cognitivos relacionados con las habilidades del pensamiento (Labusch & Eickelmann, 2017), los cuales figuran en varios marcos de aprendizaje (Kong et al., 2017, Julio). Las teorías de resolución de problemas también son tenidas en cuenta para obtener un análisis más preciso de estos procesos puesto que investigaciones previas sugieren una alta relación entre estos procesos cognitivos y el pensamiento computacional (Yadav, Gretter, Good y McLean, 2017; Wing, 2008).

Conocer los elementos que componen el pensamiento computacional y sus habilidades asociadas que un estudiante debe poseer son indispensables para que puedan ser medidas y evaluadas, convirtiendo este proceso en todo un desafío (Brennan y Resnick, 2012). El papel de la educación y especialmente de las escuelas también es un tema de discusión en este contexto puesto que plantea nuevas tareas y desafíos para ellas (Fraillon, Ainley, Schulz, Friedman y Gebhardt, 2019). Algunos expertos ya han desarrollado marcos educativos (Brennan y Resnick, 2012; Kong et al., 2017) para fomentar la capacidad de los estudiantes de ver el mundo a través de una lente computacional (Kong et al., 2017, p. 84).

Grover y Pea (2013) enfatizan la necesidad de que todos los estudiantes deberían aprender el pensamiento computacional, incluyendo aquellos que manifiestan no tener interés en estas áreas. De hecho, todos deberían tener la oportunidad de adquirir estas competencias en el campo del pensamiento computacional, que les permitan participar con éxito en un mundo digitalizado (Wing, 2017). Esto básicamente confirma el argumento presentado por Wing (2006), quien afirma que el pensamiento computacional es de vital importancia para todas las ciencias, no solo para la informática (Yadav et al., 2017). Desde esta perspectiva, se propone que la comprensión del pensamiento computacional y sus procesos cognitivos asociados ayudaría a los profesores a integrar cualquier concepto relativo al ámbito de la enseñanza (Labusch y Eickelmann, 2017).

Es de interés prioritario en este trabajo conocer propuestas de inclusión de pensamiento computacional en etapas de infantil y primaria, centrando exclusivamente el proceso de investigación en aquellas experiencias que nos muestren evidencias en cuanto a la evaluación

de la intervención, donde será importante no solo la forma en la que se puede evaluar el pensamiento computacional, sino también cómo se está evaluando y el qué se está evaluando. De esta forma, podremos realizar relaciones con otros constructos de interés y conocer el impacto que estas intervenciones tienen en la educación de nuestros estudiantes.

Para concretar ese problema es necesario plantearse una serie de cuestiones a las que se les quiere ofrecer una respuesta, y que, de paso, nos servirán para delimitar el ámbito de acción de nuestra búsqueda en la investigación. Tal y como se ha avanzado conceptualmente en la justificación y marco teórico de este trabajo, la principal pregunta de investigación de este trabajo sería ¿cómo se está evaluando el pensamiento computacional en las etapas de educación infantil y primaria?

No obstante, existen otras preguntas que aportarán una visión más global a la hora de dar respuesta al tema en cuestión.

1. ¿Qué herramientas existen para evaluar el pensamiento computacional?
2. ¿Se está evaluando el pensamiento computacional en las aulas?
3. ¿A qué etapas educativas van dirigidas?
4. ¿Están validadas científicamente?
5. ¿En qué ambientes de aprendizaje se desarrollan?
6. ¿Qué se está evaluando? ¿Cómo se está evaluando?
7. ¿Existen evidencias de relación entre el pensamiento computacional y otros constructos de interés en estas etapas educativas?

3.2 Objetivos

3.2.1 Objetivo General

El objetivo general de este trabajo será indagar sobre el lugar que ocupa en el contexto educativo el pensamiento computacional así como comprobar cómo se está evaluando su inclusión en el aula. Para conseguir este objetivo, se realizará una revisión sistemática a través del análisis de casos, estudios y experiencias realizadas en el ámbito educativo en las etapas de Infantil y Primaria.

3.2.2 Objetivos específicos

Una revisión sistemática puede brindar información sobre el diseño e implantación de intervenciones educativas efectivas relativas al ámbito del pensamiento computacional, así como mejorar y diseñar herramientas de evaluación efectivas. En esta revisión se pretende alcanzar los siguientes objetivos específicos:

1. Clasificación de los diferentes métodos de evaluación de la inclusión del pensamiento computacional en las aulas
2. Comprobar la evidencia empírica de las intervenciones educativas seleccionadas para revisión
3. Analizar los contextos de aprendizaje en los que se producen las intervenciones detalladas en los artículos seleccionados
4. Analizar cuáles son las principales habilidades del pensamiento y capacidades cognitivas relacionadas con el pensamiento computacional que sean objeto de estudio
5. Comprobar si existe relación con otros constructos de interés

3.3 Método

3.3.1 La revisión sistemática como técnica rigurosa de investigación documental

Las revisiones sistemáticas son documentos orientados a seleccionar una muestra de la evidencia disponible respecto a un tema de investigación en particular, basándose en estudios primarios que han tenido en cuenta revisiones cuantitativas y cualitativas. Entre sus principales ventajas podemos encontrar que nos puede ofrecer de forma rápida, sencilla y concisa una información para estar al día de un tema en concreto, convirtiéndose en un estudio de estudios, recopilando toda la información alrededor de ese tema (Manterola, Astudillo, Arias, Claros y MINCIR, 2013).

Las razones que nos pueden llevar a realizar una revisión sistemática pueden ser variadas:

- a. conocer en profundidad el efecto que han producido una serie de intervenciones educativas
- b. cuando se han detectado cierta incertidumbre sobre las evidencias de alguna intervención educativa que puedan ser contrarias respecto a una utilidad real

- c. cuando el propósito es analizar dentro de una intervención el comportamiento de subgrupos de sujetos

De todo lo anterior se deduce que la motivación para realizar o leer revisiones sistemáticas son puramente prácticas. Ante la gran cantidad de artículos que se generan día a día, sería imposible mantenerse al día en nuestras respectivas áreas de investigación, y una revisión sistemática nos puede ayudar a encontrar una selección de los mejores artículos que puedan aportar rigor científico e intervenciones de calidad.

Según Manteriola (2013), la principal fortaleza de las revisiones sistemáticas es que se tratan de diseños de investigación eficientes, que nos permitirán elevar la precisión de la estimación y la validez y generalización de los resultados obtenidos. Por el contrario, su principal debilidad radica en la posible inclusión de artículos de baja calidad metodológica que den lugar a resultados que no se correspondan con la realidad y que estos estudios tienen que ser realizados por profesionales que posean los conocimientos prácticos, así como experiencia en métodos de búsqueda y revisión de artículos científicos, así como en la interpretación de los resultados que puedan ser obtenidos (Bellomo y Bagshaw, 2006).

Será fundamental prestar especial atención a los posibles sesgos a los que puede verse afectado una revisión sistemática, destacando los siguientes:

- a. sesgo de selección, en cuanto a los diferentes grupos de estudiantes que están incluidos en las investigaciones seleccionadas
- b. sesgo de publicación, puesto que a veces podemos encontrarnos con artículos que por alguna razón no están indexados en bases de datos y pueden aportar información relevante para nuestra investigación
- c. sesgo del observador, que puede tener tendencia a favorecer o desfavorecer autores para él conocidos

Las revisiones sistemáticas han sido usadas en investigación médica desde hace más de 30 años. En investigación educativa, hay varias razones por las que consideran un aspecto clave (Bennett, Lubben, Hogarth y Campbell, 2005):

- a. existe un interés creciente desde los ámbitos de decisiones políticas donde la toma de decisiones está vinculada a la evidencia en una serie de áreas, no solo en la educación. Para los responsables de tomar estas decisiones, apoyarse en revisiones sistemáticas de

la literatura de investigación aporta el potencial de presentar pruebas que avalen determinadas propuestas

- b. existe un impulso hacia estrechar los lazos entre la política, la práctica y la investigación. Especialmente en aquellos resultados de investigación procedentes de prácticas del aula a los que se denomina “basados en evidencia” o “práctica enriquecida con evidencias”

El objetivo de este trabajo de investigación es aprovechar todas estas cualidades que nos ofrece la posibilidad de realizar una revisión sistemática como diseño de investigación eficiente, que nos permita interpretar y evaluar todo el conocimiento científico en cuanto a la evaluación del pensamiento computacional, centrada en los artículos científicos producidos en el periodo del 2016 al 2020 (ambos inclusive) de tal forma que permita reforzar el conocimiento previo a estas fechas, así como mantener de forma actualizada el estado de la cuestión en cuanto a método de evaluación del pensamiento computacional, centrado en las etapas de infantil y primaria. Se espera que este nuevo conocimiento que se genere en este estudio pueda servir a la comunidad educativa y científica para comenzar y/o mejorar en sus investigaciones sobre la inclusión del pensamiento computacional de una manera curricular.

3.3.2 PRISMA

Las revisiones sistemáticas tienen que ser valoradas en cuanto a su validez interna, la magnitud de los resultados que puedan ofrecer y también su validez externa (Manterola et al., 2013). La falta de calidad en los metaanálisis llevó a un grupo de investigación internacional a crear una nueva guía sobre cómo hacer realizar este tipo de trabajos: la Declaración QUOROM (Moher et al., 2000), cuyo principal objetivo estaba enfocado a establecer un conjunto de normas que ayudaran a mejorar la calidad de la presentación de los metaanálisis de ensayos clínicos aleatorizados, y que estaría compuesta por una lista de comprobación estructuras de 18 ítems junto con un diagrama de flujo que describe todo el proceso, y que deberían de ser seguidas por los autores de metaanálisis así como editores de revistas (Urrútia, G., & Bonfill, X., 2010). Años después, tras seguir encontrando deficiencias en este método y para abrir también la posibilidad de poder aplicarlo en revisiones sistemáticas, se desarrolla la propuesta “Preferred Reporting Items for Systematic Reviews and Meta Analyses—PRISMA”, con el objetivo fundamental de tratar avances conceptuales y prácticas en la rigurosidad necesaria

para realizar revisiones sistemáticas (Moher, Liberati, Tetzlaff, Altman y PRISMA, 2009). Estos aspectos conceptuales son los siguientes:

- a. La realización de una revisión sistemática es un proceso interactivo donde los revisores pueden necesitar modificar sus protocolos originales de revisión durante su realización.
- b. Evaluar el riesgo de sesgo a nivel de estudio frente a los resultados, que implica evaluar la fiabilidad y validez de los datos
- c. Conducir y publicar una investigación son conceptos diferentes
- d. Los sesgos relacionados con la publicación son importantes ya que pueden dificultar la realización y la interpretación en dichas revisiones sistemáticas.

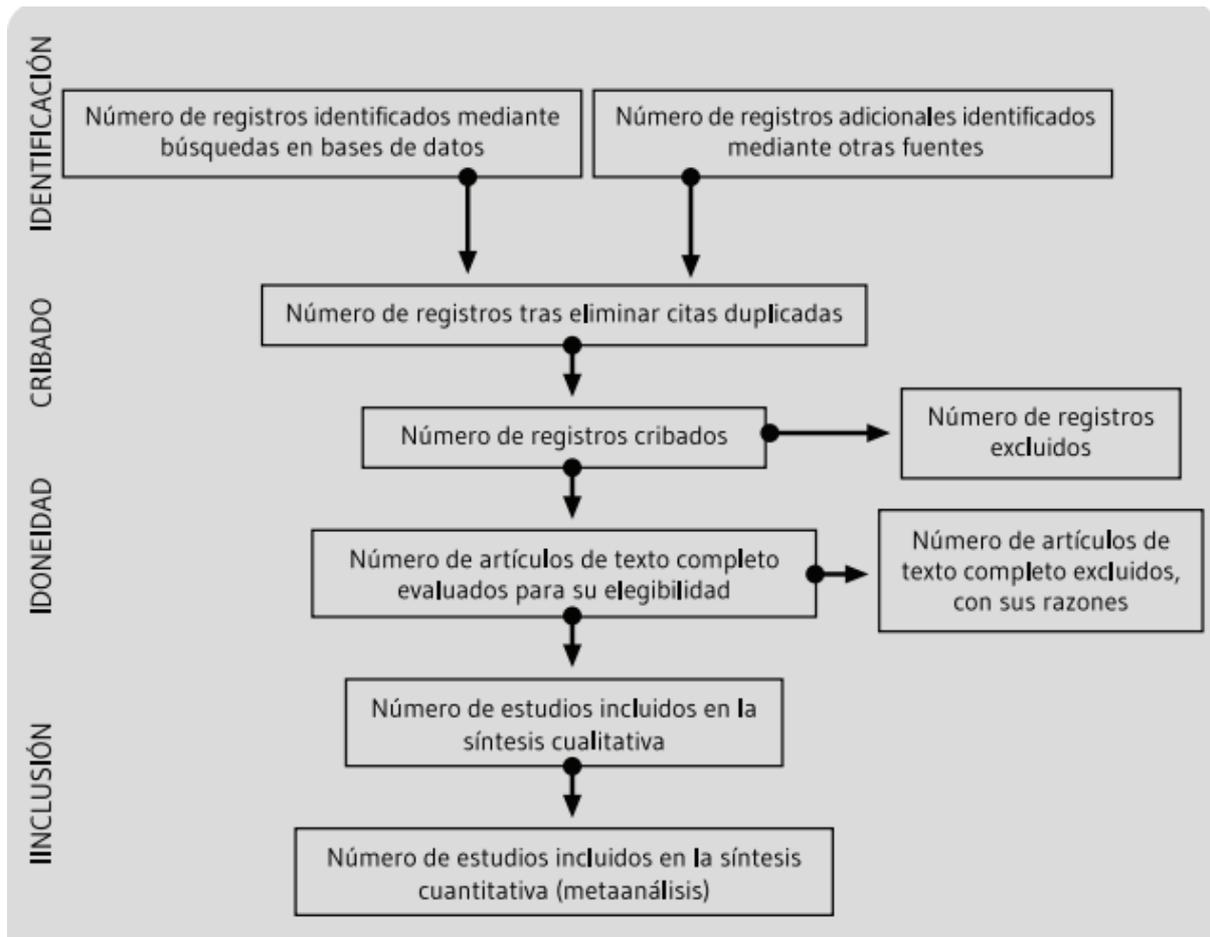
Tal y como podemos extraer de Moher, Liberati, Tetzlaff, Altman y PRISMA, G., (2014), la Declaración PRISMA consiste en una lista de comprobación de 27 ítems (ver Tabla 1) y un diagrama de flujo de cuatro fases (ver Figura 1)

Tabla 1. Lista de comprobación de 27 elementos traducida al español (Moher et al., 2014)

Sección/tema	#	ítem
TÍTULO		
Título	1	Identificar la publicación como revisión sistemática, metaanálisis o ambos.
RESUMEN		
Resumen estructurado	2	Facilitar un resumen estructurado que incluya, según corresponda: antecedentes; objetivos; fuente de los datos; criterios de elegibilidad de los estudios, participantes e intervenciones; evaluación de los estudios y métodos de síntesis; resultados; limitaciones; conclusiones e implicaciones de los hallazgos principales; número de registro de la revisión sistemática.
INTRODUCCIÓN		
Justificación	3	Describir la justificación de la revisión en el contexto de lo que ya se conoce sobre el tema
Objetivos	4	Plantear de forma explícita las preguntas que se desea contestar en relación con los participantes, las intervenciones, las comparaciones, los resultados y el diseño de los estudios (PICOS).
MÉTODOS		
Protocolo y registro	5	Indicar si existe un protocolo de revisión al que se pueda acceder (por ejemplo, dirección web) y, si está disponible, la información sobre el registro, incluyendo su número de registro.
Criterios de elegibilidad	6	Especificar las características de los estudios (por ejemplo: PICOS, duración del seguimiento) y de las características (por ejemplo: años abarcados, idiomas o estatus de publicación) utilizadas como criterios de elegibilidad y su justificación.
Fuente de información	7	Describir todas las fuentes de información (por ejemplo: bases de datos y períodos de búsqueda, contacto con los autores para identificar estudios adicionales, etc.) en la búsqueda y la fecha de la última búsqueda realizada.
Búsqueda	8	Presentar la estrategia completa de búsqueda electrónica en, al menos, una base de datos, incluyendo los límites utilizados de tal forma que pueda ser reproducible.
Selección de los estudios	9	Especificar el proceso de selección de los estudios (por ejemplo: el cribado y la elegibilidad incluidos en la revisión sistemática y, cuando sea pertinente, incluidos en el metaanálisis).

Proceso de recopilación de datos	10	Describir los métodos para la extracción de datos de las publicaciones (por ejemplo: formularios dirigidos, por duplicado y de forma independiente) y cualquier proceso para obtener y confirmar datos por parte de los investigadores
Lista de datos	11	Listar y definir todas las variables para las que se buscaron datos (por ejemplo, PICOS fuente de financiación) y cualquier asunción y simplificación que se hayan hecho
Riesgo de sesgo en los estudios individuales	12	Describir los métodos utilizados para evaluar el riesgo de sesgo en los estudios individuales (especificar si se realizó al nivel de los estudios o de los resultados) y cómo esta información se ha utilizado en la síntesis de datos.
Medida de resumen	13	Especificar las principales medidas de resumen (por ejemplo: razón de riesgos o diferencia de medias).
Síntesis de resultados	14	Describir los métodos para manejar los datos y combinar resultados de los estudios, si se hiciera, incluyendo medidas de consistencia (por ejemplo, cuantificación de la heterogeneidad mediante el índice estadístico para cada metaanálisis.
Riesgo de sesgo entre los estudios	15	Especificar cualquier evaluación del riesgo de sesgo que pueda afectar la evidencia acumulativa (por ejemplo, sesgo de publicación o comunicación selectiva).
Análisis adicionales	16	Describir los métodos adicionales de análisis (por ejemplo: análisis de sensibilidad o de subgrupos, metarregresión), si se hiciera, indicar cuáles fueron preespecificados.
RESULTADOS		
Selección de estudios	17	Facilitar el número de estudios cribados, evaluados para su elegibilidad e incluidos en la revisión, y detallar las razones para su exclusión en cada etapa, idealmente mediante un diagrama de flujo
Características de los estudios	18	Para cada estudio, presentar las características para las que se extrajeron los datos (por ejemplo: tamaño, PICOS y duración del seguimiento) y proporcionar las citas bibliográficas
Riesgo de sesgo en los estudios individuales	19	Presentar datos sobre el riesgo de sesgo en cada estudio y, si está disponible, cualquier evaluación del sesgo en los resultados (ver ítem 12).
Resultados de los estudios individuales	20	Para cada resultado considerado para cada estudio (beneficios o daños), presentar: a) el dato resumen para cada grupo de intervención y b) la estimación del efecto con su intervalo de confianza, idealmente de forma gráfica mediante un diagrama de bosque (forest plot).
Síntesis de los resultados	21	Presentar resultados de todos los metaanálisis realizados, incluyendo los intervalos de confianza y las medidas de consistencia.
Riesgo de sesgo entre los estudios	22	Presentar los resultados de cualquier evaluación del riesgo de sesgo entre los estudios (ver ítem 15).
Análisis adicionales	23	Facilitar los resultados de cualquier análisis adicional, en el caso de que se hayan realizado (por ejemplo, análisis de sensibilidad o de subgrupos, metarregresión (ver ítem 16)).
DISCUSIÓN		
Resumen de la evidencia	24	Resumir los hallazgos principales, incluyendo la fortaleza de las evidencias para cada resultado principal; considerar su relevancia para grupos clave (por ejemplo: proveedores de cuidados, usuarios y decisores en salud).
Limitaciones	25	Discutir las limitaciones de los estudios y de los resultados (por ejemplo, riesgo de sesgo) y de la revisión (por ejemplo: obtención incompleta de los estudios identificados o comunicación selectiva).
Conclusiones	26	Proporcionar una interpretación general de los resultados en el contexto de otras evidencias así como las implicaciones para la futura investigación.
FINANCIACIÓN		
Financiación	27	Describir las fuentes de financiación de la revisión sistemática y otro tipo de apoyos (por ejemplo, aporte de los datos), así como el rol de los financiadores en la revisión sistemática.

Figura 1. Flujo de información a través de las diferentes fases de una revisión sistemática traducida al español (Moher et al., 2014).



Las siguientes son las etapas definidas por PRISMA para la realización de una revisión sistemática (Moher et al., 2009):

1. Formulación del problema. Se comienza por la identificación del problema y se define una pregunta de investigación que nos ayude a acotar el problema que se desea abordar.
2. Localización y selección de estudios primarios. En primer lugar, habrá que concretar cuáles serán nuestros criterios de selección de artículos, seleccionando palabras clave apoyándonos de operadores booleanos para un mejor filtro. Finalmente se aplican estas palabras clave en las bases de datos seleccionadas. Eventualmente también se incluirán artículos no procedentes de estas bases de datos, que puedan aportar información relevante.
3. Evaluación de la calidad de los estudios en cuanto a la metodología aplicada, prestando especial atención a la validez interna, así como a los posibles sesgos que se puedan encontrar.

4. Extracción de datos, mediante el cual se detalla el proceso seguido para seleccionar los artículos, siguiendo unos criterios de inclusión y donde se han eliminado otros en base a unos criterios de exclusión.
5. Análisis y preparación de los resultados. Estos pueden realizarse desde un punto de vista cualitativo, así como cuantitativo, por ejemplo, realizando un metaanálisis.
6. Presentación de los resultados, donde la presentación de un diagrama de flujo del proceso de la selección de artículos es fundamental, así como la inclusión de gráficas que puedan aportar información visual y relevante de estos resultados, para una comprensión más intuitiva.

En septiembre de 2018 se inicia el proceso para actualizar a la versión actual de PRISMA 2020, donde finalmente se actualizan 15 de los elementos de la lista de comprobación original, se mantienen 7 de ellos y se incorporan otros nuevos (ver Tabla 2) y también se modifica la plantilla para el diagrama de flujo de PRISMA (ver Figura 2).

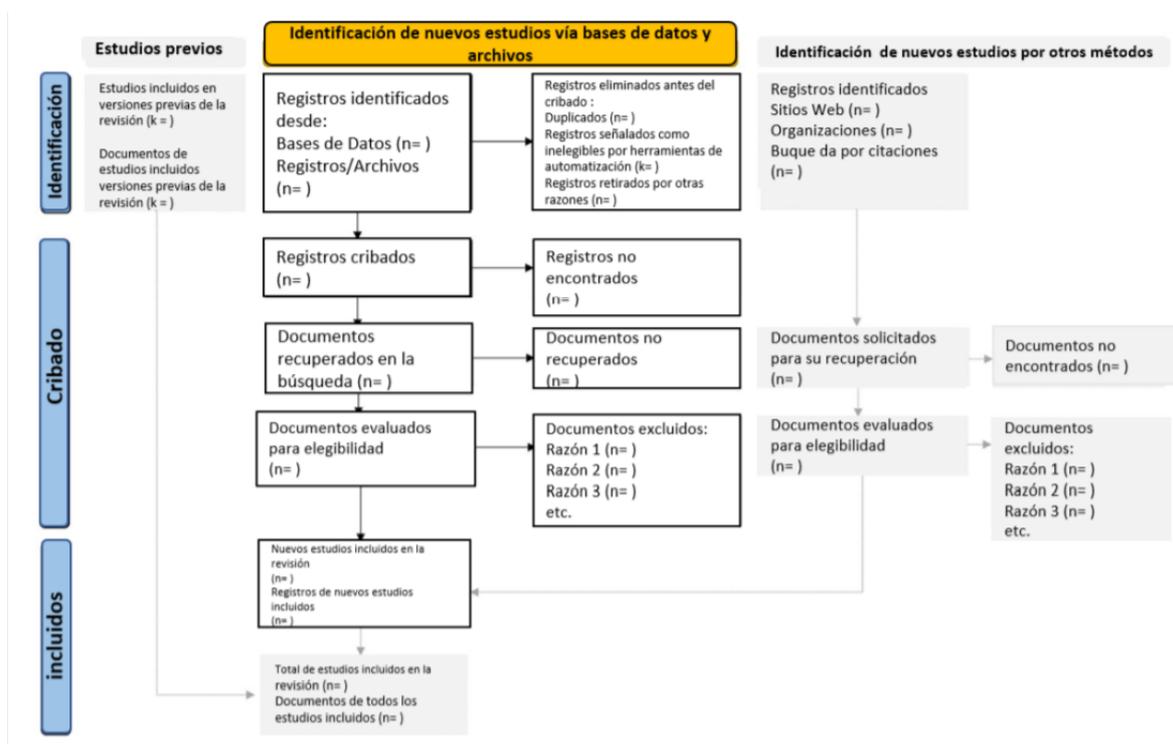
Tabla 2. Lista de comprobación de 27 elementos traducida al español de PRISMA 2020 (Page et al., 2021)

Sección y tema	#	elemento lista de verificación
TÍTULO		
Título	1	Identifique el informe como una revisión sistemática.
RESUMEN		
Resumen	2	Consulte la lista de verificación PRISMA 2020 para resúmenes.
INTRODUCCIÓN		
Justificación	3	Describa la justificación de la revisión en el contexto del conocimiento existente.
Objetivos	4	Proporcione una declaración explícita de los objetivos o preguntas que aborda la revisión.
MÉTODOS		
Criterios de elegibilidad criterios	5	Especifique los criterios de inclusión y exclusión para la revisión y cómo se agruparon los estudios para la síntesis.
Fuentes de información	6	Especifique todas las bases de datos, registros, sitios web, organizaciones, listas de referencias y otras fuentes buscadas o consultadas para identificar estudios. Especifique la fecha de la última búsqueda o consulta de cada fuente.
Estrategia de búsqueda	7	Presente las estrategias de búsqueda completas para todas las bases de datos, registros y sitios web, incluidos los filtros y límites utilizados.
Proceso de selección	8	Especifique los métodos utilizados para decidir si un estudio cumplió con los criterios de inclusión de la revisión, incluido cuántos revisores examinaron cada registro y cada informe recuperado, si trabajaron de forma independiente y, si corresponde, detalles de las herramientas de automatización utilizadas en el proceso.
Proceso de recopilación de datos	9	Especifique los métodos utilizados para recopilar datos de los informes, incluido cuántos revisores recopilaron datos de cada informe, si trabajaron de forma independiente, cualquier proceso para obtener o confirmar datos de los investigadores del estudio y, si corresponde, detalles de las herramientas de automatización utilizadas en el proceso.
Ítems de datos	10a	Enumere y defina todos los resultados para los que se buscaron datos. Especifique si se buscaron todos los resultados que eran compatibles con cada dominio de resultado en

		cada estudio (por ejemplo, para todas las medidas, puntos temporales, análisis) y, en caso contrario, los métodos utilizados para decidir qué resultados recopilar.
	10b	Enumere y defina todas las demás variables para las que se buscaron datos (por ejemplo, características del participante y de la intervención, fuentes de financiación). Describa las suposiciones hechas sobre cualquier información faltante o poco clara.
Evaluación del riesgo de sesgo del estudio	11	Especifique los métodos utilizados para evaluar el riesgo de sesgo en los estudios incluidos, incluidos los detalles de las herramientas utilizadas, cuántos revisores evaluaron cada estudio y si trabajaron de forma independiente y, si corresponde, detalles de las herramientas de automatización utilizado en el proceso.
Medidas de efecto	12	Especifique para cada resultado la medida o medidas de efecto (p. Ej., Razón de riesgo, diferencia de medias) utilizadas en la síntesis o presentación de los resultados.
Métodos de síntesis	13a	Describa los procesos utilizados para decidir qué estudios fueron elegibles para cada síntesis (por ejemplo, tabulando las características de la intervención del estudio y comparándolas con los grupos planificados para cada síntesis (item # 5)).
	13b	Describa los métodos necesarios para preparar los datos para su presentación o síntesis, como el manejo de estadísticas resumidas que faltan o las conversiones de datos.
	13c	Describa cualquier método utilizado para tabular o mostrar visualmente los resultados de estudios y síntesis individuales.
	13d	Describa cualquier método utilizado para sintetizar los resultados y proporcione una justificación para las opciones. Si se realizó un metanálisis, describa el modelo (s), método (s) para identificar la presencia y el grado de heterogeneidad estadística, y paquetes de software utilizados.
	13e	Describa cualquier método utilizado para explorar las posibles causas de heterogeneidad entre los resultados del estudio (por ejemplo, análisis de subgrupos, metarregresión).
	13f	Describa cualquier análisis de sensibilidad realizado para evaluar la solidez de los resultados sintetizados.
Evaluación del sesgo de notificación	14	Describa cualquier método utilizado para evaluar el riesgo de sesgo debido a resultados faltantes en una síntesis (que surgen de sesgos de notificación).
Evaluación de la certeza	15	Describa cualquier método utilizado para evaluar la certeza (o la confianza) en el conjunto de pruebas de un resultado.
RESULTADOS		
Selección de estudios	16a	Describa los resultados del proceso de búsqueda y selección, desde el número de registros identificados en la búsqueda hasta el número de estudios incluidos en la revisión, idealmente utilizando un diagrama de flujo.
	16b	Cite estudios que parezcan cumplir con los criterios de inclusión, pero que fueron excluidos, y explique por qué fueron excluidos.
Características del estudio		
Riesgo de sesgo en los estudios	18	Presentar evaluaciones del riesgo de sesgo para cada estudio incluido.
Resultados de estudios individuales	19	Para todos los resultados, presente, para cada estudio: (a) estadísticas resumidas para cada grupo (cuando corresponda) y (b) una estimación del efecto y su precisión (por ejemplo, intervalo de confianza / creíble), idealmente utilizando tablas estructuradas o parcelas.
Resultados de las síntesis	20a	Para cada síntesis, resume brevemente las características y el riesgo de sesgo entre los estudios contribuyentes.
	20c	Presentar los resultados de todas las investigaciones de las posibles causas de heterogeneidad entre los resultados del estudio.
	20d	Presentar los resultados de todos los análisis de sensibilidad realizados para evaluar la solidez de los resultados sintetizados.
	21	Presentar evaluaciones del riesgo de sesgo debido a la falta de resultados (que surgen de sesgos de notificación) para cada síntesis evaluada.
Certeza de la evidencia	22	Presentar evaluaciones de certeza (o confianza) en el cuerpo de evidencia para cada resultado evaluado.
DISCUSIÓN		
Discusión	23a	Proporcione una interpretación general de los resultados en el contexto de otra evidencia.
	23b	Discuta las limitaciones de la evidencia incluida en la revisión.
	23c	Analice las limitaciones de los procesos de revisión utilizados.
	23d	Discuta las implicaciones de los resultados para la práctica, la política y la investigación futura.

OTRA INFORMACIÓN		
Registro y protocolo	24a	Proporcione información de registro para la revisión, incluido el nombre de registro y el número de registro, o indique que la revisión no se registró.
	24b	Indique dónde se puede acceder al protocolo de revisión o indique que no se preparó un protocolo.
	24c	Describa y explique cualquier enmienda a la información proporcionada en el registro o en el protocolo.
Apoyo	25	Describa las fuentes de apoyo financiero o no financiero para la revisión y el papel de los financiadores o patrocinadores en la revisión.
Conflicto de intereses	26	Declare cualquier conflicto de intereses de los revisores.
Disponibilidad de datos, códigos y otros materiales	27	Informe cuáles de los siguientes están disponibles públicamente y dónde se pueden encontrar: formularios de recopilación de datos de plantilla; datos extraídos de los estudios incluidos; datos utilizados para todos los análisis; código analítico; cualquier otro material utilizado en la revisión.

Figura 2. Versión traducida Plantilla de diagrama de flujo PRISMA 2020 para revisiones sistemáticas (Page et al., 2021)



La declaración PRISMA es una de las principales herramientas que los investigadores tienen a su alcance para realizar una revisión sistemática y que de esta forma puedan informar de una manera transparente el porqué de cada revisión sistemática, cuáles fueron los procesos seguidos y los resultados encontrados. La declaración PRISMA 2020 es una versión actualizada de la original publicada en 2009, y será la utilizada para la realización de esta revisión sistemática.

3.3.3 Procedimiento

3.3.3.1 Estrategia de búsqueda y fuentes de datos

Antes de comenzar con el proceso de investigación en primer lugar debemos seleccionar las bases de datos que vamos a utilizar para realizar nuestras búsquedas. La búsqueda se ha realizado en dos etapas: en primer lugar, se utilizó una fuente de datos de habla hispana y en segundo lugar 3 fuentes de datos de habla inglesa. Asimismo, la presente investigación se basa en artículos publicados en revistas científicas, por lo que debemos tener especial atención en no hacer uso de repositorios. Estas bases de datos han sido seleccionadas al ofrecer un amplio catálogo de artículos focalizados en el ámbito educativo y que nos han permitido una descarga libre y completa de cada uno de los artículos.

Las seleccionadas son las siguientes:

1. SCOPUS (Inglés)
2. ERIC (Ingles)
3. REDALYC (Castellano)
4. SCIENCEDIRECT

Para ampliar nuestro rango de búsqueda se buscan keywords para el idioma inglés y otras para el español:

- a. Inglés: computational AND thinking and (evaluation or assessment)
- b. Español: pensamiento AND computacional

Los criterios de búsqueda aplicados en esta investigación están centrados en cuales van a ser los campos en los que vamos a realizar las búsquedas, las keywords comentadas y el rango de fechas de publicación. Todo ello en artículos procedentes de revistas científicas que ofrecen una disponibilidad completa y gratuita del artículo (ver Tabla 3).

Se detalla a continuación cada uno de los criterios con los que se van a concretar qué artículos van a ser incluidos en la revisión y cuales no:

1. C1. Acometer la búsqueda en los campos abstract o resumen nos va a permitir acceder a una cantidad de artículos mayor que incluyan nuestras palabras de búsqueda y por lo tanto reducir nuestro campo de acción, prescindiendo de artículos que no incluyan estos términos.

2. C2. Utilizar palabras clave en idiomas inglés y español nos permite ampliar nuestro radio de acción a nivel mundial al ser el inglés el idioma comúnmente utilizado en artículos de revistas científicas por un lado y utilizando el español, aunque de uso más reducido, nos permite incluir el mundo hispano-hablante, que, aunque de uso menor, tiene una cuota importante también. Utilizando todas estas palabras clave nos amplía por un lado el rango de localización de artículos, prescindiendo también de artículos que no nos vayan a interesar.
3. C3. Aunque en las revisiones sistemáticas suelen utilizarse rangos de fechas de publicación más amplio, en un campo en constante y evolución con el del pensamiento computacional, reduciremos a los artículos publicados en los últimos cinco años, lo cual nos permitirá no tener una visión desfasada en el estado de la cuestión.
4. C4. Tipo. Para realizar la investigación se seleccionarán exclusivamente aquellos artículos que hayan sido publicados en revistas científicas, de tal forma que nos permita asegurar una máxima calidad, rigurosidad y concreción en las investigaciones.
5. C5. Disponibilidad. Será necesario poder acceder al artículo en su versión completa para una revisión completa del mismo, asegurándonos poder acceder a toda la información que necesitamos para realizar nuestro estudio.

Tabla 3. Criterios de búsqueda aplicados en la investigación

N°		Criterio
C1	Campo	Title - Abstract - Full Text
C2	Palabras claves	Computational Thinking - Pensamiento computacional - Assessment - Evaluation
C3	Fecha de publicación	2016 - 2020
C4	Tipo	Artículo de Revista Científica
C5	Disponibilidad	Descargar papel completo

El motor de búsqueda de cada base de datos seleccionada utiliza diferentes mecanismos y estándares. Por lo tanto, se adaptó la cadena de búsqueda a cada base de datos, para llevar a cabo una búsqueda avanzada en títulos, resúmenes y palabras claves de artículos, obteniendo así un primer conjunto de estudios primarios (ver Tabla 4).

Tabla 4. Cadenas de búsqueda en bases de datos

Base de Datos	Cadena de búsqueda
ERIC	"computational thinking" AND (evaluation OR assessment)
Scopus	TITLE-ABS-KEY ("computational thinking" AND (evaluation OR assessment)) AND DOCTYPE (ar OR re) AND ACESSTYPE (OA) AND PUBYEAR > 2015
REDALYC	"pensamiento computacional"
SCIENCEDIRECT	"computational thinking" AND (evaluation OR assessment) después filtrar por años y marcar open access

3.3.3.2 Criterios de inclusión

Una vez seleccionadas las bases de datos nuestro siguiente paso es la definición de los criterios de inclusión que vamos a utilizar. Estos deben estar claros para ser usados en la fase de elegibilidad y que dé lugar a una selección de artículos que se adecue con los objetivos de la investigación. Se detalla a continuación cada uno de los criterios con los que se van a concretar los artículos van a ser incluidos en la revisión y cuáles no (ver Tabla 5):

1. CI1. Será requisito que incluya algún tipo de propuesta de intervención en torno al pensamiento computacional ya sea en aula con estudiantes o con docentes.
2. CI2. Será requisito que el artículo incluya algún tipo de medición para valorar el impacto de dicha intervención educativa.
3. C3. Se seleccionarán sólo aquellos que ofrezcan algún tipo de relación con constructos propios del pensamiento computacional, así como otros constructos de interés en etapas educativas de este estudio
4. CI4. Se seleccionarán sólo aquellas publicaciones cuyo rango de edad donde se ha realizado dicha intervención sean las etapas de Infantil y Primaria.

Tabla 5. Criterios de inclusión

Nº	Criterio

CI1	Contenido	Incluya intervención educativa
CI2	Estudio empírico	Incluya algún tipo de medición respecto a la evaluación de la intervención educativa
CI3	Contenido	Se establezca una relación con constructos de interés
CI4	Rango	Intervenciones exclusivamente en etapas de Infantil y Primaria

3.3.3.3 Criterios de exclusión

En los últimos años han aumentado la aparición de revisiones sistemáticas por los motivos comentados anteriormente. No obstante, para que puedan ofrecer la calidad que se espera de ellas, no solo es importante fijar de manera previa unos correctos criterios de inclusión, sino que también deberemos detallar qué criterios de exclusión (CE) debemos utilizar para descartar artículos de la selección final. Con estas decisiones previas se espera obtener unos resultados más rigurosos en los términos concretos de nuestra investigación, y que nos permitan alcanzar los objetivos propuestos.

Estos son los criterios predefinidos por los cuales se descartarán artículos de la revisión final (ver Tabla 6):

1. CE1. Se descartarán aquellos artículos que, si bien mencionan el pensamiento computacional, su relevancia en el contexto global es ínfima o no incluye una intervención educativa en torno al pensamiento computacional.
2. CE2. Se descartarán aquellos artículos que no hayan realizado algún tipo de evaluación para medir el impacto en el aula en torno al pensamiento computacional.

Tabla 6. Criterios de exclusión

Nº		Criterio
CE1	Contenido	Artículos que solo mencionan el pensamiento computacional pero donde no es el epicentro de la investigación o no incluye intervención

		relacionada
CE2	Contenido	Aquellos cuyo contenido no gire en torno a la evaluación del impacto del trabajo del pensamiento computacional

3.3.3.4 Informe de elegibilidad

A continuación, se detallan los procedimientos que se tuvieron en cuenta para realizar la selección final de artículos, que siguieron las recomendaciones de Frampton, Livoreil y Petrokofsky (2017):

1. Se utilizó un gestor de referencias bibliográficas (Refworks, cuenta de estudiante de Universidad de Zaragoza) donde se crearon 4 categorías de artículos para distinguir los seleccionados inicialmente en cada una de las 4 bases de datos elegidas para realizar la búsqueda. Este gestor facilitó la tarea de poder leer rápidamente el abstract de un artículo, sus autores y el año de publicación.
2. Se eliminaron los artículos duplicados en Refworks lo cual evitó un esfuerzo innecesario el trabajo de investigación.
3. No se evaluó la elegibilidad de los artículos de revisión, sin embargo, se utilizaron como fuente de estudios adicionales no identificados previamente
4. El proceso de screening nos aseguró que los criterios de elegibilidad eran aplicados de forma consistente e imparcial, para de esta forma reducir el riesgo de incluir errores. Los artículos identificados en las búsquedas tenían la estructura básica de título, abstract y/o un texto completo. Para asegurar la calidad en el proceso se siguieron los siguientes pasos según detalle el diagrama Prisma (ver Figura 3):
 - a. Identificación: realización de las búsquedas con las palabras clave elegidas en las 4 bases de datos previamente seleccionadas. Esto nos ofrece un total de 120 artículos encontrados, de los cuáles son eliminados 8 de ellos por ser duplicados y 3 por no tener una lectura completa de los artículos
 - b. Screening: Se aplicaron los criterios de inclusión y exclusión previamente establecidos. El uso de estos criterios explícitos y específicos aplicados de una manera transparente y objetiva, ayudaron a reducir posibles errores en el caso de que las decisiones de inclusión o exclusión fueran selectivas, subjetivas o inconsistentes y que no incluyeran artículos no relevantes para nuestra pregunta

de investigación. Estos criterios representan de forma objetiva la pregunta de investigación, así como el objetivo general y específicos de nuestra investigación, de tal forma que facilita la exclusión de los artículos que no reunían los requisitos previamente establecidos. En primer lugar, se realiza una primera criba entre los artículos que no cumplen los criterios de inclusión tras la lectura de título y abstract para eliminar de forma eficiente artículos irrelevantes. A continuación, se eliminan aquellos que cumplen con los criterios de inclusión. Finalmente, en casos donde la información incluida en título y abstract fuera insuficiente, se procedió a la lectura completa del artículo en cuestión, donde se elimina finalmente un artículo debido a su bajo nivel en el diseño de investigación (Bolstad, 2016).

- c. Incluidos: finalmente son 21 artículos los que finalmente son elegidos para la realización de esta revisión sistemática en profundidad.

3.3.3.5 Extracción de datos

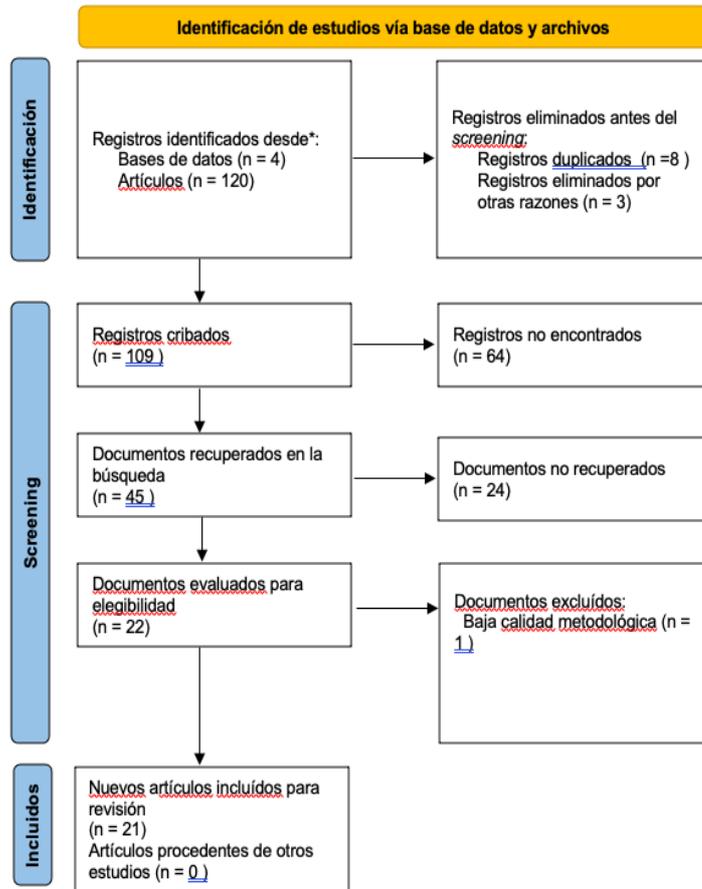
Una vez finalizada la selección de los artículos, es necesario extraer la información que nos ayude a identificar dichos artículos, así como la información clave que nos ayude a resolver nuestras preguntas de investigación.

Para realizar este proceso se creó un formulario de google y los datos recogidos fueron tabulados en una hoja de cálculo con los siguientes campos (ver Figura 4):

1. Código
2. Título de artículo
3. Nombre de primer autor
4. Año de publicación (2016 /2017 /2018 / 2019 / 2020)
5. Idioma (Español / Inglés)
6. Etapa educativa de la intervención (Infantil / Primaria)
7. Herramienta/Técnica de evaluación
8. Ambiente de aprendizaje (formal / informal)
9. Cita APA
10. Fuente

11. Habilidades relacionadas con el pensamiento computacional
12. Relación con otros constructos (SI/NO)
13. Otras variables que se tienen en cuenta
14. Observaciones (se recogen principales hitos del artículo)

Figura 3. Diagrama PRISMA de identificación de estudios



Las características de los artículos seleccionados es que incluyeran algún tipo de intervención educativa en las etapas de infantil y primaria, que aportarán algún tipo de evaluación del impacto de dicha intervención, así como ofrecieran relaciones con otros constructos de interés.

Figura 4. Extracción de datos con Google Form

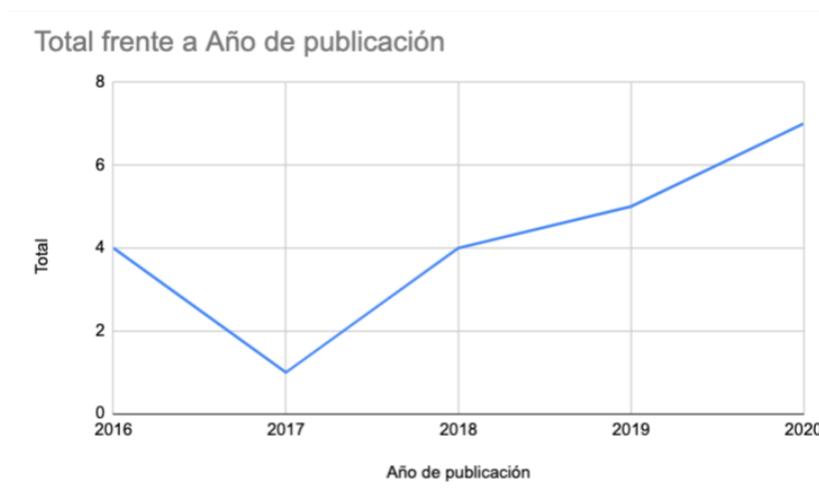
Código	Título del artículo	Nombre de primer autor	Año de publicación	Lidioma	Etapa educativa	Herramienta/Técnica
ARPC0001	Acompañamiento pedagógico del Programa de Tecnologías para el Aprendizaje	Castro Araya, Hazel	2020	Castellano	Infantil	Entrevista
ARPC0002	Computational thinking through unplugged activities in early years of Primary	del Olmo-Muñoz, J.	2020	Inglés	Infantil	Test validado, Cuestionario
ARPC0003	Designing Computational Thinking and Coding Badges for Early Childhood Ed	Hunsaker, E	2019	Inglés	Infantil	Cuestionario
ARPC0004	Learning to code via tablet applications: An evaluation of Daisy the Dinosaur a	Pila, S.	2019	Inglés	Infantil	Entrevista, Observación
ARPC0005	Robótica para desarrollar el pensamiento computacional en Educación Infantil	Muñoz-Repiso, A. G. V.,	2019	Castellano	Infantil	Test validado
ARPC0006	Extending the nomological network of computational thinking with non-cogniti	Román-González, M.	2018	Inglés	Infantil, Primaria	Test validado
ARPC0007	Which cognitive abilities underlie computational thinking? Criterion validity of	Román-González, M.	2017	Inglés	Infantil, Primaria	Test validado
ARPC0008	A Computation Model for Learning Programming and Emotional Intelligence	Memoona Rafique	2020	Inglés	Primaria	Test validado
ARPC0009	A study of primary school students' interest, collaboration attitude, and progr	Kong, S. C.	2018	Inglés	Primaria	Test validado
ARPC0010	Active Learning Environments with Robotic Tangibles: Children's Physical and	Burleson, W. S.	2018	Inglés	Primaria	Diario de campo
ARPC0011	Análisis del efecto de la robótica en la motivación de estudiantes de tercero de	Merino-Armero, José Miguel	2018	Castellano	Primaria	Test validado
ARPC0012	Assessing computational thinking process using a multiple evaluation approac	Yasemin Allsop	2018	Inglés	Primaria	Portfolio, Entrevista, Test
ARPC0013	Assessing elementary students' computational thinking in everyday reasoning	Guanhua Chen	2017	Inglés	Primaria	Test validado
ARPC0014	Computational thinking in primary education: a systematic literature review	Panagiotis Kakavasa	2019	Inglés	Primaria	Artículo de interés
ARPC0015	Designing a Programming Game to Improve Children's Procedural Abstraction	Simon P. Rose	2020	Inglés	Primaria	Test validado, Test valid
ARPC0016	Development and Validation of Unplugged Activity of Computational Thinking	Tracy Mensan	2020	Inglés	Primaria	Test validado
ARPC0017	Exploring Media Literacy and Computational Thinking: A Game Maker Curricul	Jennifer Jenson	2016	Inglés	Primaria	Cuestionario, Diario de
ARPC0018	Game-Coding Workshops in New Zealand Public Libraries: Evaluation of a Pilot Proj	Bolstad, R	2016	Inglés	Primaria	Cuestionario
ARPC0019	Instructional supports for students with disabilities in K-5 computing: Findings	Snodgrass, M. R.	2016	Inglés	Primaria	Entrevista, Observación
ARPC0020	Primary School Pupils' Attitudes toward Learning Programming through Visua	Asad, K.	2016	Inglés	Primaria	Cuestionario, Entrevist
ARPC0021	Programación de tecnologías para la inclusión social con Scratch: Prácticas s	Monjelat, Natalia	2019	Castellano	Primaria	Cuestionario, Portfolio
ARPC0022	Teacher development in computational thinking: Design and learning outcome	Kong, S. C.	2020	Inglés	Primaria	Test validado, Cuestion
ARPC0023	TechCheck: Development and Validation of an Unplugged Assessment of Com	Relkin, E.	2020	Inglés	Primaria	Test validado

4 - Resultados

El proceso de selección de los artículos que finalmente han sido escogidos para analizar en profundidad en busca de dar respuesta a los objetivos de esta investigación, nos ofrece una serie de datos, que aún no teniendo la importancia de los que vendrán a continuación, tienen cierto valor descriptivo merecedores de ser detallados.

En cuanto a la fecha de publicación, el estudio restringe su búsqueda a los últimos 5 años. En los resultados se revela que, a excepción del año 2017, desde 2016 el número de artículos publicados en esta área está en crecimiento (ver Figura 5).

Figura 5. Número de artículos por cada año de publicación



En cuanto al idioma en el que los artículos han sido escritos, existe una clara mayoría de los escritos en idioma inglés (17), respecto a los que se pudieron localizar en castellano (4) (Ver Figura 6):

Figura 6. Porcentaje de publicaciones por idioma



A continuación, se detalla apartado por apartado los principales resultados obtenidos en cuanto a la pregunta y los objetivos planteados en esta investigación.

4.1 Análisis y clasificación de las diferentes herramientas de evaluación del impacto en el aula

La lectura detallada de cada uno de los artículos nos ofrece información sobre las diferentes técnicas o herramientas que utilizaron los autores para medir el impacto de su intervención. De entre todas ellas, destacan principalmente los test, y en menor medida, los cuestionarios (ver Tabla 7).

Tabla 7. Desglose del nº de veces que se ha utilizado cada herramienta/técnica

Herramienta/Técnica	Total
Test	16
Cuestionario	5
Porfolio	1
Entrevista	3
Diario de campo	3
Técnica observacional	2
Estudio de caso	2
Redacciones	1
Grabaciones	3
Badgets	1

Los test son utilizados primordialmente para medir aspectos relacionados con algunos de los factores cognitivos asociados o alguna otra variable de interés de forma concreta. La mayoría de ellos han sido utilizados para la medición del nivel de pensamiento computacional ya sea en alumnos o docentes:

- a. en el diseño y validación de una herramienta para medir el pensamiento computacional con una muestra de 1251 alumnos españoles (Román-González et al., 2017). Las conclusiones arrojan evidencias de confiabilidad para poder usar este test en alumnos desde 5º primaria hasta 4º ESO. En el mismo estudio se utilizaron dos test para medir capacidades mentales y resolución de problemas con el objetivo de validar la correlación de estas capacidades con las habilidades básicas del pensamiento computacional (Román-González et al., 2017). El mismo autor utiliza el mismo test para buscar una correlación entre los resultados de este test y constructos como la

autoeficacia y diferentes rasgos de la personalidad (Ramón-González, et al., 2018). Este test también es utilizado en diseños pre-post para medir la mejora de pensamiento computacional tras una intervención (Rose, Habgood y Jay, 2020)

- b. medir el pensamiento abstracto (parte del pensamiento computacional) mediante la utilización de un videojuego enfocado a practicar esa área donde el objetivo principal es evitar el desarrollo de programas excesivamente largos (con código duplicado) mediante la utilización de funciones (Rose et al., 2020)
- c. en niños de 5 a 9 años sin experiencia en programación para analizar la mejora producida tras una intervención con actividades unplugged (sin ningún tipo de tecnología de programación o robótica) como desafíos de secuenciación, series de símbolos perdidos, caminos de resolución de puzzles o problemas de simetría (Relkin, Ruiter y Bers, 2020)
- d. en un programa de formación específico en habilidades del pensamiento computacional para docentes en Hong Kong, donde se demostró que con un programa de capacitación básico los docentes pueden mejorar el entendimiento de las habilidades del pensamiento computacional y la necesaria pedagogía para llevarlo al aula (Kong, Lai y Sun, 2020)
- e. en programas orientados a mejorar la competencia curricular en la asignatura de Ciencias en Primaria mediante la inclusión de actividades unplugged como pueden ser los diagramas de flujo y los mapas conceptuales a partir de los cuales se trabajaban habilidades del pensamiento computacional orientadas a la resolución de problemas (Mensan, Osman y Majid, 2020)
- f. para evaluar el éxito de un programa curricular en computer science mediante un test que evaluaba conceptos de pensamiento computacional en dos tipos de situaciones: programación de robots y situaciones cotidianas del día a día (Chen et al., 2107).
- g. un tipo especial de test fue uno realizado ad-hoc a partir de diferentes pruebas extraídas del Bebras Contest para medir el pensamiento computacional con preguntas dirigidas a cualquier persona sin conocimientos de programación. El objetivo era medir el nivel de pensamiento computacional en un grupo de niños sin experiencia previa, en una etapa intermedia de la intervención y el post, donde se pudo corroborar una clara mejoría especialmente en el grupo experimental que realizó actividades mixtas unplugged-plugged respecto del grupo control que solo hizo actividades plugged del Olmo-Muñoz, Cózar-Gutiérrez y González-Calero, 2020)

- h. para analizar la efectividad del aprendizaje de conceptos básicos del pensamiento computacional mediante dispositivos móviles como tablets usando dos apps diferentes (Pila, Alade, Sheehan, Lauricella y Wartella, 2019)

Han sido utilizados también para medir el aumento de la motivación:

- a. analizando la influencia de la robótica en el grado de motivación del alumnado de tercer curso de Educación Primaria en la entrega de tareas enfocadas hacia la orientación espacial e interpretación de planos asociadas a aprendizajes curriculares de matemáticas y ciencias sociales. Para ello se utilizó un test específico para evaluar la motivación de los alumnos, partiendo del *Instructional Material Motivational Survey* (Keller, 1983), donde se puede analizar la motivación en cuatro dimensiones: atención, relevancia, confianza y satisfacción (Armero, Taranilla, Somoz y Gutiérrez, 2018)
- b. analizar la motivación hacia el aprendizaje de pensamiento computacional con actividades unplugged y plugged en educación primaria, aislando también esa posible mejora de la motivación diferenciando entre chicos y chicas (del Olmo-Muñoz et al., 2020)

Hay un caso especial dentro de la categoría de los métodos cuantitativos que serían las rúbricas, utilizadas para evaluar el desempeño de los estudiantes ante retos en los que tienen que demostrar estas habilidades computacionales, inicialmente y una vez que han completado un programa formativo con actividades de robótica. Y de este modo, valorar la eficacia del programa en función de las capacidades desarrolladas por estos estudiantes. Esta rúbrica es una adaptación de la rúbrica SSS empleada en el programa TangibleK (Bers, 2010). La rúbrica es aplicada por el investigador y la maestra de forma conjunta, consensuando la valoración de los resultados obtenidos por cada uno de los estudiantes (Muñoz-Repiso y González, 2019).

Los cuestionarios han sido utilizados para los siguientes propósitos:

- a. para medir el grado de percepción propia de mejora en competencia computacional tras un programa de capacitación de docentes (Kong et al., 2020)
- b. analizar la mejora en inteligencia emocional junto con la resolución de problemas mediante el uso de un robot. La inteligencia emocional es otra de las habilidades cognitivas que rara vez se tiene en cuenta y fue incluida en la misma planificación de resolución de problemas de elección del camino más corto con un robot. Los resultados del test arrojan una mejora en la inteligencia emocional en todas las muestras de niños de 4 a 8 años (Rafique, Hassan, Jaleel, Khalid y Bano, 2020)

- c. conceptualizar el empoderamiento en el aprendizaje de la programación en conceptos como la significación, el impacto, la auto-eficacia en creatividad y la autoeficacia en programación, además de buscar relaciones con el interés en el aprendizaje de programación, actitud hacia el trabajo colaborativo en programación, género y edad (Kong, et al., 2018)
- d. el nivel de competencias digitales en general y el nivel de conocimientos de conceptos básicos del pensamiento computacional como variables, bucles o funciones, así como para evaluar posibles diferencias de género en la actitud y confianza hacia el aprendizaje de programación (Jenson & Droumeva, 2016)
- e. en un estudio cuyo objetivo era analizar la actitud hacia el aprendizaje de programación mediante entornos visuales. Para ello los investigadores adaptaron un cuestionario basado en uno ya existente (Baser, 2013) con 29 preguntas relacionadas con la actitud hacia el aprendizaje de programación. Además se apoyaron en métodos cualitativos mediante la recopilación de datos basados en entrevistas y observaciones de los comentarios y comportamientos físicos de los alumnos, así como observaciones a través de grabación de vídeo e imágenes, que sirvieron para clarificar los datos obtenidos de este cuestionario, donde los resultados fueron positivos antes y también después de la intervención, mejorando también sus habilidades en pensamiento computacional, aunque sin aclarar cuáles de ellas (Asad, Tibi y Raiyn, 2016)

Podemos también encontrar estudios que se han apoyado en métodos cualitativos:

- a. una investigación donde se han usado observación participante, entrevistas informales, así como entrevistas semi-estructuradas para lograr una evaluación mediante diferentes aproximaciones no sólo para medir el pensamiento computacional, sino también las prácticas metacognitivas y comportamientos durante el aprendizaje de los estudiantes. Las entrevistas informales y la observación participante permitieron examinar el lenguaje de los alumnos en sus propias autoexplicaciones o discusiones de grupo, los gestos y la relación con el profesor y otros alumnos. La observación participante permitió observar cómo los alumnos testaban continuamente su código y sus errores, y cómo ellos los resolvían, así como todos se ayudaban unos a otros a solucionar los problemas que surgían. Las entrevistas semi-estructuradas fueron usadas para destapar el conocimiento más profundo del proceso con algunos de los participantes (Allsop, 2019)

- b. grabaciones para estudiar interacciones, colaboraciones y la capacidad de secuenciación de los estudiantes en entornos de programación espacial de manipulación física con robots a diferencia de la utilización de un entorno virtual de programación. Estas grabaciones permitieron analizar los diferentes tipos de interacciones que se producían en el entorno físico a diferencia del virtual, así como la forma en la que colaboran entre ellos y la capacidad de improvisación en la resolución de problemas (Burleson et al., 2017)
- c. notas de campo con grabaciones de las reacciones de estudiantes para profundizar en cuestiones de diferencias de género en cuanto a la actitud y confianza en el aprendizaje de programación (Jenson & Droumeva, 2016)
- d. reflexiones en formato de redacciones de un máximo de 500 palabras de los propios participantes sobre su proceso de mejora en conocimientos sobre habilidades del pensamiento computacional tras un proceso de capacitación específico de esas áreas en docentes de Hong Kong (Kong et al., 2020)
- e. entrevistas individuales para investigar sobre las decisiones que cada alumno tomó en la resolución de desafíos de programación (Rose et al., 2020)
- f. grabaciones al comienzo y final de la intervención con entrevistas a cada uno de los participantes en un programa piloto que quería conocer la efectividad de utilizar tablets en la enseñanza de programación en la etapa de infantil y también si la apariencia visual de la app mejoraba el interés por practicar con dicha aplicación (Pila et al., 2019)
- g. residualmente encontramos dos estudios de caso: en uno de ellos se persigue la mejora docente mediante creación de tecnologías de inclusión social a través de la programación, con el propósito de ofrecer una investigación inicial que sugiera un diseño de experiencias de formación docente en pensamiento computacional desde una perspectiva situada, significativa y compleja, y donde se utilizó la narrativa a partir de unas grabaciones y una posterior fase analítica, donde las transcripciones de esas grabaciones se clasificaron en un sistema de categorías, focalizando el análisis en el material discursivo obtenido a partir de las transcripciones. Este tipo de análisis permitió estudiar en profundidad cómo los docentes se acercan a los conceptos básicos del pensamiento computacional, centrándose en el cómo, en un contexto de creación específico (Monjelat, 2019); en el otro se busca crear un marco de trabajo para que los alumnos con algún tipo de discapacidad puedan disfrutar también del aprendizaje del pensamiento computacional utilizando la observación de las reacciones de alumnos ante las tareas y entrevistas semi-estructuradas con sus profesores para conocer si era

necesario crear unos recursos específicos para este área o una instrucción convencional podría ser suficiente (Snodgrass, Israel y Reese, 2016)

Por último, mencionar un estudio de caso donde se realizó ad-hoc un itinerario formativo a profesores para aprender conceptos básicos de programación y la forma de integrarlos en el currículo de infantil, donde la forma de comprobar que se alcanza determinado nivel de conocimiento y dominio en pensamiento computacional era a través de un sistema de reconocimiento de insignias (budgets) (Hunsaker y West, 2020)

4.2 Análisis de las evidencias empíricas de las intervenciones

A continuación, se detallan las investigaciones que hayan aportado algún tipo de evidencia empírica de sus intervenciones, de las herramientas utilizadas o de los constructos respectivos analizados.

En primer lugar, se destacan los artículos cuyo eje principal era medir la motivación ya sea como elemento para hacer más atractivos los contenidos de una asignatura o para comprobar qué forma de enseñar enganchaba más a los alumnos:

- a. un estudio donde la utilización de la robótica en las áreas de matemáticas y ciencias sociales en 3º de Primaria obtuvo unos resultados favorables en los 4 componentes de la motivación estudiados (atención, relevancia, confianza y satisfacción), destacando especialmente el de la satisfacción (Armero et al., 2018)
- b. en la validación de una herramienta para conceptualizar el empoderamiento en el arte de la programación en estudiantes mediante el uso de un diseño exploratorio donde el análisis de factorial confirmatorio ofreció resultados como que los alumnos más motivados durante la intervención en la enseñanza de programación le encontraron más significado e impacto y demostraron mayor autoeficacia en creatividad y programación. Por otra parte, los alumnos que demostraron mayor actitud hacia el trabajo colaborativo tenían más autoeficacia creativa pero no más autoeficacia en programación (Kong et al., 2018)
- c. en un estudio donde había 3 objetivos a evaluar tras la intervención con actividades plugged and unplugged: mejoría del nivel de pensamiento computacional por un lado y la motivación por otro, y por último evaluar estos dos resultados respecto al género. Los resultados estadísticos concluían que tanto el nivel de motivación como el de nivel de pensamiento computacional aumentaban especialmente en el grupo experimental que tuvo actividades mixtas y el de motivación especialmente en el de actividades

unplugged respecto al de actividades plugged, donde además hubo grandes diferencias de género puesto que la motivación de las chicas cayó significativamente (del Olmo-Muñoz et al., 2020)

Un grupo importante de los artículos que ofrecieron datos contrastados fueron aquellos que se centraban en validar herramientas para la medición y/o validación del nivel de pensamiento computacional:

- a. en intervenciones orientadas a medir el nivel de pensamiento computacional y su correlación con capacidades mentales y resolución de problemas (Román-González et al., 2017) así como la relación del pensamiento computacional con constructos como la autoeficacia y rasgos de personalidad (Ramón-González et al., 2018).
- b. medir la mejora del pensamiento abstracto (P. Rose, 2020) donde se utiliza el test validado por Román-González (2017).
- c. medir el pensamiento computacional en actividades sin tecnología en niños de 5-9 años (Relkin et al., 2020). Este test tuvo inicialmente el feedback de 19 expertos (investigadores, educadores y estudiantes) que eliminaron por consenso 4 de las 19 preguntas inicialmente. Los resultados del test aplicados a más de 700 estudiantes correlaciona positivamente en sus resultados con otro test que medía las mismas habilidades (TACTIC-KIBO assessment) y que fue también administrado al mismo grupo de alumnos (Relkin y Bers, 2019)
- d. comprobar el impacto del programa de pensamiento computacional con robots en la etapa de infantil (Muñoz-Repiso y González, 2019). En primer lugar, se realizó un estudio de normalidad de la muestra poblacional empleando para esto el test de normalidad Kolmogorov-Smirnov. Posteriormente se realizó el análisis de los datos obtenidos en el pre-test y post-test en los grupos control y experimental, donde se arrojaron los resultados favorables hacia el grupo experimental
- e. mejorar los conocimientos en el área de ciencias del currículum británico mediante actividades unplugged orientadas a la resolución de problemas. Se realizó un pre y un post en habilidades del pensamiento computacional y se evidenció una clara mejora en la puntuación del grupo experimental (Mensan et al., 2020)
- f. evaluar un programa de competencia curricular en computer science donde se comprueba la validez de las propiedades psicométricas de un test desarrollado por los autores que busca una correlación entre el aprendizaje de habilidades básicas del pensamiento computacional y resolución de situaciones cotidianas del día a día, donde se evidenció dificultades en esta correlación y en la discusión se propone como mejora

profundizar en esa transferencia de conocimientos en la siguiente versión del currículo (Chen et al., 2107)

El resto de artículos focalizan su interés en constructos diversos, de los cuales pasamos a explicar sus aportaciones uno a uno:

- a. medir la actitud hacia el aprendizaje de programación con herramientas visuales, donde no se encontraron diferencias significativas en el análisis estadístico del pre y el post, por lo que se concluyó que la actitud era favorable antes y después de la intervención educativa (Asad et al., 2016)
- b. ofrecer datos estadísticos sobre la mejora en inteligencia emocional gracias a la utilización de un cuestionario validado anteriormente (Zang, 2017) tras una intervención en pensamiento computacional mediante el uso de robots. Este test está basado en reconocimiento de expresiones faciales (Rafique et al., 2020)
- c. medir la mejora de la competencia docente en el área de pensamiento computacional mediante la codificación de 76 redacciones procedentes de reflexiones de docentes tras un proceso de capacitación profesional sobre programación y posterior análisis de dos investigadores en busca de menciones sobre mejora de actitud y conocimiento del pensamiento computacional (Kong et al., 2020)
- d. medir la efectividad de aprender a programar con tablets y la importancia que tiene la apariencia de las apps utilizadas para estimular el aprendizaje. Las conclusiones del estudio fueron que las tablets pueden ser una herramienta válida para la enseñanza en esta área y que la apariencia de las apps y su usabilidad es un factor importante en el proceso de aprendizaje de los alumnos (Pila et al., 2019).
- e. Mención especial merece uno de los artículos donde se hacía una revisión sistemática de las formas de evaluar el pensamiento computacional en Primaria (Kakavas y Ugolini, 2019) y donde se citaba específicamente a uno de los autores que a su vez más ha sido citado en los artículos seleccionados debido al Test de Pensamiento Computacional creado por él mismo y que es utilizado en varios de ellos (Román-González et al., 2017)

4.3 Análisis de los entornos de aprendizaje (formal/informal)

Uno de los criterios de inclusión de este trabajo es que los artículos seleccionados deberían tener relación con las etapas de Infantil y Primaria. Tras el análisis se concluye que el número de experiencias desarrolladas en la Etapa de Infantil aún son clara minoría en contraste con las experiencias que tengan relación con la Etapa de Primaria (ver Tabla 8).

Tabla 8. Intervenciones educativas en Educación Infantil y Educación Primaria

Etapa	Total
Infantil	5
Primaria	16

En cuanto a cuál ha sido el ambiente en el que se han desarrollado estas intervenciones (formal/informal), solo 7 de los 21 artículos seleccionados se basan en experiencias realizadas en ámbitos más informales, donde destacan los procesos de capacitación de docentes:

Las desarrolladas en aula podemos encontrar las siguientes:

- a. comprobar la intervención respecto a la mejora en inteligencia emocional en unas sesiones dedicadas a la resolución de problemas con robots (Rafique et al., 2020)
- b. analizar la eficacia de un programa de pensamiento computacional que buscaba no solo medir el nivel de pensamiento computacional sino también el índice de transferencia de estas habilidades a resolución de situaciones cotidianas del día a día (Chen et al., 2107)
- c. estudio longitudinal para la evaluación del pensamiento computacional desde diferentes métodos cualitativos donde también se evaluaba los procesos metacognitivos y los comportamientos respecto al proceso de aprendizaje de un grupo de alumnos (Allsop, 2019)
- d. medir el nivel de desempeño de alumnos de infantil con robots (Muñoz-Repiso y González, 2019)
- e. mejora curricular en asignaturas de Ciencias en Primaria (Mensan et al., 2020)
- f. evaluar la actitud hacia el aprendizaje de programación con entornos visuales (Asad et al., 2016) o mezclando entornos de manipulación física en contraposición a programación en entornos virtuales (Burlson et al., 2017)
- g. estudio en alumnos para analizar cuestiones de género en el interés y actitud hacia el aprendizaje de programación (Jenson & Droumeva, 2016)

- h. promover la inclusión del pensamiento computacional con alumnos con algún tipo de discapacidad, así como evaluar el tipo de recursos necesarios (Snodgrass et al., 2016)
- i. validar herramientas donde se pueda evaluar el pensamiento computacional en actividades “unplugged” (sin tecnología) (Relkin et al., 2020; del Olmo-Muñoz et al., 2020)
- j. validar un test para medir el pensamiento computacional y buscar correlaciones con autoeficacia y rasgos de la personalidad (Ramón-González et al., 2017; Ramón-González et al., 2018)
- k. analizar los conceptos clave del empoderamiento en programación y su relación con actitudes como el trabajo colaborativo o el género (Kong et al., 2018).

4.4 Análisis de constructos relativos al pensamiento computacional

En cuanto a los procesos cognitivos propios del pensamiento computacional, en algunos casos nos encontramos que se focalizan especialmente en alguno de ellos, como en el pensamiento abstracto (Rose, 2020) y en otros una variedad de habilidades relacionadas con el pensamiento computacional (ver Tabla 9):

- a. prácticas del pensamiento computacional como el direccionamiento básico, las secuencias, bucles repetitivos con y sin condiciones, estructuras de selección simples y complejas y funciones sencillas (Román-González et al., 2017; Román-González et al., 2018)
- b. pensamiento algorítmico, modularidad, estructuras de control, representación y debugging mediante un test que utilizaba desafíos unplugged tales como la resolución de caminos más cortos, puzzles o problemas de simetría (Relkin y Bers, 2019)
- c. secuencias: capacidad de secuenciar acciones para dar respuesta a un desafío a través de actividades de programación. Correspondencia acción-instrucción: capacidad de relacionar las instrucciones que dan a un robot con la acción que desempeña. Depuración: habilidad para identificar y corregir los errores existentes en una secuencia de programación (Muñoz-Repiso y González, 2019)
- d. conceptos básicos cómo la selección, iteración o evaluación en la salida de un programa en un test sobre conceptos del pensamiento computacional y prácticas de debugging, iteración, reusar código, abstracción, modularización y pensamiento algorítmico en otro test sobre prácticas de programación básicas (Kong et al., 2020)

- e. prácticas resultantes del pensamiento computacional como experimentar y crear, testear y depurar, reusar y remezclar así como abstraer y modular (Monjelat, 2019)
- f. resolución de problemas y pensamiento lógico-matemático en la creación de materiales didácticos específicos para el área de educación musical en estudiantes de Magisterio (Castro, Ríos y Arguedas, 2020)
- g. abstracción, pensamiento algorítmico, descomposición de problemas y reconocimiento de patrones, utilizados para resolver problemas cotidianos en el área de Ciencias en Primaria (Mensan et al., 2020)
- h. secuencias, bucles, eventos, paralelismo, condicionales, operadores, variables y abstracción fueron los conceptos analizados en un estudio longitudinales cuyo objetivo principal era evaluar los procesos metacognitivos así como los comportamientos que los alumnos utilizaron durante el proceso de aprendizaje para desde una evaluación múltiple analizar cómo se alcanzaban estos conceptos (Allsop, 2019)
- i. resolución de problemas mediante los conceptos de secuenciación, selección y repetición en la resolución de caminos más cortos utilizados por robots para llegar a su destino (Rafique et al., 2020)
- j. abstracción, pensamiento algorítmico, descomposición, evaluación y generalización, en un estudio donde se pretendía analizar la mejora de estos conceptos después de una intervención con actividades plugged y unplugged mediante una selección de pruebas procedentes de varios Bebras Contest. El resultado final arrojó resultados a favor especialmente en el grupo que tuvo actividades mixtas respecto al grupo que solo tuvo plugged (del Olmo-Muñoz et al., 2020)
- k. habilidades del pensamiento computacional en general sin mencionar ninguna en concreto en una intervención educativa orientada a medir el grado de interés hacia el aprendizaje de programación con herramientas visuales (Asad et al., 2016)
- l. resolución de problemas, pensamiento algorítmico, pensamiento lógico-matemático, la depuración o la abstracción en un estudio de caso donde se pretendía también analizar aspectos como la colaboración, interacciones o la improvisación comparando dos contextos de aprendizaje: el físico mediante la manipulación de robots o el virtual con un entorno virtual de programación de robots (Burlson et al., 2017)

- m. constructos como la descomposición, paralelismo, abstracciones y generalización de patrones mediante conceptos computacionales como las variables, las operaciones matemáticas, las funciones y las estructuras condicionales (Jenson & Droumeva, 2016)
- n. también se seleccionaron estudios donde no solo se medían las habilidades básicas relacionadas con el pensamiento computacional (resolución de problemas, pensamiento algorítmico) sino que buscan también la mejora en la aplicación del razonamiento en las tareas del día a día (Chen et al., 2107).

Tabla 9. Relación de constructos propios del pensamiento computacional

Constructo	Total
Resolución de problemas	7
Abstracción	4
Pensamiento lógico-matemático	3
Pensamiento algorítmico	4
CT Skills en general	13
Depuración	13

4.5 Análisis de otros constructos relacionados con el pensamiento computacional

Otro de los principales objetivos de esta investigación era seleccionar aquellos artículos que establecieran algún tipo de relación con otros constructos u otros tipos de variables de interés para la propia investigación. Han sido muchos y variados, entre los que destacan el género y la motivación (ver Tabla 10).

En cuanto a los factores cognitivos encontrados destaca especialmente el de motivación ya sea por mejorar el interés por una asignatura mediante la utilización de la robótica para la resolución de problemas planteados, elevando los niveles de satisfacción, confianza, relevancia y atención (Armero et al., 2018) y también en otro estudio donde uno de los objetivos era medir el nivel de motivación en cuatro categorías (atención, relevancia, confianza y satisfacción) y donde se pudo observar una clara mejoría de la motivación por el aprendizaje de pensamiento computacional mediante actividades unplugged respecto de plugged, y sobre todo en chicas (del Olmo-Muñoz et al., 2020).

En otro estudio se busca evaluar el empoderamiento en programación en estudiantes prestando especial atención en conceptos como la autoeficacia creativa y de programación,

significado e impacto. Además, confirma la relación entre el interés en el trabajo colaborativo para aprender a programar con una mejor eficacia creativa (Kong et al., 2018). Allsop (2019) propone una aproximación múltiple desde diferentes perspectivas cualitativas (procesos metacognitivos y comportamientos) para analizar cómo se alcanzaban esos conceptos relacionados con el pensamiento computacional. Las prácticas metacognitivas analizadas fueron la planificación, la monitorización, la evaluación y el autocuestionamiento.

Tabla 10. Relación de otros constructos relacionados con el pensamiento computacional

Constructo	Total
Motivación	3
Inteligencia Emocional	1
Capacidades mentales	1
Autoeficacia	2
Mejora docente	3
Mejora curricular	2
Interés hacia el aprendizaje	1
Género	4
Colaboración	1
Perseverancia	1
Comunicación	1
Planificación	1
Autocuestionamiento	1
Rasgos de la personalidad	1

La planificación fue utilizada por todos los alumnos al comienzo de realizar los programas y en algunos casos utilizaban dibujos para comunicar sus ideas, otros textos exclusivamente y otros una mezcla de ambas. Las entrevistas semi-estructuradas ofrecían como uno de los datos interesantes que los alumnos después de haber utilizado la planificación para hacer sus tareas, habían visto su utilidad y pensaban utilizarla para otras tareas. En cuanto a la monitorización y evaluación, los estudiantes se dieron cuenta de que al ejecutar el programa y obtener errores, estaba en sus manos monitorizar esos errores y repararlos, llegando a explicar cómo habían tenido dichos errores y explicar la solución. Respecto a los comportamientos en el aprendizaje destacan la colaboración, donde las notas de campo reflejaban cómo los alumnos se levantaban y caminaban para ver lo que otros hacían o para hacer sugerencias. La perseverancia fue otro comportamiento observado, principalmente en el proceso que

transcurría desde que un alumno tenía errores en el programa hasta que lo solucionaba utilizando diferentes estrategias. La comunicación entre estudiantes ya sea para expresar las soluciones que habían escogido, ayudar a otros, o solicitar ayuda, siendo uno de los comportamientos principales en la resolución de problemas.

Encontramos un ejemplo parecido en un estudio donde se comparaban dos entornos de programación diferentes (programación con robots tangibles y programación de robots en un entorno virtual) donde se analizaron aspectos como la capacidad de colaboración, los diferentes tipos de interacciones, así como la forma de improvisar en la programación. Mientras en el entorno físico que permitía manipular la programación del robot la colaboración entre los alumnos era más favorable puesto que los alumnos tomaban las decisiones juntos, y mientras uno podía decidir las instrucciones del robot, el otro las ejecutaba, el entorno virtual no ofrecía esa capacidad de colaboración puesto que tenían que negociar juntos tanto la planificación como la ejecución. En cuanto a la capacidad de improvisar y rectificar en tiempo real la programación de los robots tras comprobar el resultado de una programación previa, ambos sistemas permitían esta improvisación, si bien es cierto que en el entorno virtual el feedback era más rápido (Burlison et al., 2017).

En otro artículo se analiza el grado de interés hacia el aprendizaje de programación mediante herramientas visuales, donde se concluye que los resultados de ese interés son igual de favorables antes que después de la intervención en las 4 categorías analizadas: motivación, desafío, competición y atención general (Asad et al., 2016).

Cabe destacar también que se encuentra un artículo donde se realiza una intervención donde se usan herramientas relacionadas con el pensamiento computacional para la resolución de problemas donde también se introdujeron actividades relacionadas para trabajar el constructo de Inteligencia Emocional (Rafique et al., 2020).

En cuanto a otras variables que se han tenido en cuenta como interesantes para este estudio podemos destacar aquellas experiencias que han buscado la mejora docente:

- a. Diseño de un trayecto formativo para crear Tecnologías de Inclusión Social, donde se propuso formar tanto a docentes en servicio como en formación, en los conceptos básicos de la programación, desde propuestas contextualizadas y ejemplos significativos en un marco comunitario y colaborativo de reflexión e intercambio (Monjelat, 2019)
- b. Programas de mejora de capacitación de docentes en el área de computación (Kong et al., 2020)

- c. Propuestas de innovación en el área de música de los estudiantes de Magisterio, donde se fomenta la creatividad y la innovación en la elaboración de proyectos musicales con herramientas de programación como Scratch y Makey Makey, promoviendo a su vez la utilización de tecnologías digitales para la creación de materiales didácticos (Castro et al., 2020)
- d. Programas de mejora curricular en el área de Ciencias incluyendo actividades como el diagrama de flujo o los mapas conceptuales para la resolución de problemas del día a día (Mensan et al., 2020)
- e. Integración en el currículo educativo de infantil de actividades orientadas a trabajar el pensamiento computacional mediante el diseño de una serie de actividades con diferentes clases de robots (Hunsaker y West, 2020). Evaluar los recursos necesarios para incluir el pensamiento computacional de manera curricular para alumnos con algún tipo de discapacidad, donde se concluyó que esta área puede ser trabajada con alumnos con esta problemática siempre y cuando se puedan adaptar los recursos convencionales. Si lo anterior sucede y tienen un profesor que los guía, pueden alcanzar este tipo de habilidades igualmente (Snodgrass et al., 2016).

Mención especial merece la atención a la variable género donde en un artículo se llegó a la conclusión que después de una intervención de una semana con pensamiento computacional, donde al comienzo ambos géneros pensaban que el otro género programaba peor y donde al finalizar las chicas se sentían menos capaces que en un principio en programación y reparación de problemas en ordenadores, y por el contrario los chicos se veían más capaces de lo que inicialmente pensaban (Jenson y Droumeva, 2016). En otros estudios no se encontraron diferencias estadísticamente significativas en cuanto al nivel de pensamiento computacional (Relkin et al., 2020; Pila et al., 2019) y en otro los chicos demostraron tener más interés y motivación por estas materias que las chicas, pero sin apreciar diferencias de género en cuanto al nivel de pensamiento computacional (Kong et al., 2018). En otro estudio donde se pretendía analizar la eficiencia de incorporar actividades unplugged se pudo comprobar donde las chicas sacaban mucha mejor puntuación en un pretest pero que a medida que se produjo la intervención, finalmente los chicos igualaron y superaron esas puntuaciones en el mid-test y post-test respectivamente (Relkin et al., 2020).

Para finalizar este apartado, cabe destacar las aportaciones de Román-González (2017) en la creación de un test de pensamiento computacional cuyo diseño correlaciona

positivamente con factores como el verbal, el espacial, razonamiento y el numérico y procesos cognitivos como la resolución de problemas. El mismo test fue utilizado en otro estudio (Román-González et al., 2018) donde se pudo verificar la correlación entre el pensamiento computacional y otros factores como la autoeficacia, así como con alguno de los rasgos de la personalidad (Román-González, et al., 2018). Respecto a la autoeficacia, se indagaba en dos variables: autoeficacia respecto al pensamiento computacional y autoeficacia respecto al dominio de usar ordenadores. Se encontró una brecha de género en la autoeficacia respecto al pensamiento computacional favorable al género masculino, y se fue acentuando a medida que los alumnos eran más mayores. Sin embargo, no se apreciaron esas diferencias respecto a la autoeficacia con ordenadores. En cuanto a los rasgos de la personalidad se buscaron correlación con las cinco dimensiones de la personalidad del modelo Big Five (Barbaranelli et al., 2003): sinceridad, extroversión, amabilidad, inestabilidad emocional y escrupulosidad, y se encontró correlación positiva en los rasgos de la personalidad como la sinceridad, extroversión y escrupulosidad respecto a los resultados del test de pensamiento computacional. Según este estudio, aquellos estudiantes con resultados altos en test de pensamiento computacional se correspondían con rasgos como la sinceridad y escrupulosidad mientras que los que sacaban peores resultados tenían puntuaciones bajas a su vez en extroversión, resultado este que se contradice con estereotipos donde los programadores son seres con baja capacidad de extroversión y no les gusta la interacción social (Schott y Selwing, 2000).

5 - Discusión

El pensamiento computacional es un campo de investigación relativamente nuevo que ha sido explorado por investigadores especialmente en la última década. A través de esta revisión sistemática se ha realizado un mapeo orientado a medir el impacto de las intervenciones educativas con pensamiento computacional en las etapas educativas de infantil y primaria. Concretamente, se han seleccionado artículos que no solo hayan tenido la inclusión del pensamiento computacional en el aula, sino que además incluyan elementos que nos permitan indagar en cómo se ha evaluado dicha intervención, así como el qué y el cómo se ha evaluado. También ha sido de máximo interés en esta investigación indagar en la relación del pensamiento computacional con otros constructos y aspectos relacionados con el proceso de Enseñanza-Aprendizaje. Los datos más relevantes detectados en este trabajo de investigación son expuestos a continuación en aras de mantener actualizado el estado de la cuestión, así como para realizar algunas recomendaciones para futuras líneas de investigación que puedan solventar las brechas detectadas.

En cuanto a la forma de evaluar, sin poder tener una herramienta que sea capaz de medir esta capacidad, ¿cómo seremos capaces de poder nivelar los conocimientos y aptitudes que deberán aprender nuestros alumnos? O ¿cómo seremos capaces de comprobar si el trabajo realizado en nuestras aulas tiene especial significación para ellos en cuanto a la hora de mejorar sus habilidades del pensamiento? Los estudios incluidos en esta revisión sistemática han incluido diferentes formas de evaluar dicho impacto educativo, siendo las más utilizadas las técnicas cuantitativas como pueden ser los test y los cuestionarios. Se han presentado estudios donde han validado una herramienta para medir el nivel de pensamiento computacional desde diferentes perspectivas computacionales en los últimos años y esto ha supuesto un impulso definitivo para poder medir nuestras intervenciones educativas en el aula (Román-González, 2016; Relkin et al., 2020). Por el contrario, han sido escasas las muestras representativas de análisis más cualitativos a través de estudios de caso, entrevistas, observación o diarios de campo. Una bibliografía más rica en este tipo de estudios, por la propia naturaleza de los mismos, nos aportaría un conocimiento más profundo de la forma de integrar estos conceptos en los estudiantes, así como nos ayudaría a mejorar la forma en la que se llevan al aula y nos ayudaría también a comprender esos procesos cognitivos asociados (Snodgrass et al., 2016). Una propuesta como futura línea de trabajo sería la de utilizar una combinación de ambos métodos de investigación para poder triangular mejor los resultados obtenidos y de esta forma ofrecer una comprensión más profunda del proceso de aprendizaje de los estudiantes.

En cuanto a las etapas educativas donde se han realizado estas intervenciones, la gran mayoría se han realizado en edades comprendidas en la etapa de educación primaria (6-11 años) y escasas encontradas en la etapa de infantil. Esta escasa presencia en Infantil puede ser debida a que el pensamiento computacional es aún un constructo joven que ha sido medido especialmente en etapas muy superiores porque puede estar asociado a una madurez cognitiva. Zapata-Ros (2015), argumenta que, por ejemplo, el pensamiento abstracto, una de las principales habilidades relacionadas con el pensamiento computacional, tiene mucho que ver con la edad y que es producto del desarrollo cognitivo, siendo relevante incluirlo en estas etapas mediante un diseño curricular que facilite el desarrollo intelectual de los estudiantes. La realidad que nos aportan las experiencias seleccionadas en esta etapa ofrece datos relevantes que muestran la importancia y el impacto que puede tener incluirlo para trabajar por ejemplo las direcciones o las secuencias. Por lo tanto, se muestra necesario enriquecer la literatura sobre inclusión del pensamiento computacional y su correspondiente evaluación en la etapa de educación infantil, para de esta forma enriquecer la visión de los educadores y el propio currículum educativo. Es importante destacar que se muestran también necesarias más experiencias de inclusión y evaluación del pensamiento computacional en contextos educativos informales. Grover y Pea (2013) sugieren que contextos informales tales como makerspaces o talleres DIY (“hazlo tu mismo”) juegan un papel importante en implementar una educación en pensamiento computacional. Pocas han sido las experiencias encontradas en esta revisión sistemática para evaluar su impacto y por lo tanto se sugiere como posible línea de investigación futura.

En la misma línea que lo aportado por Tang (2020), en esta revisión se han encontrado pocas experiencias relativas a procesos formativos de docentes, que finalmente serán quienes lleven estos conocimientos al aula. Tal y como se comenta en el marco teórico de este trabajo, la ausencia de formación en estos conceptos a los futuros maestros en las universidades se muestra como una brecha a solucionar. Una formación no solo en conceptos computacionales sino también en la relación que tienen con otras áreas como el idioma, la música, o las matemáticas, con la que comparte unos vínculos profundos ya desde la creación de Logo como lenguaje de programación, que permitió el aprendizaje de matemáticas a través de ofrecer movimiento y direccionalidad con su famosa tortuga (Berrocoso, Sánchez y Arroyo, 2015) . Trabajar elementos cognitivos puede llevar al alumno a que se involucre más en el área. Los lleva a implicarse de otra manera para hacer las tareas más allá de la forma habitual. Si

enfocamos el trabajo en el aula hacia la resolución de problemas en nuestros desafíos de programación, el alumno, al poder comprobar que es lo que funciona y lo que no de manera inmediata, puede hacer subir su nivel de motivación y de autoeficacia, tal y como ha quedado demostrado en investigaciones aportadas en este trabajo (Kong et al., 2017).

Uno de los objetivos fundamentales de introducir el pensamiento computacional era comprobar el impacto que la inclusión del pensamiento computacional puede producir entre nuestros estudiantes, y una de las formas en las que se espera dicho impacto es en la transferencia de conocimiento hacia otras asignaturas. Tal y como se comenta en el párrafo anterior, a pesar de lo obvio que puede ser la conexión de programas para aprender a programar como Scratch con áreas como las matemáticas, los idiomas, la música o el arte, son pocas las experiencias que se han encontrado en este trabajo de investigación. Esto puede ser debido a varios factores según se ha detectado: las intervenciones educativas son realizadas por expertos en el área y no por maestros (Jenson y Droumeva, 2016), la juventud y novedad de este constructo y seguramente la falta de confianza y de formación de estos maestros a los que hay que apoyar con formaciones específicas que sirvan para actualizarse en estos contenidos (Castro et al., 2020). El objetivo último de integrar el pensamiento computacional va más allá de realizar creaciones de cualquier tipo, sino sobre todo que los alumnos puedan integrar estos constructos en su proceso de enseñanza-aprendizaje para que puedan aplicarlo a cualquier área en su vida. El hecho de que pueda ser utilizado en cualquier área en todos los niveles educativos debería ayudar a que así fuera, pero en este trabajo han sido escasas las experiencias encontradas. Por lo tanto, cultivar estas habilidades fuera de las asignaturas STEM, tal y como ya argumentaban Czerkawski y Lyman (2015) se constata que es todavía un desafío por afrontar.

Además de evaluar el impacto de las intervenciones educativas, este trabajo también buscaba indagar en el “cómo” se introducían este tipo de habilidades. En este sentido se detecta una apuesta clara por hacerlo mediante trabajos o proyectos orientados a programación, lo cual por una parte es una buena práctica para la integración de dichas habilidades, pero no la única, y por lo tanto, podríamos considerar también como una brecha metodológica el hecho de no alcanzar estos factores cognitivos de alto nivel de una manera alternativa, como pueden ser las tareas unplugged (actividades sin tecnología) que pueden ser muy beneficiosas en el ámbito de la investigación no sólo desde el punto de vista de cómo se alcanzan estas habilidades sino también cómo es el proceso de transición hacia actividades plugged (del Olmo-Muñoz et al., 2020).

Nos queda por analizar el “qué” se estaba analizando, qué constructos se trabajaban propios del pensamiento computacional y con cuales se establecía una relación contrastada. En cuanto a detectar cuales son las principales habilidades investigadas relacionadas con el pensamiento computacional, mientras que en algunos artículos se centran más en las prácticas computacionales (variables, bucles, estructuras de selección, etc) otros se centran más en habilidades cognitivas como la resolución de problemas, abstracción, descomposición, pensamiento algorítmico o evaluación. Se echa en falta una actualización del estado de cuestión donde se consiga un consenso entre los diferentes marcos existentes que definen el pensamiento computacional y sus habilidades relacionadas. Nos hemos encontrado algunos estudios donde lo prioritario son los conceptos computacionales, y en otros donde lo importante son las prácticas computacionales, sin establecer una relación con constructos, lo cual debería ser el principal foco de la cuestión: las habilidades del pensamiento a través del pensamiento computacional (Zapata-Ros, 2015). En cuanto a la relación con otros constructos de interés, dos llaman especialmente la atención: la motivación y el género. Es muy popular la creencia de que las chicas tienen menos interés que los chicos en aprender a programar, por eso ha resultado un poco sorprendente que hayan sido pocas las investigaciones las que han aportado algún dato a tal efecto. Esto seguramente pueda ser debido a que este trabajo está centrado en las etapas de infantil y primaria y donde se suele detectar esta brecha suele ser ya al final de la etapa secundaria (Jenson y Droumeva, 2016), la cual no era objeto de estudio en este trabajo. En cuanto a los datos obtenidos al respecto, cabe destacar dos cosas especialmente: tanto ellos como ellas tienen la misma capacidad en esta área, y que se ha detectado una falta de autoconfianza y de interés en las chicas, algo que no ocurre en los chicos. Sobre la motivación, queda demostrado algo ya conocido como que trabajar con nuevas tecnologías siempre motiva mucho más al alumnado, pero lo realmente motivador y quizás hasta innovador para la comunidad educativa, es que en ocasiones no son tan importantes los medios técnicos con los que cuentas, sino la metodología que un docente puede llevar al aula para conseguir sus objetivos didácticos (Mensan et al., 2020)

6 - Limitaciones

Este trabajo de investigación tiene una serie de limitaciones a considerar, la mayoría de ellas directamente relacionadas con las decisiones que se tomaron en el propio diseño de la investigación. Para asegurar una máxima calidad en cualquier revisión sistemática, el procedimiento recomienda que dos (o más) sean los autores de cualquier trabajo de revisión, no siendo necesario que todos los autores trabajen en todo el proceso, pero sí especialmente en la parte de elegibilidad y en los análisis de sesgos. Sin embargo, por las propias limitaciones académicas de este trabajo de investigación, fue realizado exclusivamente por un solo autor.

Las keywords elegidas pueden suponer una limitación en sí mismas. En nuestra búsqueda en idioma español se decidió utilizar “pensamiento computacional” y en inglés el término “computational thinking” más un segundo término que hiciera mención a la parte de evaluación como “assessment” o “evaluation”. Se dejaron fuera términos relacionados como “coding”, “computing” o “programming” precisamente por incidir en el aspecto relacionado con la evaluación de las intervenciones. En la misma fase del diseño, otro aspecto clave era la elección de las bases de datos para encontrar los artículos que finalmente han sido evaluados. La elección de otras bases de datos o la ampliación de las ya existentes nos habría facilitado seguramente una fuente más rica de información y nos habría aportado quizás otros artículos como por ejemplo los dos que se incluyeron manualmente (Román-González et al., 2017; Román-González et al., 2018) y que inicialmente no se correspondían a los resultados de las búsquedas en las bases de datos seleccionadas. Este trabajo de investigación ha sido realizado exclusivamente por un autor y el número de base de datos que ha podido abordar era limitado. Por esta misma razón, se acotó la búsqueda a los últimos cinco años. Siendo un constructo joven seguramente no se espera haber encontrado muchos más, pero en un estudio con más autores no solo hubiera sido factible alcanzar más bases de datos, sino también aumentar el rango de años en las búsquedas.

Otro de los aspectos clave del método fue la elección de Prisma como método a seguir para hacer la revisión sistemática. Podría merecer la pena considerar otros modelos más cuantitativos en los procesos de revisión en cuanto a las aportaciones y de esta forma incluso plantear líneas de investigación futuras como un metaanálisis desde una perspectiva más cuantitativa. De la misma forma, una de las mejoras conceptuales que incluye PRISMA en sus procesos es la evaluación del sesgo al nivel de los estudios o el de los resultados, algo que faltaría por incluir en este trabajo de investigación. Tal y como indica PRISMA, las conclusiones fiables de una investigación dependen de la validez de los datos y resultados de

los estudios incluidos en la propia revisión (Urrútia y Bonfill, 2010). La evaluación de la validez de dichos estudios será un aspecto fundamental de una revisión y como tal debería incluirse en los análisis, interpretación y en las conclusiones.

Una fase importante en la aplicación de PRISMA es la fase de screening, donde se tienen que aplicar los criterios de inclusión y exclusión. El hecho de descartar aquellos que no hayan medido el impacto de la intervención seguramente ha supuesto perder alguna experiencia valiosa para este trabajo de investigación. El haber restringido los resultados a las etapas de infantil y primaria, donde la presencia del pensamiento computacional es más escasa en comparación con la etapa de secundaria, por el mero hecho de que en esta etapa ya podemos encontrar alguna asignatura que pueda trabajar de forma específica el pensamiento computacional, seguramente nos ha privado de trabajos que haya podido aportar información también valiosa por ejemplo en cuanto a la relación con otros constructos de interés.

Finalmente, como aspecto importante a destacar es el hecho de que solo un estudio longitudinal ha sido encontrado en esta revisión. Este tipo de estudios nos ofrecen una visión más profunda sobre el impacto de las intervenciones educativas y por lo tanto son fuertemente recomendados para acotar el potencial de incluir el pensamiento computacional en las aulas.

7 - Conclusiones e implicaciones prácticas

La revisión sistemática realizada en este trabajo de investigación analiza la forma de evaluar no solo el pensamiento computacional sino el impacto de las intervenciones educativas asociadas, así como muestra datos sobre brechas y posibles mejoras en la forma de medir estas habilidades. Para ello se han seleccionado 21 artículos relevantes para su inclusión del periodo 2016-2020.

Los resultados nos muestran cómo las experiencias educativas fueron desiguales no sólo en cuanto a la etapa educativa y el contexto en el que se suceden. Se han encontrado mayor presencia en la etapa de educación primaria que en la etapa de infantil, donde por otra parte se han utilizado mayoritariamente intervenciones en modo unplugged, algo que ha demostrado no ser solo muy útil para el aprendizaje en esta área, sino que además se puede hacer sin tecnología y aplicado a cualquier área curricular, lo cual es de especial relevancia. También se han encontrado mayoría de intervenciones en entornos formales, lo cual por otra parte era de esperar al ser un requerimiento para esta investigación aportar una evidencia empírica. De la misma manera, es una constante encontrar que dichas intervenciones han estado orientadas a la realización de programas con diferentes softwares educativos con el objetivo de medir la mejora en los skills propios relacionados con el pensamiento computacional, pero escasas las aportaciones donde precisamente el trabajo de estas habilidades esté orientado a solucionar problemas del día a día o de otras asignaturas.

Alrededor del 60% de las diferentes herramientas que se han utilizado para evaluar el impacto educativo han sido los métodos más tradicionales como lo son los test y los cuestionarios, y entre ellos, el 75% han sido los test, con sus procesos de validación, los elegidos por los investigadores para medir ese impacto. Los constructos medidos con estos u otros instrumentos principalmente han sido los directamente relacionados con las habilidades cognitivas propias del pensamiento computacional como la resolución de problemas, la depuración o la abstracción, destacando también los no relacionados directamente como pueden ser la motivación, el género o la autoeficacia.

Los datos aportados en esta revisión sistemática evidencian que se ha hecho un gran trabajo en los últimos años, pero también que es necesario seguir profundizando en encontrar un marco único de referencia en cuanto a las habilidades relacionadas con el pensamiento computacional. Hay algunos que son comunes en varios artículos (Brennan y Resnick, 2012)

pero la gran mayoría tiene sus propias referencias y esto hace más complicado realizar una comparación de cuales son las habilidades más utilizadas en estos trabajos.

Partiendo de nuestro objetivo general, que era realizar una revisión sistemática sobre la forma de evaluar el impacto de la inclusión del pensamiento computacional en el aula, esperamos que este trabajo de investigación haya sido útil para los futuros lectores y que las siguientes implicaciones que derivan del mismo, puedan ser beneficiosas para toda la comunidad educativa ofreciendo nuevas vías de exploración en torno a esta área:

1. la comunidad académica y docentes en general pueden mantenerse actualizados sobre cómo evaluar el pensamiento computacional y las intervenciones aquí mencionadas puedan ser fuente de inspiración
2. los investigadores podrán posteriormente agregar información actualizada sobre los constructos más relevantes en la evaluación del pensamiento computacional, formatos de evaluación apropiados para medirlo, todo ello con la evidencia necesaria de un marco empírico
3. docentes e investigadores de diferentes disciplinas y etapas educativas puedan incrementar su colaboración para que podamos evaluar y promocionar el pensamiento computacional como una herramienta poderosa para trabajar habilidades cognitivas, así como otros constructos de interés en el proceso de enseñanza-aprendizaje de los estudiantes
4. pueda ser útil para administraciones educativas, diseñadores de currículums educativos, investigadores o docentes en general que apuesten por el potencial de la inclusión del pensamiento computacional en cualquier etapa educativa

8 - Referencias bibliográficas

- Acevedo Borrega, J. (2017). *El pensamiento computacional en la educación obligatoria. Una revisión sistemática de la literatura* (Master's thesis). Universidad de Extremadura.
- Allsop, Y. (2019). Assessing computational thinking process using a multiple evaluation approach. *International journal of child-computer interaction*, 19, 30-55.
- Armero, J. M. M., Taranilla, R. V., Somoza, J. A. G. C., & Gutiérrez, R. C. (2018). Análisis del efecto de la robótica en la motivación de estudiantes de tercero de Educación Primaria durante la resolución de tareas de interpretación de planos. *REXE: Revista de estudios y experiencias en educación*, 2(3), 163-173.
- Asad, K., Tibi, M., & Raiyn, J. (2016). Primary School Pupils' Attitudes toward Learning Programming through Visual Interactive Environments. *World journal of education*, 6(5), 20-26.
- Astrachan, O., & Briggs, A. (2012). The CS principles project. *ACM Inroads*, 3(2), 38-42.
- Barrera, Lizardo (2013). *Algoritmos y programación para la enseñanza y aprendizaje de la matemática escolar*. En SEMUR, Sociedad de Educación Matemática Uruguay (Ed.), VII Congreso Iberoamericano de Educación Matemática (pp. 6680-6687). Montevideo, Uruguay: SEMUR.
- Basogain Olabe, X., Olabe Basogain, M. Á., & Olabe Basogain, J. C. (2020). Enseñanza del pensamiento computacional. *Innovación educativa y gestión del conocimiento* (pp 135-160)
- Bellomo, R., & Bagshaw, S. M. (2006). Evidence-based medicine: classifying the evidence from clinical trials—the need to consider other dimensions. *Critical Care*, 10(5), 1-8.
- Bennett, J., Lubben, F., Hogarth, S., & Campbell, B. (2005). Systematic reviews of research in science education: rigour or rigidity?. *International Journal of Science Education*, 27(4), 387-406.
- Berrocoso, J. V., Sánchez, M. R. F., & Arroyo, M. D. C. G. (2015). El pensamiento computacional y las nuevas ecologías del aprendizaje. *Revista de Educación a Distancia (RED)*, (46).
- Bers, M. U. (2010). The TangibleK robotics program: Applied computational thinking for young children. *Early Childhood Research & Practice*, 12(2), n2.

- Bers, M. U. (2017). *Coding as a playground: Programming and computational thinking in the early childhood classroom*. Routledge.
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing computational thinking in compulsory education-Implications for policy and practice* (No. JRC104188). Joint Research Centre (Seville site).
- Bolstad, R. (2016). Game-Coding Workshops in New Zealand Public Libraries: Evaluation of a Pilot Project. *New Zealand Council for Educational Research*.
- Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017, November). Development of computational thinking skills through unplugged activities in primary school. *In Proceedings of the 12th Workshop on Primary and Secondary Computing Education*, 65-72
- Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *In Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada* (Vol. 1, p. 25).
- Burleson, W. S., Harlow, D. B., Nilsen, K. J., Perlin, K., Freed, N., Jensen, C. N., ... & Muldner, K. (2017). Active learning environments with robotic tangibles: Children's physical and virtual spatial programming experiences. *IEEE Transactions on Learning Technologies*, 11(1), 96-106.
- Castro Araya, H., Ríos Cortés, K., & Arguedas Quesada, C. (2020). Acompañamiento pedagógico del Programa de Tecnologías para el Aprendizaje [Protea]: una experiencia constructivista que aprovecha el Makey Makey y Scratch para enriquecer un curso de expresión musical. *Revista Educación*, 44(2), 364-380.
- Chaabi, H., Azmani, A., & Doderó, J. M. (2019, October). Analysis of the relationship between computational thinking and mathematical abstraction in primary education. *In Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality*, 981-986
- Chatley, R., Donaldson, A., & Mycroft, A. (2019). The next 7000 programming languages. *In Computing and Software Science*, 250-282
- Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & Education*, 109, 162-175.
- Clark, J., Rogers, M. P., Spradling, C., & Pais, J. (2013). What, no canoes? Lessons learned while hosting a Scratch summer camp. *Journal of Computing Sciences in Colleges*, 28, 204-210.

- CSTA (2011). K–12 Computer Science Standards (Level 2) [Documento en línea]. Recuperado de http://csta.acm.org/Curriculum/sub/CurrFiles/CSTA_K12_CSS.pdf
- CSTA & ISTE (2011). Operational Definition of Computational Thinking for K–12 Education [Documento en línea]. Recuperado de <http://csta.acm.org/Curriculum/sub/CurrFiles/CompThinkingFlyer.pdf>
- Crick, T., & Sentance, S. (2011, November). Computing at school: stimulating computing education in the UK. *In Proceedings of the 11th Koli Calling International Conference on Computing Education Research*, 122-123
- Czerkawski, B. C., & Lyman, E. W. (2015). Exploring issues about computational thinking in higher education. *TechTrends*, 59(2), 57-65.
- de Madrid, C. (2015). Decreto 48/2015, de 14 de mayo, del Consejo de Gobierno, por el que se establece para la Comunidad de Madrid el currículo de la Educación Secundaria Obligatoria. *Boletín oficial de la Comunidad de Madrid*, 20.
- del Mar Sánchez-Vera, M. (2019). El pensamiento computacional en contextos educativos: una aproximación desde la Tecnología Educativa/Computational Thinking in Educational Environments: An Approach from Educational Technology/El pensamiento computacional en contextos educativos: una aproximación desde la Tecnología Educativa. *Research in Education and Learning Innovation Archives (REALIA)*, (23), 24-40.
- del Estado, B. O. (2015). Real Decreto 1105/2014, de 26 de diciembre, por el que se establece el currículo básico de la Educación Secundaria Obligatoria y del Bachillerato. Ministerio de Educación, Cultura y Deporte, Madrid (3 de enero de 2015), 169-546
- del Olmo-Muñoz, J., Cózar-Gutiérrez, R., & González-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of Primary Education. *Computers & Education*, 150, 103832.
- DFE, U. (2013). National curriculum in England: computing programmes of study. Retrieved July, 16, 2014.
- Dodero, J. M., Mota, J. M., & Ruiz-Rube, I. (2017, October). Bringing computational thinking to teachers' training: a workshop review. *In Proceedings of the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality* (pp. 1-6).
- Fraillon, J., Ainley, J., Schulz, W., Friedman, T., & Gebhardt, E. (2014). *Preparing for life in a digital age: The IEA International Computer and Information Literacy Study international report* (p. 308). Springer Nature.
- Frampton, G.K., Livoreil, B. & Petrokofsky, G (2017). Eligibility screening in evidence synthesis of environmental management topics. *Environ Evid* 6, 27

- Gal, D. (2019). Coding For Fun or For the Future? RSIS Commentaries. Singapore: Nanyang Technological University. *RSIS Commentaries*, 202, 19
- García, J. D. R., León, J. M., González, M. R., & Robles, G. (2019, November). Developing computational thinking at school with machine learning: an exploration. *In 2019 International Symposium on Computers in Education (SIIE)*, 1-6
- García, J. D. R., Moreno-León, J., Román-González, M., & Robles, G. (2020). LearningML: A Tool to Foster Computational Thinking Skills through Practical Artificial Intelligence Projects. *Revista de Educación a Distancia (RED)*, 20, 63.
- Google for Education. (2014). Exploring Computational Thinking. Recuperado 15 de octubre de 2018, de Google for Education website: <https://edu.google.com/resources/programs/exploring-computational-thinking/>
- González, M. R. (2016). *Códigoalfabetización y pensamiento computacional en educación primaria y secundaria: validación de un instrumento y evaluación de programas* (Doctoral dissertation, UNED. Universidad Nacional de Educación a Distancia (España)).
- 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*, 2019, Vol. 20, n. 1, 35
- Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational researcher*, 42(1), 38-43.
- Holmboe, C., McIver, L., & George, C. (2001). Research agenda for computer science education. *Proceedings of the 13th Workshop of the Psychology of Programming Interest Group*, 207-233.
- Hsu, Y. C., Irie, N. R., & Ching, Y. H. (2019). Computational thinking educational policy initiatives (CTEPI) across the globe. *TechTrends*, 63(3), 260-270.
- Hunsaker, E., & West, R. E. (2020). Designing Computational Thinking and Coding Badges for Early Childhood Educators. *TechTrends*, 64(1), 7-16.
- Jenson, J., & Droumeva, M. (2016). Exploring media literacy and computational thinking: A game maker curriculum study. *Electronic Journal of e-Learning*, 14(2), 111-121
- Kafai, Y. B., & Burke, Q. (2014). *Connected code: why children need to learn programming*. MIT Press.
- Kakavas, P., & Ugolini, F. C. (2019). Computational thinking in primary education: a systematic literature review. *Research on Education and Media*, 11(2), 64-94.

- Keller, J. M. (1983). Motivational design of instruction. *Instructional design theories and models: An overview of their current status, 1(1983)*, 383-434.
- Kong, S. C., Abelson, H., Sheldon, J., Lao, A., Tissenbaum, M., Lai, M., Lang, K., & Lao, N. (2017). *Curriculum activities to foster primary school students' computational practices in block-based programming environments*. In S. C. Kong, J. Sheldon & K. Y. Li (Eds.), *Conference Proceedings of International Conference on Computational Thinking Education*, 178-189.
- Kong, S. C., Chiu, M. M., & Lai, M. (2018). A study of primary school students' interest, collaboration attitude, and programming empowerment in computational thinking education. *Computers & Education*, 127, 178-189.
- Kong, S. C., & Abelson, H. (2019). *Computational thinking education*. Springer Nature
- Kong, S. C., Lai, M., & Sun, D. (2020). Teacher development in computational thinking: Design and learning outcomes of programming concepts, practices and pedagogy. *Computers & Education*, 151, 103872.
- Labusch, A., & Eickelmann, B. (2017). Computational thinking as a key competence—A research concept. In S. C. Kong, J. Sheldon & K. Y. Li (Eds.), *Conference Proceedings of International 5 Computational Thinking Processes and Their Congruence. Conference on Computational Thinking Education*, 103–106
- Lugo, M. J. R., & Olabe, X. B. (2018). Diseñando un material educativo digital: nuevas formas de enseñar habilidades del pensamiento computacional. *Teknologia berrien erabilera eta gaur egungo Hezkuntza joerak Uso de nuevas tecnologías y tendencias actuales*, 81.
- Lugo, M. J. R., Olabe, X. B., & Niño, N. M. (2018). “Evolution”: Design and Implementation of Digital Educational Material to Strengthen Computational Thinking Skills. *IEEE Revista Iberoamericana de Tecnologías Del Aprendizaje*, 13(1), 37-45.
- Manterola, C., Astudillo, P., Arias, E., Claros, N., & MINCIR, G. (2013). Revisión sistemática de la literatura. Qué se debe saber acerca de ellas. *Cirugía española*, 91(3), 149-155.
- MEFP, I. CNIIE. (2018). Programación, robótica y pensamiento computacional en el aula. Situación en España y propuesta normativa (MEFP y INTEF, Eds.). Recuperado de <http://code.intef.es/wp-content/uploads/2018/10/Ponencia-sobre-Pensamiento-Computacional.-Informe-Final.pdf>
- Mensan, T., Osman, K., & Majid, N. A. A. (2020). Development and Validation of Unplugged Activity of Computational Thinking in Science Module to Integrate Computational

- Thinking in Primary Science Education. *Science Education International*, 31(2), 142-149.
- Moher, D., Cook, D. J., Eastwood, S., Olkin, I., Rennie, D., & Stroup, D. F. (2000). Improving the quality of reports of meta-analyses of randomised controlled trials: the QUOROM statement. *Oncology Research and Treatment*, 23(6), 597-602.
- Moher, D., Liberati, A., Tetzlaff, J., Altman, D. G., & Prisma Group. (2009). Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement. *Physical therapy*, 89(9), 873-880.
- Moher, D., Liberati, A., Tetzlaff, J., Altman, D. G., & PRISMA, G. (2014). Ítems de referencia para publicar revisiones sistemáticas y metaanálisis: la Declaración PRISMA. *Revista Española de Nutrición Humana y Dietética*, 18(3), 172-181.
- Monjelat, N. (2019). Programación de tecnologías para la inclusión social con Scratch: Prácticas sobre el pensamiento computacional en la formación docente. *Revista Electrónica Educare*, 23(3), 182-206.
- Moreno-León, J., Robles, G., & Román-González, M. (2015). Dr. Scratch: Automatic analysis of scratch projects to assess and foster computational thinking. *RED. Revista de Educación a Distancia*, (46), 1-23.
- Moreno-León, J., Robles, G., & Román-González, M. (2016). Code to learn: Where does it belong in the K-12 curriculum. *Journal of Information Technology Education: Research*, 15, 283-303.
- Moreno-Léon, J., Robles, G., & Román-González, M. (2017). Programar para aprender en educación primaria y secundaria: ¿qué indica la evidencia empírica sobre este enfoque. *Revista de investigación en docencia universitaria de la informática*, 10(2).
- Moreno-León, J., Román-González, M., Hartevelt, C., & Robles, G. (2017, May). On the automatic assessment of computational thinking skills: A comparison with human experts. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (pp. 2788-2795).
- Moreno-León, J., Román-González, M., & Robles, G. (2018, April). On computational thinking as a universal skill: A review of the latest research on this ability. In *2018 IEEE Global Engineering Education Conference (EDUCON)* (pp. 1684-1689). IEEE.
- Moschovakis, Y. N. (2001). What is an algorithm?. In *Mathematics unlimited—2001 and beyond* (pp. 919-936). Springer, Berlin, Heidelberg.

- Muñoz-Repiso, A. G. V., & González, Y. A. C. (2019). Robótica para desarrollar el pensamiento computacional en Educación Infantil. *Comunicar: Revista científica iberoamericana de comunicación y educación*, (59), 63-72.
- Olabe, X. B., Basogain, M. Á. O., & Basogain, J. C. O. (2015). Pensamiento Computacional a través de la Programación: Paradigma de Aprendizaje. *Revista de Educación a Distancia (RED)*, (46)
- Olabe, X. B., & Parco, M. E. O. (2020). Integración de Pensamiento Computacional en Educación Básica. *Dos Experiencias Pedagógicas de Aprendizaje Colaborativo online. Revista de Educación a Distancia (RED)*, 20(63).
- P. Rose, S., Habgood, M. J., & Jay, T. (2020). Designing a programming game to improve children's procedural abstraction skills in scratch. *Journal of Educational Computing Research*, 58(7), 1372-1411.
- Page, M. J., Moher, D., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., ... & McKenzie, J. E. (2021). *PRISMA 2020 explanation and elaboration: updated guidance and exemplars for reporting systematic reviews. bmj*, 372.
- Pea, R. D., & Kurland, D. M. (1984). On the cognitive effects of learning computer programming. *New ideas in psychology*, 2(2), 137-168.
- Pila, S., Aladé, F., Sheehan, K. J., Lauricella, A. R., & Wartella, E. A. (2019). Learning to code via tablet applications: *An evaluation of Daisy the Dinosaur and Kodable as learning tools for young children. Computers & Education*, 128, 52-62.
- Rafique, M., Hassan, M. A., Jaleel, A., Khalid, H., & Bano, G. (2020). A Computation Model for Learning Programming and Emotional Intelligence. *IEEE Access*, 8, 149616-149629.
- Relkin, E., & Bers, M. U. (2019). Designing an assessment of computational thinking abilities for young children. *STEM for early childhood learners: how science, technology, engineering and mathematics strengthen learning. New York: Routledge*. <https://doi.org/10.4324/9780429453755-5>.
- Relkin, E., de Ruiter, L., & Bers, M. U. (2020). TechCheck: Development and validation of an unplugged assessment of computational thinking in early childhood education. *Journal of Science Education and Technology*, 29, 482-498.
- Rico, M. J., & Olabe, X. B. (2018). Pensamiento computacional: rompiendo brechas digitales y educativas. *Edmetic*, 7(1), 26-42.
- Román-Gonzalez, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2015, October). Test de Pensamiento Computacional: diseño y psicometría general. *In III Congreso*

- Internacional sobre Aprendizaje, Innovación y Competitividad (CINAIC 2015)* (pp. 1-6).
- Román González, M. (2016). *Codigofabetización y pensamiento computacional en educación primaria y secundaria: validación de un instrumento y evaluación de programas*. Doctoral dissertation, UNED. Universidad Nacional de Educación a Distancia .
- Román-González, M., Moreno-León, J., & Robles, G. (2017, July). Complementary tools for computational thinking assessment. *In Proceedings of International Conference on Computational Thinking Education (CTE 2017)*, S. C Kong, J Sheldon, and K. Y Li (Eds.). *The Education University of Hong Kong* (pp. 154-159).
- Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in human behavior*, 72, 678-691.
- Román-González, M., Pérez-González, J. C., Moreno-León, J., & Robles, G. (2018). Extending the nomological network of computational thinking with non-cognitive factors. *Computers in Human Behavior*, 80, 441-459.
- Román-González, M., Pérez-González, J. C., Moreno-León, J., & Robles, G. (2018). Can computational talent be detected? Predictive validity of the Computational Thinking Test. *International Journal of Child-Computer Interaction*, 18, 47-58.
- Royal Society (Great Britain). (2012). *Shut down or restart?: The way forward for computing in UK schools*. Royal Society.
- Rushkoff, D. (2010). *Program or be programmed: Ten commands for a digital age*. Or Books.
- Schott, G., & Selwyn, N. (2000). Examining the “male, antisocial” stereotype of high computer users. *Journal of Educational Computing Research*, 23(3), 291-303.
- ScratchEd Team [Portal Web] (2015). Computational Thinking webinars. Recuperado 2 de Junio de 2015, de <http://scratched.gse.harvard.edu/content/1488>
- Smith, M. (2016). Computer Science For All. obama white house. Recuperado de 30 Enero de 2016, de <https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all>
- Snodgrass, M. R., Israel, M., & Reese, G. C. (2016). Instructional supports for students with disabilities in K-5 computing: Findings from a cross-case analysis. *Computers & Education*, 100, 1-17.
- Stošić, L. & Bogdanović, M. (2013). M-learning – A new form of learning an education. *International Journal of Cognitive Research in Science, Engineering and Education (IJCRSEE)*, 1(2), 114–118

- Thurstone, L. L., & Thurstone, T. G. (1938). *Primary mental abilities* (Vol. 119). Chicago: University of Chicago Press.
- Urrútia, G., & Bonfill, X. (2010). Declaración PRISMA: una propuesta para mejorar la publicación de revisiones sistemáticas y metaanálisis. *Medicina clínica*, 135(11), 507-511.
- 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. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717-3725.
- Wing, J. M. (2017). Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology*, 25(2), 7–14.
- Yadav, A., Gretter, S., Good, J., & McLean, T. (2017a). Computational thinking in teacher education. In P. Rich & C. B. Hodges (Eds.), *Emerging research, practice, and policy on computational thinking* (pp. 205–220). Springer Publishing Company.
- Zapata-Ros, M. (2015). Pensamiento computacional: Una nueva alfabetización digital. *Revista de Educación a Distancia (RED)*, (46).
- Zapata-Ros, M. (2018). Pensamiento computacional. Una tercera competencia clave. El pensamiento computacional como una nueva alfabetización en las culturas digitales. *Murcia: Universidad de Murcia*, 4-87.
- Zapata-Ros, M. (2019). Computational thinking unplugged. *Education in the Knowledge Society*, 20, 1-29.
- Zapata-Ros, M. (2020). El pensamiento computacional, una cuarta competencia clave planteada por la nueva alfabetización. *Educación y Tecnología*.
- Zhang, L., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers & Education*, 141, 103607.

Anexo I. Artículos seleccionados para la revisión sistemática

- Allsop, Y. (2019). Assessing computational thinking process using a multiple evaluation approach. *International journal of child-computer interaction*, 19, 30-55.
- Armero, J. M. M., Taranilla, R. V., Somoza, J. A. G. C., & Gutiérrez, R. C. (2018). Análisis del efecto de la robótica en la motivación de estudiantes de tercero de Educación Primaria durante la resolución de tareas de interpretación de planos. *REXE: Revista de estudios y experiencias en educación*, 2(3), 163-173.
- Asad, K., Tibi, M., & Raiyn, J. (2016). Primary School Pupils' Attitudes toward Learning Programming through Visual Interactive Environments. *World journal of education*, 6(5), 20-26.
- Burleson, W. S., Harlow, D. B., Nilsen, K. J., Perlin, K., Freed, N., Jensen, C. N., ... & Muldner, K. (2017). Active learning environments with robotic tangibles: Children's physical and virtual spatial programming experiences. *IEEE Transactions on Learning Technologies*, 11(1), 96-106.
- Castro Araya, H., Ríos Cortés, K., & Arguedas Quesada, C. (2020). Acompañamiento pedagógico del Programa de Tecnologías para el Aprendizaje [Protea]: una experiencia constructivista que aprovecha el Makey Makey y Scratch para enriquecer un curso de expresión musical. *Revista Educación*, 44(2), 364-380.
- Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & Education*, 109, 162-175.
- del Olmo-Muñoz, J., Cózar-Gutiérrez, R., & González-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of Primary Education. *Computers & Education*, 150, 103832.
- Hunsaker, E., & West, R. E. (2020). Designing Computational Thinking and Coding Badges for Early Childhood Educators. *TechTrends*, 64(1), 7-16.
- Jenson, J., & Droumeva, M. (2016). Exploring media literacy and computational thinking: A game maker curriculum study. *Electronic Journal of e-Learning*, 14(2), 111-121
- Kong, S. C., Chiu, M. M., & Lai, M. (2018). A study of primary school students' interest, collaboration attitude, and programming empowerment in computational thinking education. *Computers & Education*, 127, 178-189

- Kong, S. C., Lai, M., & Sun, D. (2020). Teacher development in computational thinking: Design and learning outcomes of programming concepts, practices and pedagogy. *Computers & Education, 151*, 103872.
- Mensan, T., Osman, K., & Majid, N. A. A. (2020). Development and Validation of Unplugged Activity of Computational Thinking in Science Module to Integrate Computational Thinking in Primary Science Education. *Science Education International, 31(2)*, 142-149.
- Monjelat, N. (2019). Programación de tecnologías para la inclusión social con Scratch: Prácticas sobre el pensamiento computacional en la formación docente. *Revista Electrónica Educare, 23(3)*, 182-206.
- Muñoz-Repiso, A. G. V., & González, Y. A. C. (2019). Robótica para desarrollar el pensamiento computacional en Educación Infantil. *Comunicar: Revista científica iberoamericana de comunicación y educación, (59)*, 63-72.
- P. Rose, S., Habgood, M. J., & Jay, T. (2020). Designing a programming game to improve children's procedural abstraction skills in scratch. *Journal of Educational Computing Research, 58(7)*, 1372-1411.
- Pila, S., Aladé, F., Sheehan, K. J., Lauricella, A. R., & Wartella, E. A. (2019). Learning to code via tablet applications: *An evaluation of Daisy the Dinosaur and Kodable as learning tools for young children. Computers & Education, 128*, 52-62.
- Rafique, M., Hassan, M. A., Jaleel, A., Khalid, H., & Bano, G. (2020). A Computation Model for Learning Programming and Emotional Intelligence. *IEEE Access, 8*, 149616-149629.
- Relkin, E., de Ruiter, L., & Bers, M. U. (2020). TechCheck: Development and validation of an unplugged assessment of computational thinking in early childhood education. *Journal of Science Education and Technology, 29*, 482-498.
- Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in human behavior, 72*, 678-691.
- Román-González, M., Pérez-González, J. C., Moreno-León, J., & Robles, G. (2018). Extending the nomological network of computational thinking with non-cognitive factors. *Computers in Human Behavior, 80*, 441-459.
- Snodgrass, M. R., Israel, M., & Reese, G. C. (2016). Instructional supports for students with disabilities in K-5 computing: Findings from a cross-case analysis. *Computers & Education, 100*, 1-17.