



pensamiento computacional

estado del arte

2018 - marzo 2019

FSED_2_2018_1_150624
Fondo Sectorial “Inclusión Digital:
Educación con Nuevos
Horizontes” - 2018



**AGENCIA NACIONAL
DE INVESTIGACIÓN
E INNOVACIÓN**



pensamiento computacional
estado del arte
2018 - marzo 2019



AGENCIA NACIONAL
DE INVESTIGACIÓN
E INNOVACIÓN

pensamiento computacional estado del arte 2018 - marzo 2019

revisión de la literatura:

Guadalupe Vadillo y Jackeline Bucio

con el apoyo de: Esther Angeriz Pampin, Cristina Cárdenas, Juan José Goyeneche, Emiliano Pereiro y Elena Tarditi.

diseño editorial:

Ana María Romero Velázquez

Imágenes de Pixabay

participantes e instituciones:

Esther Angeriz Pampin

*Académica de la Universidad de la República/ Facultad de Psicología - UDeLaR
Instituto de Psicología, Educación y Desarrollo Humano / Uruguay*

Jackeline Bucio

Subdirectora de innovación académica, B@UNAM, CUAED, Universidad Nacional Autónoma de México

Cristina Cárdenas

Asesora independiente

Juan José Goyeneche

*Académico, Universidad de la República de Uruguay - UdeLaR.
Instituto de Estadística*

Emiliano Pereiro

Jefe de Pensamiento Computacional en Plan Ceibal, Uruguay

Elena Tarditi

Coordinadora de innovación y servicios, Paraguay Educa, Paraguay

Guadalupe Vadillo

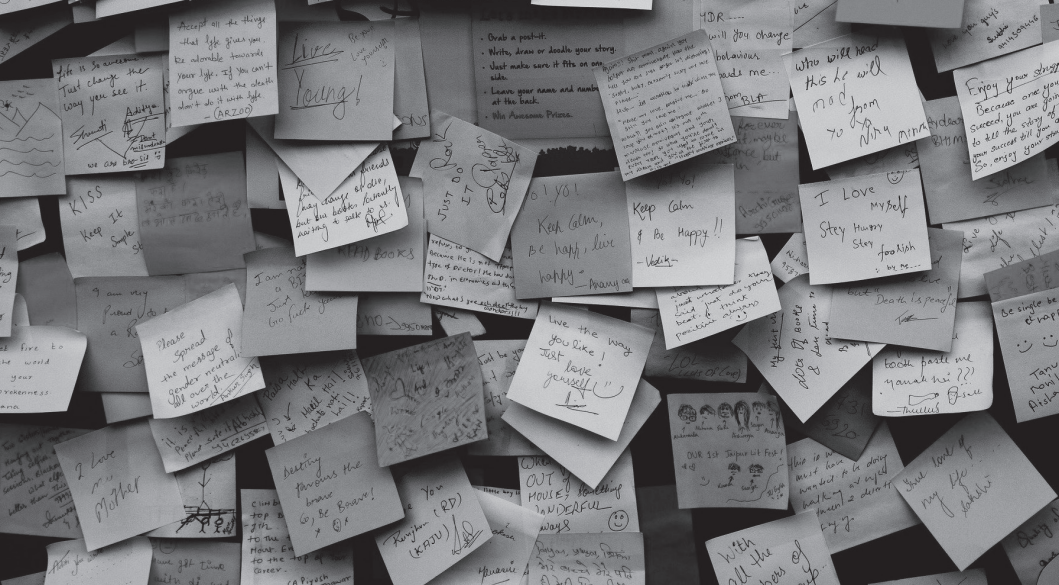
Directora de B@UNAM, CUAED, Universidad Nacional Autónoma de México

Ciudad de México, febrero 2020

Se prohíbe la reproducción, el registro o la transmisión parcial o total de esta obra por cualquier medio impreso, mediático, fotomecánico, magnético o cualquier otro existente o por existir, sin el permiso previo del titular de los derechos correspondientes.

índice

I. Método	5
II. Estado del arte	7
III. Herramientas para evaluar el PC	17
IV. Propuestas para potenciar el PC	20
V. Formación docente PC	31
VI. Resultados de investigación	34
VII. Retos y tendencias	36
VIII. Otros factores	38
Referencias	41



I. Método

a. Términos de la búsqueda

Para el presente estado del arte se realizó una búsqueda sistemática en la base de datos Clarivate (ISI Web of Science), en su CoreCollection. Se identificaron los artículos publicados durante el año 2018 y hasta el 15 de marzo de 2019, día en que se realizó la búsqueda, que presentaran el término “computational thinking” en el título. Se recuperaron 101 resultados con la siguiente distribución por país con más de una publicación:

21 USA	6 GERMANY	4 INDIA	3 MALAYSIA	3 MEXICO	3 TAIWAN
19 SPAIN	6 ITALY	4 TURKEY	2 ARGENTINA	2 ENGLAND	2 GREECE
	5 SINGAPORE	3 BRAZIL	2 AUSTRALIA		2 LITHUAN
9 PEOPLES R CHINA	4 CHILE	3 COLOMBIA	2 AUSTRIA	2 PERU	2 SOUTH KOREA
			2 DENMARK	2 PORTUGAL	

Figura 1. Distribución por países de origen de las publicaciones resultantes de la búsqueda.

De estos artículos, la distribución por idioma fue la siguiente: 94 artículos en inglés, 5 en español, uno en portugués y uno en turco.

b. Extracción de los datos y categorías resultantes

Se identificó, de cada uno de los artículos los siguientes elementos: autores, resumen, palabras clave, propósito, método, resultados y conclusiones principales. En los casos en que había definiciones de PC citadas o aportadas por los autores en la introducción o en su revisión de la literatura, se tomó dicha información para la sección de “concepto de PC”. Con base en los elementos identificados, se sintetizó la información considerada relevante y se distribuyó en los siguientes apartados: herramientas para evaluar el PC, propuestas para potenciarlo, formación docente, resultados de investigación, retos y tendencias, así como otros factores.

Fue posible localizar en texto completo un total de 94 documentos, con los que se elaboró el presente estado del arte. En la parte final de este documento, se relacionan siete artículos no disponibles en Internet. Adicionalmente, por su falta de calidad, se eliminó un artículo (Vera, Villalba-Condori & Castro, 2018).



II. Estado del arte

a. Concepto de PC

De 1990 hasta 2019 se ha intentado definir el PC a partir de esfuerzos de una variedad de investigadores. Hoy se concluye que aún no se cuenta con una definición universalmente aceptada. A continuación se presentan los trabajos del periodo considerado, agrupados por los énfasis y enfoques que les subyacen. En este apartado se abordan los problemas para definir al PC, se presenta una línea de tiempo sobre las definiciones aportadas, las definiciones incluidas en los artículos revisados, y los elementos prevalentes en ellas. Al final de este apartado aparece una figura que sintetiza los elementos hallados en las definiciones, en el marco de la propuesta de Brennan y Resnick del 2009.

Problemas para definir al PC

Nardelli (2018) en su artículo cuestiona si es necesario el término PC, dado que no existe una definición universalmente acordada. Considera que no hay que tomarlo como un concepto nuevo y que es necesario entender a la informática tanto como un campo disciplinario como un eje transversal que contribuye a entender las demás disciplinas.

Roman-Gonzalez, Perez-Gonzalez, Moreno-Leon & Robles (2018) señalan el problema de que el PC es un constructo pobremente definido en tanto que su red nomológica no ha sido establecida. En su artículo plantean las correlaciones con factores no cognitivos (auto-eficacia y algunos factores del modelo de personalidad de los cinco grandes -*Big Five*). Encuentran

correlaciones significativas entre PC y auto-eficacia relativa a desempeño de PC, apertura a la experiencia, extraversión y responsabilidad. Presentan, finalmente, una red extendida de PC que integra factores cognitivos (que explican el 27% del PC) y no cognitivos (que explican el 24% del PC). El modelo completo se encuentra en la página 456 de dicho artículo.

Los investigadores españoles Moreno-León, Román-González & Robles (2018) sintetizan investigación reciente sobre PC y lo proponen como el énfasis que la educación debe tener, en lugar de la enseñanza de programación (asociada a la demanda creciente de empleos en este sector). Señalan que el PC se caracteriza por el pensamiento algorítmico (habilidad para formular problemas que transforman un input en un output deseado a partir de algoritmos). Destacan que estudios previos que intentan demostrar que el PC es una habilidad y un constructo en sí mismo y concluyen que es una habilidad para resolver problemas ligada a inteligencia fluida. Mencionan el estudio de Roman-Gonzalez et al. (2018), presentado arriba, que identifica que solo el 27% de la varianza del PC se explica por las cuatro principales habilidades mentales, lo que apunta a que es una entidad en sí misma. Además, hay evidencia de que el 24% de su varianza se explica por factores no cognitivos (rasgos de personalidad) lo que prueba la existencia de factores no cognitivos que deben tomarse en cuenta.

Adell, Llopis, Esteve & Novella (2018) analizan las consecuencias de no contar con una definición consensuada y las problemáticas de la inclusión de PC en el currículum español. La falta de una definición que todos acepten ha llevado a carecer de criterios sobre cómo enseñar PC en los distintos niveles educativos, cómo integrarlo a las disciplinas o si se debe considerar una actividad extraescolar, así como la forma de evaluarlo o de formar maestros. Señalan que en junio de 2018, por una moción de censura, no es claro si en España seguirá la política de introducir programación, robótica y PC en el currículum oficial. Señalan que las políticas curriculares “rápidas” están asociadas a intereses de las grandes empresas de tecnología.

Historia de las propuestas de definición del PC

Hsu, Chang y Hung (2018) señalan que el concepto se introdujo en 1990 por Papert y desde entonces su definición, enseñanza y evaluación han sido centro de estudio de diversos académicos. De hecho, una década antes Papert (1980 en Pei, Wientrop & Wilensky, 2018, p. 78) había señalado: “C computer languages that simultaneously provide a means of control over the computer and offer new and powerful descriptive languages for thinking will undoubtedly be carried into the general culture”. Wing (2006 en Wu, Looi, Liu & How, 2018, p. 735) define el PC como “resolver problemas, diseñar sistemas y comprender la conducta humana partir de conceptos fundamentales de la ciencia computacional”. En 2006 Wing señalaba que es una habilidad cotidiana

que todos requerimos y que merece de todos una actitud positiva, al incorporar el pensamiento computacional en la práctica y generar análisis lógico. En 2008 Wing señalaba que el PC constituye una forma para resolver problemas humanos y lleva al estudiante a aplicar múltiples capas de pensamiento abstracto, conduce a mayor flexibilidad al aplicar su experticia. Indicaba también que es una combinación de pensamiento matemático y pensamiento ingenieril. Yagci (2019) señala que Wing en 2016 redefinió el PC como resolver problemas cuyas soluciones se representan en una forma tal que un agente procesador de información pueda llevarlo a cabo de manera efectiva.

Esta definición de Wing prevalece en 2017 (en Kwon y Kim, 2018) y se complementa con lo siguiente: todos deben aprender PC al igual que a leer, escribir y calcular. Wing (2014 en Durak & Saritepeci, 2018) considera que para mitad del siglo XXI, el PC será parte de la formación básica junto con lectura, escritura y aritmética. Vinayakumar, Soman & Pradeep (2018) consideran que el PC constituye una práctica clave en la educación científica. Como Pei y sus colegas (2018) recuerdan, dos reuniones del National Research Council de Estados Unidos se dedicaron a analizar el espectro, naturaleza e implicaciones educativas del PC. Del informe correspondiente se derivan alrededor de 20 habilidades de nivel superior como abstracción de problemas y razonamiento heurístico. Leifheit, Jabs, Ninaus, Moeller & Ostermann (2018) señalan que el desarrollo de PC es relevante porque ayuda a los estudiantes a comprender problemas complejos y abordarlos de manera sistemática.

Kynigos & Grizioti (2018) identifican en el estudio de Noss y Hoyles de 1996 que si bien en la década de 1980 hubo interés por incorporar la programación en la educación formal, en la siguiente década la tendencia se redujo, por falta de maestros calificados y la falta de integración de contenidos de las materias. Ellos consideran que estos desafíos prevalecen actualmente.

Definiciones propuestas en el periodo analizado

En los artículos revisados hay 13 que hacen una revisión de la literatura. El primero es el de Hsu y sus colegas (2018). Presentan una clasificación amplia de PC, a partir de la revisión de la literatura que abarca de 2006 a 2017. De su análisis concluyen que el PC tiene cada vez más investigación, se utiliza sobre todo en cursos de diseño de programas, ciencia computacional, biología y robótica.

En su revisión, Lee y Recker (2018) presentan la definición del consorcio K-12 Computer Science Framework Steering Committee de 2016: estándares de ciencia computacional para estudiantes tales como el desarrollo y uso de abstracciones, la colaboración con base en cómputo, y la generación de artefactos computacionales.

Yasar (2018) plantea una perspectiva alternativa: el PC como una liga a una competencia cognitiva involucrada en ciencia, ingeniería y vida cotidiana. Lo define como un pensamiento generado y facilitado por la computación, sin importar el dispositivo en que se realiza dicha computación. Considera que debe separarse la computación electrónica de la biológica y que sólo es en la intersección que se da entre ambas que se dan los procesos de computación independiente de dispositivos. Su perspectiva es interdisciplinaria e incluye ciencia computacional, epistemología, cognición y neurociencias.

Hoppe y Werneburg (2018 en Werneburg, Manske y Hoppe, 2018, p. 543) precisan que la esencia del PC “se encuentra en la creación de ‘artefactos lógicos’ que externalizan y materializan ideas humanas en una forma que puede ser interpretada y ‘corrida’ en computadoras”.

Kusnendar & Prabawa (2018) señalan que el PC implica pensar sobre computación, al requerir que los estudiantes formulen problemas en formato computacional y aporten soluciones computacionales correctas (en forma de algoritmos) o expliquen su ausencia.

Lübbers & Jansen (2018) presentan la definición de Selby y Wollward de 2014, que señala que el PC es una actividad cerebral que permite que se resuelvan problemas, se comprendan mejor situaciones, y se expresen mejor valores a través de la aplicación sistemática de los procesos de abstracción, decomposición, diseño algorítmico, generalización y evaluación en la producción de una automatización que puede implementarse en un dispositivo digital o humano.

Sabitzer, Demarle-Meusel & Jamig (2018) rescatan la definición de 2012 de Aho: PC es diseño relativo a un modelo computacional.

Nardelli (2018), siguiendo a Wing, define al PC como “ser capaz de pensar como un científico de la computación y de aplicar esta competencia a todo campo de la actividad humana” (p. 32).

Mohanty & Das (2018) definen al PC como un proceso de pensamiento en el que se formulan problemas de tal forma que las soluciones pueden representarse como pasos y algoritmos computacionales.

Rico et al. (2018) destacan que el PC tiene como objetivo resolver problemas del siglo XXI que requieren soluciones innovadoras, creativas y que signifiquen un reto para el individuo. Requiere que el estudiante tenga habilidades para analizar y comprender los problemas, que asuma posturas en pro o en contra del tópico. Se trata de que encuentre múltiples soluciones y sea capaz de sustentarlas.

Rijke, Bollen. Eysink & Tolboom (2018) consideran que el PC puede incidir

en la eficiencia de actividades cotidianas y puede mejorar la comprensión del uso generalizado de computadoras y software.

Wang & Wang (2018) destacan las características de automatización y abstracción en el PC. Por su manejo del idioma, se trata de un artículo cuya aportación no es clara.

Tramonti & Dochshanov (2018), a partir de las definiciones de Wing de 2006 y 2011, y de las aportaciones de Bocconi y sus colegas en 2016, destacan dos componentes fundamentales: el que sea considerado como un proceso de razonamiento (y, por tanto, independiente del uso de tecnología) y como una forma de resolver problemas que requiere habilidades específicas relativas a: la definición del problema, la organización de datos y análisis lógico, el desarrollo de modelos y simulaciones para visualización de datos, y una efectiva combinación de recursos para alcanzar soluciones y aplicarlas a diferentes contextos y situaciones.

Velazquez-Iturbide (2018) analiza el concepto de PC e identifica los elementos principales de seis definiciones (s/p):

- Wing (2010), formulación de problemas y soluciones
- Aho (2011), formulación de problemas
- Brennan y Resnick (2009), conceptos prácticos y perspectivas
- CAS (2012), procesos clave
- CAS (2015), conceptos y técnicas
- CSTA & ISTE (2011) características, disposiciones y actitudes

Con objeto de visualizar las principales aportaciones, se presenta la siguiente línea de tiempo:

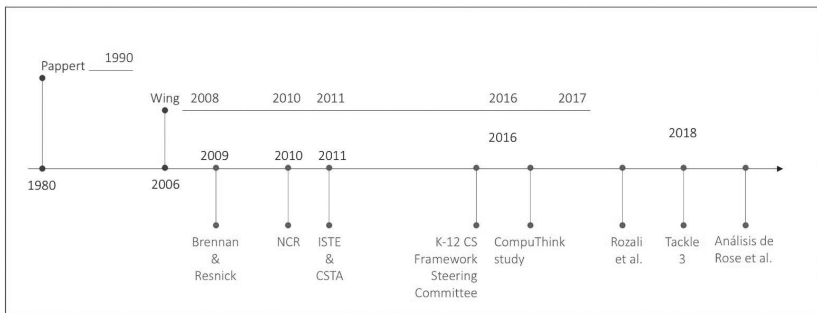


Figura 2. Línea de tiempo

Elementos prevalentes en las definiciones

Como señalan Rico, Basogain & Moreno (2018), en 2011 se reunió la Sociedad internacional para la tecnología en la educación (ISTE, por sus siglas en inglés) y la Asociación de maestros de ciencias computacionales (CSTA, por sus siglas en inglés). Definieron que el PC es un proceso de solución de problemas que incluye seis características:

- Formular problemas de manera que computadoras y otras herramientas puedan usarse para resolverlos,
- Organizar datos en formas lógicas y analizarlos,
- Representar datos a través de abstracciones (como modelos y simulaciones),
- Automatizar soluciones a través de pensamiento algorítmico,
- Identificar, analizar e implementar posibles soluciones para encontrar más eficientes combinaciones de pasos y recursos,
- Generalizar y transferir el proceso de solución de problemas a un abanico de problemas.

Kwon y Kim (2018) identificaron nueve conceptos centrales al PC a partir de las aportaciones de ISTE (International Society for Technology in Education) y de CSTA (Computer Science Teachers Association). Esos conceptos son: colección de datos, análisis de datos, representación de datos, descomposición de problemas, abstracción, algoritmos y procedimientos, automatización, simulación, y paralelización. En un estudio con estudiantes universitarios sin área de profundización en software encontraron que colección de datos y algoritmos y procedimientos constituyen los factores de mayor influencia para dichos aprendices.

En su estudio, Durak & Saritepeci (2018) encontraron que la variable tipos de pensamiento (medida por el test de Sternberg y Wagner de 1992) constituye el mayor predictor de PC, en sus elementos de pensamiento judicial, legislativo y ejecutivo. Otra variable relevante es la creatividad, así como el éxito previo en clase de matemáticas, actitud hacia la matemática, nivel educativo (con impacto negativo, quizá porque en un periodo la computación no fue obligatoria en Turquía, donde se realizó el estudio y los estudiantes de nivel bachillerato salieron bajos en PC), entre otras.

En Europa, el European Commission Science Hub ha promovido el PC como una habilidad del siglo XXI, al mismo tiempo que dicho hub incluye la competencia digital (CD) como parte de su marco 2.0. Juskeviciene y Dagienė (2018) presentan un análisis comparativo y relacional de ambos términos

y concluyen después de su análisis de la literatura que hay muchos puntos de contacto entre ellos, excepto la solución de problemas digitales de la CD que no aparece en el PC. Hubo cinco habilidades de PC que no tuvieron conexiones: identificación de abstracciones, descripción del modelado en contextos computacionales, descripción de computación con lenguaje, notaciones o visualizaciones precisos y correctos, síntesis del propósito de un artefacto computacional. El PC tiene ocho grupos de componentes:

- a. análisis de datos y representación (procesos de colección, análisis y representación de datos)
- b. artefactos computacionales (proceso de diseño y desarrollo de artefactos y aplicación de técnicas computacionales para resolver problemas creativamente)
- c. decomposición (partir una tarea en trozos manejables)
- d. abstracción (solución de un problema más general al ignorar ciertos detalles)
- e. algoritmos (diseño de algoritmos, automatización como proceso de insertar piezas en un algoritmo para obtener un resultado)
- f. comunicación y colaboración (comunicarse y trabajar con otros para lograr una solución)
- g. cómputo y sociedad (influencia del cómputo y sus implicaciones en individuos y sociedad)
- h. evaluación (identificar qué tan apropiados son las soluciones y los artefactos)

Por su parte, Park y Park (2018) proponen un marco de práctica del PC (*Computational thinking practice framework*) a través de la identificación de componentes de PC. Identifican dos etapas, la primera es de práctica directa que se refiere a la formación de conceptos de ciencia a través de los componentes: recolección de datos, análisis de datos y representación de datos. Se trata de prácticas cognitivas y físicas (porque los aprendices usan computadora y equipo). La segunda es de aplicación indirecta, sólo con práctica cognitiva, es decir se expresa a través de dibujos o diagramas y se logra a través de componentes como decomposición, abstracción, algoritmos y procedimiento, automatización, simulación y generalización.

Rozali, Zaid, Noor & Ibrahim (2018) proponen un modelo unificado para la enseñanza de PC. Utilizan el mismo marco de Palts y Pedaste de 2015, al clasificar el acercamiento al PC en concepto, práctica y perspectiva. Sin embargo el marco de Palts y Pedaste corresponde a aprendizaje, mientras el suyo, a enseñanza. Realizaron una revisión sistemática de la literatura que los

llevó a doce artículos. Concluyen que los *conceptos de PC* se refieren a una dimensión que percibe cómo los estudiantes aprenden los aspectos técnicos de la programación y solo encontraron dos artículos que la usan (el de Borges y colegas en 2017 y el de Missiroli, Russo y Ciancarini de 2017 que enfatiza la liga de PC con metodologías ágiles que implican iteraciones sucesivas que se van acercando a la solución). En cuanto a las prácticas de PC, lo encontraron en 50% de los artículos revisados y se refiere a la identificación del problema, la descomposición, abstracción, prueba y depuración. La tercera dimensión, *perspectivas de PC*, se refiere a la comprensión al resolver un problema críticamente, y está referido en cuatro de los artículos. A partir de su análisis, identificaron tres referentes que promueven el PC: la taxonomía de Bloom y PC de Burbaite, Drasute y Stuijks en 2018; el Proceso CICP en la fase de ideación, de Sharifah et al. en 2017; y el Modelo de cultivo de PC basado en programación visual de Xiao y Yu en 2017. A partir de ellos los autores proponen que un modelo unificado incluya los elementos de estos tres modelos previos con todos sus elementos: abstracción, descomposición, reconocimiento de patrones, representación de datos, algoritmos y evaluación de soluciones.

Ioannou & Makridou (2018) citan a Grover y Pea de 2013 que consideran que PC y literacidad computacional son términos equivalentes. Señalan que una de las definiciones operacionales de PC más citadas es la de Brennan y Resnick: “un dispositivo para conceptualizar el aprendizaje y desarrollo con las siguientes características: (i) formular problemas de forma que permite que usemos una computadora y otras herramientas para resolverlos; (ii) organizar y analizar datos de forma lógica; (iii) representar datos a través de abstracciones como modelos y simulaciones; (iv) automatizar soluciones a través de pensamiento algorítmico (una serie de pasos ordenados); (v) identificar, analizar e implementar posibles soluciones con el fin de lograr la más eficiente y efectiva combinación de pasos y recursos; y (vi) generalizar y transferir este proceso de solución de problemas a una amplia variedad de problemas”.

García-Penalvo y Mendes (2018) presentan una recolección de definiciones y elementos de PC, incluyendo entre estos elementos el TACCLE 3 Coding European project que incluye tres dimensiones: habilidad de interpretar fenómenos como acciones computacionales (programar), habilidad de realizar acciones computacionales para resolver problemas (pensamiento lógico) y habilidad para diseñar y controlar tareas de control autónomo (control de tecnología). García-Penalvo (2018a) destaca que la enseñanza centrada en que el niño o joven sea un buen usuario de computadoras es insuficiente ya que la rápida obsolescencia de productos hace que el tiempo invertido en su manejo no sea productivo. Es necesario que se potencia el PC.

Tsarava, Moeller, & Ninaus (2018) destacan las coincidencias conceptuales de PC como habilidad cognitiva con programación como una habilidad práctica. Señalan que conceptos centrales a la programación son las ideas genéricas de secuenciación, bucle (l oop), paralelismo, evento, condicional, operador, variables y datos. Por su parte el PC se alimenta de procesos como pensamiento algorítmico, lógica condicional, descomposición, abstracción, emparejamiento de patrones, paralelización, evaluación y generalización.

Rose, Habgood & Jay (2018) destacan la definición de Shute, Sun y Asbell-Clarke de 2017 que establece que el PC es el basamento conceptual que se requiere para resolver problemas de forma eficiente y efectiva, (con o sin apoyo de computadoras) con soluciones que pueden reusarse en distintos contextos. Señalan que no hay una definición totalmente aceptada, pero que su análisis arroja coincidencias en las siguientes áreas:

- Abstracción y generalización
- Reconocimiento de patrones
- Algoritmos y procedimientos
- Recolección, análisis y representación de datos
- Descomposición
- Paralelismo
- Depuración, prueba y análisis para identificar, remover y corregir errores
- Estructuras de control usando enunciados condicionales e iteraciones

Síntesis

La siguiente figura pretende sintetizar los elementos que prevalecen en las diferentes definiciones presentadas:

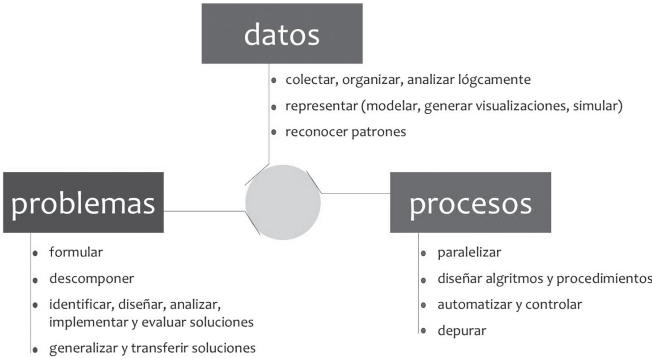
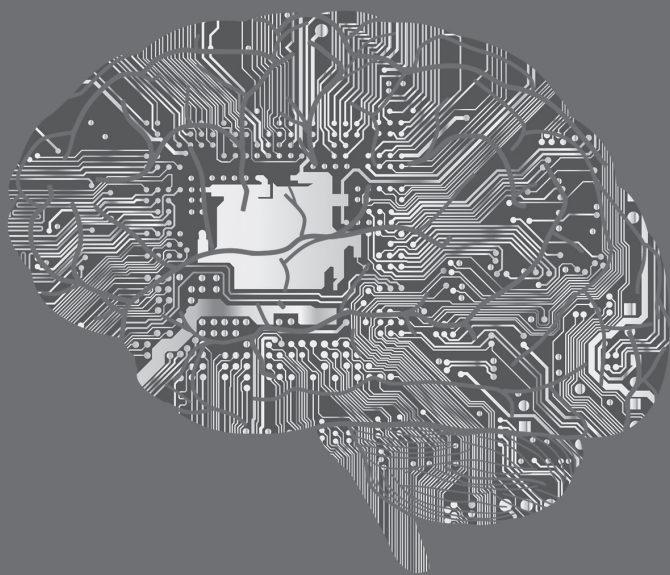


Figura 3. Elementos identificados de manera frecuente en las definiciones, en el marco del modelo propuesto por Brennan y Resnick en 2009 (elaboración propia).



III. Herramientas para evaluar el PC

En esta sección se presentan diversas herramientas presentadas en los artículos revisados. Se agrupan en las siguientes categorías. Tests y rúbricas; cursos, actividades y concursos; robots y apps; y un modelo para evaluar el PC.

Tests y rúbricas:

- a. Yagci (2019) presenta un test de 42 reactivos para evaluar PC en estudiantes de bachillerato. Tiene cuatro subescalas, resultantes del análisis factorial: solución de problemas; aprendizaje cooperativo y pensamiento crítico; pensamiento creativo; y pensamiento algorítmico. Resultó válido y confiable. Es de autoreporte. El artículo incluye los reactivos.
- b. Debido a que en Malasia se estableció como obligatoria la promoción del PC en todas las asignaturas de primaria 1, evaluar su desarrollo es una prioridad. Ling, Saibin, Naharu, Labadin & Azis (2018) desarrollaron una rúbrica para tal efecto, la enviaron a 30 escuelas e hicieron un piloto con nueve profesores que la aplicaron al inicio y término del año escolar. Los profesores completaron un cuestionario para evaluar sus percepciones sobre la rúbrica (mismo que no se incluye en el artículo).

- c. Rojas-Lopez & Garcia-Penalvo (2018a), de la Universidad Tecnológica de Puebla, presentan una serie de ejercicios para evaluar la PC, como una herramienta para determinar el impacto de los contenidos de un curso de metodología de la programación. Utilizan reactivos de dos documentos de otros países: la Olimpiada para la búsqueda de talento computacional de Sudáfrica y Bebras Computational thinking challenge, publicado por la Universidad de Oxford en 2015. Solo usan un reactivo por cada habilidad de PC a evaluar, por el tiempo que involucra para el estudiante su respuesta.
- d. Moreno-León y sus colegas (2018) mencionan la Prueba de PC desarrollada por Román González y mide el PC en términos de la habilidad para formular y resolver problemas con base en conceptos fundamentales de cómputo. Está dirigida a estudiantes de entre 10 y 16 años, contiene 28 items.

Cursos, actividades y concursos:

- a. Basogain, Olabe, Olabe y Rico (2018) realizaron cursos para medir el PC en estudiantes de educación primaria, secundaria y bachillerato en modalidad híbrida, con el apoyo de un ambiente de trabajo virtual en plataforma Moodle. El primer curso (PC-O1) se basó en una evaluación constante acompañada de retroalimentación, así como la creación de un portafolio de proyectos de cada estudiante, la evaluación sobre el manejo de conceptos fundamentales y el análisis de pequeños scripts de programación. En el segundo curso los estudiantes completan proyectos (películas, juegos o aplicaciones interactivas). A partir de una rúbrica, los estudiantes evalúan no solo sus proyectos sino los de sus compañeros. También se emplearon autoevaluaciones (conceptos fundamentales, vocabulario y reglas de programación), exámenes (para documentar el progreso) y evaluación de pares. Con este tipo de evaluación aprenden a leer código y a interpretar y evaluar su complejidad y calidad, así como a ser responsables en el comentario y evaluación del trabajo de otros. En general se observa una mejora de los resultados de los estudiantes a partir del contacto con los ejercicios y el sitio.
- b. Como ejemplos importantes de herramientas que se han desarrollado para evaluar el PC, Tran (2019) menciona: el sistema de Koh, Basawapatna, Bennett y Repenning de 2010 que se basa en un análisis semántico de juegos creados por estudiantes; el de Werner, Denner, Campe y Kawamoto a partir de la implementación de retos en un entorno 3D de juego que utiliza Alice; el análisis de Brennan y Resnick de 2012 con un marco 3D con artefactos de proyectos en Scratch y el de Bers, Flannery, Kazakoff y Sullivan de 2014 que valora niños pequeños (de 4.9 a 6.5 años) con programas escritos que se derivan de actividades de un currículum de robótica.

- c. Moreno-León y sus colegas (2018) hablan del proyecto Bebras, el cual es una iniciativa internacional para potenciar el PC. A través de un concurso que contiene tareas, sólo en 2015, logró la participación de 1.3 millones de chicos de 36 países.

Robots y apps:

- a. Kusnendar & Prabawa (2018) hicieron un estudio para ver el impacto del lenguaje de programación Karel en los estudiantes, así como indagar sobre formas para evaluar el PC. Se trata de un robot simulado usado para enseñar programación para principiantes. Los resultados resultaron positivos. Los aspectos evaluados fueron: reconocimiento de patrones, abstracción y diseño de algoritmos.
- b. El estudio de Boticki, Kovacevic, Pivalica & Seow (2018) presenta una herramienta llamada Matko (web app similar a Scratch) que se usó en Croacia con estudiantes de primer año para realizar actividades de PC en temas de lenguaje, ciencia y matemáticas. Los aspectos evaluados fueron: identificación de secuencia, seguimiento de algoritmos, reconocimiento de objetos y sus propiedades, y eliminación de pasos innecesarios para la resolución de pequeñas tareas. El estudio identifica que esta herramienta es útil para que los niños de este nivel (7-8 años) experimenten con estrategias de ensayo y error para encontrar soluciones.
- c. Chang, Sun, Wu & Guizani (2018) proponen una nueva herramienta para evaluar actividades de PC realizada en Scratch. La herramienta más utilizada para realizar esta medición es Dr. Scratch pero se ha detectado un alto nivel de error y baja eficiencia. Los autores contrastan este Scratch Analysis Tool (SAT), basado en ANTRL, con diversas herramientas y evalúan con un funcionamiento superior a Dr. Scratch en habilidades de evaluación de proyectos en Scratch.
- d. Rose, Habgood & Jay (2018) destacan el trabajo de Foerster, Foerster y Loewe de 2018 donde señalan que Dr. Scratch se ha usado en estudios recientes como una medida de PC.
- e. Moreno-León y sus colegas (2018) también señalan que Dr. Scratch es una herramienta que mide y promueve el PC.

Modelo:

- a. Burbaite, Drasute & Stuikeys (2018) presentan la revisión de un marco ya presentado con anterioridad para el área de Ciencias de la computación, basado en la taxonomía de Bloom y habilidades de PC. En esta revisión los autores integran un modelo para evaluar las actividades propuestas y presentan los resultados de dos estudios de caso. Su principal aportación es mostrar la alineación que lograron entre habilidades concretas para el curso de Ciencias de la computación, la taxonomía de Bloom y los cinco componentes clásicos de PC (Abstracción, descomposición, reconocimiento de patrones, representación de datos y algoritmos).



IV. Propuestas para potenciar el PC

Karchapakirin y Anutariya (2018) presentan el desarrollo de un chatbot (ScratchThAI) que, en un ambiente gamificado, apoya a estudiantes de primaria en el aprendizaje de Scratch en escuelas de Tailandia. Asigna misiones de forma individual para fomentar la exploración de conceptos computacionales clave, al tiempo de promover buenos hábitos de aprendizaje.

Por su parte, Kong, Chiu y Lai (2018) presentan un instrumento de 15 reactivos que permite valorar el empoderamiento, en el área de computación de los estudiantes. Incluye cuatro componentes: significatividad, impacto, auto-eficacia creativa y autoeficacia en programación. Adicionalmente evaluaron el impacto de cuatro factores en ese empoderamiento en estudiantes de primaria alta. Concluyeron que los estudiantes con mayor interés en programación la veían como más significativa, pensaban que tenía mayor impacto, presentaban mayor auto-eficacia creativa y también en programación. Los aprendices con mejores actitudes hacia la colaboración tenían mayor auto-eficacia creativa pero no en el área de programación. Los estudiantes de grados más altos percibieron la programación como menos significativa y tenían menor auto-eficacia en el área de programación. Los varones tenían mayor interés en programar que las niñas.

Omata e Imai (2019) proponen el diseño de lecciones que promueven el PC a través del uso de un robot que se integra con los dedos del estudiante (niños) para realizar conteos binarios.

El trabajo de Ortega-Ruiperez y Asensio-Brouard (2018) se enfoca en educación secundaria y propone una estrategia de enseñanza para la adquisición de pensamiento computacional a través del enfoque maker (do it yourself) DIY. Parte de tres premisas:

“1) el pensamiento computacional mejora la resolución de problemas complejos, (2) el pensamiento computacional mejora la representación del problema, (3) el pensamiento computacional mejora la ejecución del algoritmo” (p.137).

Caballero y García-Valcárcel (2018) presentan un bot (Bee-Bot) para desarrollar el PC a través de una experiencia piloto de robótica con niños en educación inicial. Los resultados fueron prometedores, incluyendo la aceptación de los maestros con los que trabajaron.

En su estudio, Chiazzese, Fulantelli, Pipitone y Taibi (2018) trabajaron con 81 niños de nivel primaria en Italia quienes usaron Microsoft Kodu, una plataforma para desarrollo de juegos, a partir de una aproximación narrativa. Sus resultados fueron positivos. Los niños mejoraron en cuanto a la percepción sobre programación. Encontraron también que el desempeño en el nivel bachillerato y el contexto familiar tuvieron una influencia en las habilidades de programación, por lo que también tuvo impacto en el PC. El desarrollo de habilidades de programación estuvo ligado de forma cercana a su nivel de asistencia a la escuela.

Vinayakumar, Soman & Pradeep (2018) utilizaron MIT Scratch (un entorno de programación gráfico y rico en media para niños de 6 a 13 años) para la promoción del PC. Scratch es un lenguaje de programación que utiliza bloques y programación y fue desarrollado por el equipo liderado por Resnick en MIT.

En Alemania, Werneburg, Manske y Hoppe (2018) presentan en su reporte un entorno gamificado que promueve la adquisición de habilidades de PC. Trabajaron con 40 estudiantes a través de una actividad con siete niveles de aprendizaje, usando *just in time learning* y midieron su progreso en términos de aprendizaje. Debido a que las herramientas que prevalecen alrededor de la promoción del PC son lenguajes de programación basados en bloques con orientación visual como Dr. Scratch y que facilitan la valoración del nivel de PC pero no su progresión, los autores presentan un robot virtual para lograr este objetivo. Identificaron niveles específicos que resultaron difíciles para estudiantes de bajo desempeño, lo que implica que se requiere el desarrollo de ejemplos más detallados en su proceso de aprendizaje.

Chiazzese, Arrigo, Chifari, Lonati & Tosto (2018) realizaron un estudio para determinar si los kits de robótica educativa son efectivos para promover habilidades de PC entre estudiantes de entre 8-10 años. En un laboratorio de Lego® Education WeDo 2.0 evaluaron aspectos como el análisis y organización de datos de manera lógica, la representación de datos a través de codificación formal, el pensamiento algorítmico y la implementación de procedimientos algorítmicos simples. Concluyen que estos kits son benéficos para el conocimiento informático y el desarrollo del PC.

Peel y Friedrichsen (2018) presentan una lección para promover el PC que utiliza prácticas de modelación y simulación para llevar a los estudiantes a una actividad complementaria de solución de problemas con base en PC. Concluyen que las clases de biología son escenarios ideales para su incorporación.

Vals, Albó-Canals & Canaleta (2018) presentan una experiencia con robótica desarrollada en Barcelona. Utilizan LEGO y Mindstorms y Scratch. Trabajaron con dos grupos y el que tenía énfasis en TIC tuvo mejores resultados.

Lübbers & Jansen (2018) trabajaron con estudiantes de entre 11 y 17 años sin entrenamiento previo en PC. Desarrollaron un microcontrolador llamado Calliope y dividieron a los aprendices en dos grupos, por edad. Concluyen que este microcontrolador permite presentar actividades que potencian el desarrollo del PC.

Mesiti, Parkes, Paneto & Cahill (2019) estudiaron los efectos de tres exhibiciones de Pixar en el desarrollo del PC: Concluyeron que es posible lograrlo en entornos de educación informal, como un museo, y que al contrastar a expertos y novicios se cuenta con información que permite mejorar la organización de las exhibiciones.

Los investigadores Mohanty & Das (2018) presentan un modelo de aprendizaje llamado Qué-por qué-cómo que plantea tareas de cuatro niveles de complejidad a los estudiantes que integra el PC para reforzar las habilidades de aprendizaje. A través de un estudio de caso, se ilustra la secuencia que involucra: identificar la información requerida para aprender el concepto; descomponerlo en subproblemas para identificar la información específica requerida; activar el conocimiento previo y contrastarlo con el nuevo; buscar patrones en la información para identificar su estructura; pensar en la organización de subconceptos; abstraer la información relevante requerida por el concepto; diseñar el algoritmo definiendo los pasos desde información inicial hasta la solución del problema; e, identificar si el concepto fue aprendido o no.

Moreno-León et al. (2018) agrupan los esfuerzos por desarrollar el PC en dos grandes categorías: los “unplugged” que no requieren de uso de tecnologías y los que involucran actividades computacionales (que, de acuerdo a su revisión, es la vía con mayor evidencia de desarrollo de PC). Ellos proponen una nueva taxonomía de esfuerzos: “unplugged”, entornos visuales basados en flechas, entornos visuales basados en bloques, lenguajes de programación en texto y programas que operan en el mundo físico.

Looi, How, Longkai, Seow & Liu (2018) consideran que para educir el PC es necesario un conjunto de esfuerzos por un periodo largo. Esta concepción implica no enseñar, sino “sacar” del estudiante el PC. En su estudio utilizan una actividad “unplugged” para ver en qué medida promueve la decomposición, la abstracción de datos y funcionalidad, la generalización, el diseño de algoritmos y la evaluación y mejora en aprendices de 9º grado. No aportan resultados concluyentes ni resulta claro si hubo un impacto de esa actividad en el PC ni, en su caso, en qué medida.

Por su parte, Rosamond (2018) trabaja con criptografía para potenciar el pensamiento algorítmico, y considera lo atractivo que resulta para los estudiantes este tipo de actividad. El artículo describe actividades del área de criptografía que no requieren tecnología (unplugged) y que potencian el interés de los niños por la informática. Leifheit, Jabs, Ninaus, Moeller & Ostermann (2018) aportan también evidencia de la utilidad de estrategias unplugged para desarrollar el PC. Desarrollaron una serie de juegos y los estudiantes lograron una comprensión del 82%, además de reportes positivos de la experiencia.

En el artículo de Tsarava, Moeller, & Ninaus (2018) se detalla el proceso de diseño de tres juegos de mesa que constituyen formatos unplugged para el desarrollo de PC en estudiantes de 8 a 9 años de edad. El trabajo explora de manera cuantitativa la experiencia de juego, y cualitativa con jóvenes jugadores y con expertos en gamificación y maestros. Los resultados fueron positivos.

Lee y Recker (2018) proponen el **uso de circuitos en papel para la promoción de PC con un muy bajo costo**. Esta opción se sume a las estrategias sin computadora para lograr esa meta (otras opciones incluyen CS unplugged, o las cuentas en la propuesta de Eisenberg en 2010). La creación y operación de circuitos en papel involucra restricciones basadas en reglas, así como operaciones y estados definidos. Se trata de material de bajo costo (1.50 dólares por aprendiz), manipulable y familiar para el estudiante. El artículo presenta una sencilla actividad que promueve lógica condicional y lógica condicional avanzada, así como paralelismo.

Pedersen, Andersen, Jorgensen, Koslich, Sherzai & Nielsen (2018) proponen el sistema robótico BRICKO para soportar interacción social, lúdica y tangible entre niños de 1º a 3er grados en el desarrollo de su PC. Señalan que si bien diversos gobiernos como el danés han incluido el PC en el currículum, los maestros carecen de una variedad amplia de opciones para llevar a cabo esta labor. Bricko usa un tablero al que el niño le coloca piezas de lego que constituyen “bloques de mando” (instrucciones para que se mueva el robot). Concluyen que su uso desarrolla PC, conocimiento matemático y conceptos y principios computacionales.

Ioannou & Makridou (2018) presentan una revisión de la literatura con un corpus inicial de 2,856 publicaciones y uno de nueve que cumplieron los criterios de inclusión (que incluyan evidencia de que la robótica educativa promueve el PC). Se destaca la ventaja de que aporta experiencias de aprendizaje embodied y situadas. Generan cuatro recomendaciones para investigación futura: lograr un acuerdo en la definición operacional de PC; establecer instrumentos para medirlo; investigar sobre la orquestación en el aula (inclusión de maestros de apoyo) para incorporar robótica educativa; y trabajar en un marco práctico para el desarrollo de PC a través de robótica.

Como un esfuerzo nacional, Rico & Bosagain (2018) describen algunos aspectos de la implementación del proyecto “Introducción del pensamiento computacional en las escuelas de Bogotá y Colombia”, que se implementó en 12 escuelas, en un entorno híbrido.

Pei, Wientrop & Wilensky (2018) proponen usar un micromundo matemático llamado Lattice Land para promover tanto el PC como hábitos mentales de tipo matemático, que los autores definen como las maneras en que los matemáticos piensan acerca de esta disciplina. El marco general de la propuesta consiste en contar con un currículum híbrido que promueva prácticas de PC en temas de otros dominios, en este caso, conceptos geométricos. Utilizan la taxonomía de Weintrop y su equipo de 2016 con cuatro categorías: datos, modelamiento y simulación, solución computacional de problemas y pensamiento sistémico. Concluyen que matemáticas y PC se apoyan mutuamente.

Rico y sus colegas (2018) generaron y probaron un material llamado Evolución para potenciar el PC en el curso de Programación lógica.

Yasar (2018) recomienda poner énfasis en el modelamiento y la simulación en educación, ya que se potencia un ciclo iterativo de razonamiento inductivo y deductivo.

En el proyecto de Anitha et al. (2018) los estudiantes de posgrado del área de trabajo social de una universidad en la India, usaron Scratch para elaborar juegos educativos sobre temas como cuidado del agua, nutrición y algunas otras cuestiones de salud. Estos juegos se usaron para educar a habitantes de poblaciones rurales. Es un proyecto interesante por la derrama social, pero la evaluación de los proyectos finalmente se elaboró con la percepción de los estudiantes acerca de la utilidad de la herramienta para los fines educativos mencionados. Una desventaja es que los productos de Scratch se pueden manejar en computadoras, pero no en dispositivos móviles. En una segunda fase del proyecto los estudiantes realizarán nuevos proyectos a partir del uso de AppInventor, otra herramienta desarrollada por el MIT, lo cual permite que los productos se puedan usar en dispositivos móviles.

Vinayakumar, Soman & Menon (2018) plantean que la experimentación con geometría fractal usando MIT Scratch es una estrategia importante para el desarrollo de habilidades de PC, en particular en los niveles preescolar a bachillerato. Destacan la importancia de la visualización.

Tran (2019) destaca la importancia de que el desarrollo de PC sea equitativo en todo tipo de estudiante y que en lo posible se promueva a través de otras asignaturas y no solo en las STEM y desde temprana edad.

Tramonti & Dochshyanov (2018) proponen en su capítulo una experiencia de robótica educativa (considerada no como una disciplina sino como una herramienta que profundiza la comprensión y la percepción del estudiante sobre las disciplinas), para promover el PC. Concluyen que la introducción temprana de artefactos cognitivos, tanto para su manipulación como para su construcción, conlleva efectos positivos en los procesos de enseñanza y aprendizaje.

Un aspecto interesante propuesto por Tissenbaum, Sheldon & Abelson (2019) es pasar del PC a la acción computacional, donde el estudiante al aprender computación debe poder hacer computación en formas auténticas, que impacten su vida y la de sus comunidades. Dos propuestas concretas son la *identidad computacional* y el *empoderamiento digital*. La primera se refiere a que el aprendiz se reconozca como capaz de diseñar e implementar soluciones computacionales para problemas y oportunidades que identifique. El segundo implica que el estudiante puede poner en acción su identidad computacional de formas auténticas y significativas. Destacan la utilidad del MIT App Inventor.

Un ejemplo del trabajo con PC en una clase de humanidades está en el artículo de Seoane (2018). Utiliza ética de máquinas (la involucrada en conducta

moral de artefactos basados en inteligencia artificial) para proponer a los estudiantes una actividad en que deben programar a una máquina para tomar decisiones en situaciones extremas. Utilizan el dilema de los autos autónomos y el software de *moral machines* de MIT. El investigador sostiene que la mejor manera de promover el PC, y por tanto potenciar el razonamiento lógico, es a través de estrategias pedagógicas para formular y resolver problemas en las asignaturas tanto científicas como humanísticas.

El artículo de Bell y Bell (2018) explora las posibilidades del PC en cursos de música a través del uso de programación básica para representar notación musical, escribir programas para escalas musicales, representación binaria a partir del uso de sonidos e incluso para composición. El artículo presenta ideas prácticas que los profesores de música podrían integrar al aula.

Dengel & Heuer (2018) proponen la enseñanza del PC con la intención de mejorar la comprensión del manejo de la información y de medios. Desarrollaron un curso en el que se abordan diferentes aspectos que competen a ambas áreas: representación de la información, ingeniería de software, algoritmos, redes, integridad y seguridad de los datos, lenguajes y cómputo físico. Su propuesta obedece a la necesidad de que los profesores en formación conozcan las nuevas características de los datos, su manejo y la implicaciones del formato digital para el área de la comunicación y los medios de información.

Bati *et al.* (2018) usaron el instrumento de Weintrop, Beheshti, Horn, Orton, Jona, Trouille & Wilensky (2014) para determinar el nivel de PC inicial de un grupo de estudiantes de octavo año de una escuela en Turquía. En un curso cuyo tema central es el tiempo, los autores integraron al programa una serie de actividades que promueven y evalúan el PC, de acuerdo a las habilidades valoradas por el instrumento de Weintrop *et al.* que toma en consideración los siguientes componentes:

- Habilidades para el manejo y la información de datos: recuperación, generación, manipulación, análisis y visualización.
- Habilidades de modelamiento/modelación y simulación: uso de modelos computacionales para comprender un concepto, comprensión de cómo y cuándo funciona un modelo de procesamiento de la información, uso de modelos de procesamiento de la información para desarrollar y probar soluciones, evaluar modelos computacionales, crear nuevos modelos y escalar modelos existentes

- Habilidades para la solución de problemas computacionales: identificar y depurar errores, programar, seleccionar herramientas de cómputo efectivas, calcular diferentes enfoques/soluciones a problemas determinados, desarrollar “modular informational computing solutions”, usar estrategias para la solución de problemas, crear abstracción.
- Habilidades de manejo de sistemas: examinar sistemas como un todo, reconocer relaciones dentro del sistema, pensar en niveles y visualización de sistemas, y definir, entender y manejar la complejidad.

En este estudio se determina que este enfoque mejora significativamente las habilidades de PC de los estudiantes.

Sabitzer, Demarle-Meusel & Jamig (2018) proponen actividades que promueven en el nivel primario y secundario el PC en clases de lengua, así como resultados informales y apreciaciones de los actores involucrados. Incluyen diagramas de relación de entidades, diagramas de clase UML, diagramas de actividad UML. Concluyen que el modelamiento promueve la creatividad y ayuda a extraer información relevante y que las actividades planteadas promueven PC así como aprendizajes de lengua. En otro artículo, dos de las autoras (Sabitzer & Demarle-Meusel, 2018) presentan una síntesis del congreso anual de aprendices, docentes y maestros en formación así como un resumen de los impactos de la inclusión de PC a través de proyectos en diversas asignaturas, a lo largo de dos años. El congreso se considera una forma de implementar PC en la escuela, en tanto que los maestros introducen actividades que lo involucran, en distintas asignaturas, mientras que los niños llevan a cabo proyectos donde integran conocimientos relativos a PC, además de que presentan trabajos ante un público. Los profesores en entrenamiento conocen conceptos computacionales y de PC y adquieren experiencia práctica.

Por su parte, Rose, Habgood & Jay (2018) plantean un juego de programación que permite a los estudiantes aprender a identificar y corregir malas prácticas de programación o code smells (traducido como “código que apesta”) al usar los repeats, custom blocks y clones de Scratch. Con ello se promueve PC.

Cai, Yang, Gong, MacLeod & Jin (2018) realizaron un estudio de caso a partir de la estrategia de *lab rotation* con modelo *blended learning* en una asignatura llamada Fundamentos de computación en una institución de educación superior en China. El rediseño del curso en esta modalidad demostró ser útil en cuanto al aprendizaje de contenidos relacionados con PC.

Para evaluarlo se utilizó un instrumento de 25 ítems producto de una tesis de maestría en China. En entrevistas algunos estudiantes indicaron que no era de su agrado el trabajo extraclase que representa la modalidad cuando las actividades no se concluían en el laboratorio.

Rojas-Lopez & Garcia-Penalvo (2018b) presentan material de apoyo para estudiantes que van a ingresar a carreras del área de computación. Utilizaron las imágenes relativas a PC desarrolladas por Chun y Piotrowski (Computational thinking illustrated) y las convirtieron en rompecabezas que los estudiantes armaban al terminar cada ejercicio. Los ejercicios provienen de los materiales de Bebras Computational thinking challenge de 2015. Debido a que se trata de una propuesta, no existen resultados de la misma.

Garcia-Penalvo (2018b) en su breve artículo presenta las ponencias presentadas en la línea de PC del congreso TEEM 2018 y que ejemplifican diferentes formas de promover el PC. Por un lado, está su inclusión en una población de estudiantes con síndrome de Down que incluye robótica educativa. En otro trabajo se presenta una iniciativa también de robótica que tiene como fin el establecimiento de una red nacional para el desarrollo de habilidades tecnológicas y científicas en los estudiantes. La tercera se refiere al estudio presentado en otro artículo de este estado del arte: el de Garcia-Penalvo y Mendes (2018). El cuarto corresponde a Villalba_Condori y su equipo, y no es posible identificar con precisión su temática. El quinto corresponde a educación inicial donde se usa robótica para promover habilidades de programación y PC. El sexto trabajo presenta un sistema que permite al maestro identificar el nivel de desempeño en programación de cada estudiante para mantener su motivación. Finalmente, Martínez-Valdés y Martínez-Ijaji presentan un estudio con resultados poco prometedores usando App Inventor y una herramienta de CodeMaster, así como una reflexión respecto de sus pobres resultados.

Hutchins, Biswas, Conlin, Emara, Grover, Basu, & McElhaney (2018) realizaron un estudio en el uso sinérgico de PC y STEM en física para lograr aprendizajes en ambas áreas de manera simultánea, en un marco de aprendizaje por modelamiento. Desarrollaron un entorno colaborativo y computacional para el aprendizaje de STEM (C2STEM, por su acrónimo en inglés). Sus conclusiones son: una aproximación de diseño sistemático para el desarrollo de tareas curriculares es benéfico al abordar conceptos y prácticas clave; los estudiantes que colaboran en tareas de aprendizaje por modelamiento pueden co-construir de manera exitosa modelos de fenómenos físicos; la identificación de niveles de aprendizaje sinérgico apoya el desarrollo y evaluación del material instruccional y del avance del estudiante; y el control del proceso de solución de problemas por parte de un miembro del equipo puede afectar las ganancias individuales.

Chiazzese, Fulantelli, Pipitone & Taibi (2018) exploraron el uso de narrativas y elementos de ludificación con el apoyo de la herramienta Kodu Game Lab, con 81 niños de primaria en Italia, en un proyecto llamado *Computational Thinking for children education*. Esta herramienta permite el desarrollo en 3D y la elaboración de personajes en un escenario virtual. Los resultados preliminares de su proyecto educativo sugieren efectos positivos en la adopción del de este enfoque para el desarrollo de habilidades de Pensamiento Computacional.

En su reseña del libro *The power of computational thinking* de Cruzon y McOwan de 2017, Giblin (2018) señala que se trata de una obra que presenta una variedad de actividades lúdicas que promueven pensamiento algorítmico, sobre todo.

Ching, Hsu & Baldwin (2018) exploraron opciones tecnológicas comerciales disponibles para ayudar a los jóvenes a realizar proyectos de PC e identifican las siguientes categorías: kits de robótica, juguetes para programar, herramientas de realidad aumentada, sitios o aplicaciones para el desarrollo de conceptos de programación, juegos de mesa, herramientas desarrollo de animaciones o juegos. Se revisan las características de cada opción a la luz de las oportunidades pedagógicas que refuerzan: conceptos computacionales, práctica de computación y perspectiva computational. Se trata de un panorama de las posibilidades existentes y es una guía para que profesores seleccionen la herramienta más adecuada de acuerdo a la edad y propósito.

Garneli & Chorianopoulos (2018) probaron con 34 aprendices divididos en dos grupos el impacto de dos tipos de proyecto, creación de un videojuego vs. de una simulación en el desarrollo de PC y en la motivación de los estudiantes. Ambos han sido señalados en la literatura previa como contextos que lo promueven. Los resultados indican que el grupo que trabajó con videojuegos tuvo muchos más primitivos (trabajo desarrollado) y habilidades avanzadas de PC que el grupo de simulaciones.

Colombi, Fronza, Pahl, & Basso (2018) presentan ideas de la fase pre-experimental del proyecto COCONATS, dirigido a estudiantes de K-12. Se trata de una serie de actividades que se construirán para fomentar el PC, el conocimiento sobre ingeniería de software y el desarrollo de habilidades de vida.

de Staey, Verlinde, Demoen & Marten (2018) presentan la plataforma Co-De (montada en Moodle) con actividades diseñadas para el desarrollo del PC en K-12 en el contexto de Ciencias de la computación. Por el momento ofrece 10 horas de clase, agrupadas en 4 paquetes de actividades que integran

los elementos tradicionales de PC: abstracción, generalización, pensamiento algorítmico y descomposición, además de actividades de evaluación para fomentar en el estudiante la reflexión y conciencia de lo que aprendió.

Gadanidis, Clements & Yiu (2018) presentan un estudio en el que usaron experiencias matemáticas a partir de una combinación de recursos de PC y manipulativos para trabajar con el área de teoría de grupos (en especial, en lo relativo a simetría) con niños de 3° a 6ª grados. Los principios de diseño que utilizaron son: agencia, acceso, sorpresa y audiencia. Utilizaron Scratch de MIT y Blockly de Google. Les interesaba identificar los usos o propósitos que el PC puede tener en el marco de dichos principios de diseño e identificar cuál de tres variaciones de las experiencias matemáticas resultaba más efectiva. Conciben el proceso de diseño de estas experiencias como un reto artístico. Concluyen que en lo relativo a agencia, aún no hay suficiente libertad y juego para el aprendiz, pero que han tenido avances importantes en las otras áreas.

Fuentes & Miranda (2018) señalan que aunque hay diversas plataformas para promover el PC, no existe una herramienta que incluya todos los conceptos y requerimientos (de aplicación de una metodología, desarrollo y monitoreo de actividades, medición de las habilidades desarrolladas en los estudiantes y generación de informes del progreso a lo largo del proceso). Presentan una propuesta para la formación, y para la evaluación utilizaron las pruebas de los concursos Bebras que incluyen 18 reactivos de 40 minutos cada uno.

Fernandez, Zuniga, Rosas & Guerrero (2018) presentan la experiencia de dos cursos de PC impartidos en nivel superior, uno de ellos solo a estudiantes en áreas STEM y el segundo, en formato de taller, se impartió a estudiantes y maestros de áreas no relacionadas con STEM. Los cursos se enfocaron en la resolución de problemas. Los resultados del primer curso se evaluaron a través de un pre-test y un post-test que demostró un notable mejora específicamente en la habilidad de abstracción. En el taller, los resultados se enfocaron en la pertinencia que los asistentes manifestaron acerca de la integración de temas y actividades de PC en asignaturas no relacionadas directamente con STEM.



V. Formación docente en PC

En su estudio, González, Estebanell & Peracaula (2018) indagaron sobre el concepto de PC que tienen los docentes en formación. Destacan que en general los docentes no han sido formados en esta área y que es necesario considerarla en la formación inicial. Plantean que necesitan: identificar qué proyectos pueden incorporar PC, cuáles son las tecnologías más adecuadas para esos proyectos y cuáles son las oportunidades de transformar la simbiosis entre esos proyectos y tecnologías para desarrollar PC. Concluyen que los maestros no comprendían el concepto de PC y que participar en su experiencia formativa mejora dicha concepción.

Hay tres estudios relacionados con el uso de Scratch. Marcelino, Pessoa, Vieira, Salvador y Mendes (2018) presentan el desarrollo de un curso en línea para enseñar conceptos de PC y uso de Scratch para profesores de primaria, usando Moodle como LMS. Concluyen que es posible que los maestros se formen, con calidad, en estos temas a distancia, aunque los resultados no fueron homogéneos entre los maestros de su muestra.

Adler & Kim (2018) realizaron un estudio con profesores a quienes les mostraron un tutorial de la “Hora del código” donde se realizan ejercicios para lograr un modelo animado del sistema solar con el uso de Scratch). Otro ejercicio con estos profesores visualizar la segunda ley del movimiento

de Newton con una gráfica dinámica (simulación). Al completar los ejercicios los profesores consideraron que ambas serían una estrategia efectiva en sus respectivas aulas (nivel K-8) y que era probable que lo incorporarán, aunque manifestaron preocupación por los recursos de cómputo que se requieren.

Hickmott & Prieto-Rodriguez (2018) examinaron seis años de experiencia diseñando, evaluando y mejorando la formación docente que conducen y realizaron un análisis a partir de las tensiones definidas por Brennan en 2015 (grupo-individuo; novicio-experto; enfoque en herramientas-en creación de entornos de aprendizaje; dirección-descubrimiento; aspiracional-actual), además de que añaden una que consideran que faltaba (antropología-evaluación, siguiendo la idea de Papert en el sentido de que los docentes deben estudiar el entorno de aprendizaje como antropólogos). Los docentes utilizaron la comunidad ScratchEd.

En cuanto a estrategias, se presentan tres artículos. El estudio de Villalba-Condori, Castro Cuba-Sayco, Guillen Chavez, et al (2018) identifica dos formas de aproximación, el enfoque profundo y el enfoque superficial. Concluye que los profesores requieren desarrollar estrategias metodológicas activas y profundas para obtener mejores resultados con el pensamiento computacional a través del fomento de una motivación intrínseca en el estudiante, esto fue entre alumnos de la carrera de Ciencias de la computación.

Kale, Akcaoglu, Cullen, Goh, Devine, Calvert y Grise (2018) presentan tres estrategias para lograr que los maestros establezcan conexiones entre PC y su práctica docente:

- a. ejemplos específicos del uso de herramientas de PC
- b. utilizar el marco de solución de problemas donde el contenido se presenta como el problema a resolver por parte de los estudiantes, al tiempo de promover el PC
- c. usar métodos de enseñanza de solución de problemas como apoyo para el conocimiento necesario de los procesos involucrados en el PC.

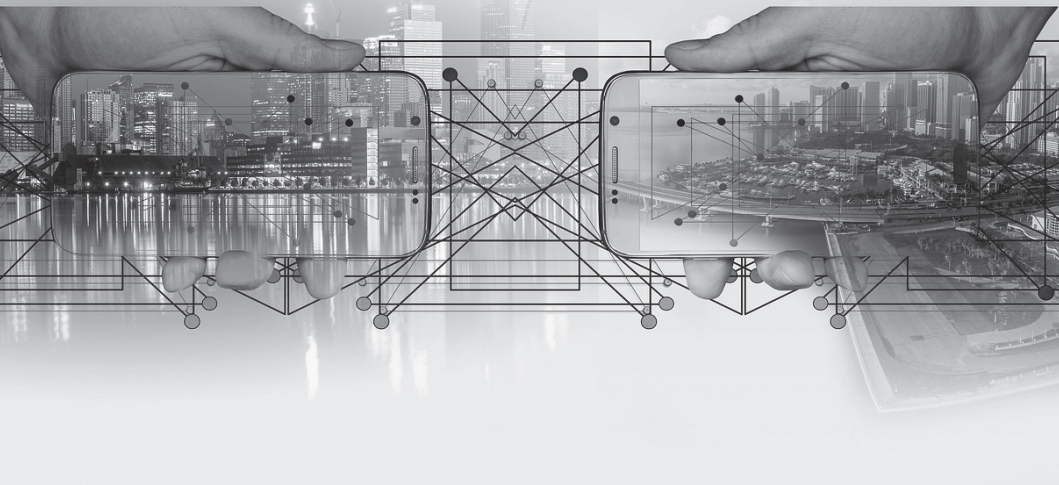
De acuerdo con Wu, Looi, Liu & How (2018), el enfoque de ciencia computacional “unplugged” que se trabaja sin usar computadoras ha sido útil para entusiasmar a los estudiantes, pero existen estudios que señalan que sí se requieren computadoras para que los alumnos trabajen con principios profundos de computación. En su estudio pudieron ver que los profesores están

motivados a trabajar PC pero no se sienten muy confiados y consideran que carecen de elementos pedagógicos y recursos de enseñanza para promoverlo entre sus estudiantes. Con el taller que ofrecieron, la confianza en sí mismos de los profesores formados en el enfoque “unplugged” subió, al igual que su interés en esta aproximación. Sin embargo, consideran que consume mucho tiempo, lo que amenaza el cumplimiento del programa que Singapur estableció en 2017 para los grados 9 y 10 titulado “nivel 0 de computación”.

Con relación a aspectos específicos de la formación, hay tres artículos. En su investigación, Leonard, Mitchell, Barnes-Johnson, Unertl, Outka-Hill, Robinson & Hester-Croff (2018) indagaron sobre los cambios que una formación docente en el área de ciencias de la computación afecta la enseñanza culturalmente responsiva y su actitud hacia el PC, usando tres contextos: robótica sola, diseño de juegos solo, robótica y diseño de juegos. En lo que respecta a PC, mejoró su entendimiento en los grupos en que se trabajó diseño de juegos y robótica combinada con diseño de juegos.

En su trabajo, Ozcan, Aktamis y Higde (2018) se dirigen a la identificación de actividades de argumentación que los profesores de Ciencia fomenten en sus clases con la intención de proporcionar a los estudiantes la oportunidad de expresar sus ideas, razonamientos, comprobaciones, etcétera. Si bien los profesores ven la habilidad de argumentación como altamente benéfica para lograr mejores procesos de pensamiento, también consideran que no recurren más a ella porque carecen de tiempo o perciben desorden en la clase cuando lo hacen. Este estudio es un puente entre el área de Humanidades y de las Ciencias.

Cateté, Lytle, Dong, Boulden, Akram, Houchins, Barnes, Wiebe, Lester, Mot y Boyer (2018), presentan los resultados iniciales de un ejercicio para integrar actividades de PC en el programa de estudios de asignaturas de Ciencia de una escuela secundaria. El estudio se centra en las estrategias aprendidas con relación al docente, por ejemplo, su nivel de compromiso con el curso. Distinguen tres niveles de compromiso: los proactivos (desean aprender habilidades y conceptos de PC), dispuestos (desean incorporar PC pero necesitan ayuda) y los poco interesados (participan para beneficiarse del estudio y que uno de los investigadores de su clase). Encontraron un mejor resultado en las tareas en aquellos grupos donde había maestros de tipo proactivo. Encontraron también que lo ideal es que los profesores vivan las lecciones de PC como estudiantes. Los profesores requieren un apoyo constante para lograr ser facilitadores efectivos de actividades de PC, pues deben trascender la barrera de la “perfección” de su trabajo y aceptar que sus respuestas o forma de trabajo con la computadora pueden no ser las únicas y las correctas.



VI. Resultados de investigación

En esta categoría hay tres estudios.

El lugar donde se vive y el grado escolar afectan el acceso a PC y sus tecnologías. Esa fue la conclusión del estudio de Kale, Akcaoglu, Cullen y Goh (2018). Los maestros de los diferentes grados estudiados y de regiones urbanas y rurales en general consideran que el PC y sus herramientas son relevantes para sus vidas y para la enseñanza. Esta opinión prevaleció en especial entre los maestros con mayor edad. Esta es una variable relevante ya que estudios previos señalan que esa motivación es un predictor de su uso futuro. La mayoría de los maestros tenían acceso a laboratorios de cómputo y alrededor del 75% a móviles; sin embargo, alrededor del 66% no estaba familiarizado con su uso, en particular en el área rural. Reportaron también un limitado uso de técnicas y herramientas de PC en su enseñanza, en especial en el nivel secundario.

Lee y Wong (2018) llevaron a cabo un estudio con aprendices de dos cursos del área de sistemas, **para identificar factores relevantes para el reuso y concluyen que el PC, junto con diferentes perspectivas de diseño, design thinking y aspectos comunicativos del diseño lo son.**

Tsai & Tsai (2018) trabajaron en Taiwan probando los efectos del aprendizaje regulado / facilitado externamente (ERL, por sus siglas en inglés) y PC en el desarrollo de habilidades y desempeño computacionales en un curso híbrido. En el ERL, existe un tutor humano que promueve el aprendizaje auto-regulado en los estudiantes. Al comparar la condición ERL vs. sin ERL, la primera tuvo una ventaja estadísticamente significativa. **Al contrastar al grupo con PC vs el que no tuvo ese entrenamiento, el primero tuvo resultados estadísticamente superiores. La condición donde los estudiantes reciben formación tanto en la condición ERL como en la de PC resultó la mejor**



VII. Retos y tendencias

De acuerdo con García-Peñalvo y Mendes (2018) el gran reto es introducir de manera transversal en el currículum proyectos colaborativos de pensamiento computacional en distintas asignaturas. Es posible agrupar en tres grandes grupos los esfuerzos previos por desarrollar el PC:

- a. Desarrollo de modelos mentales. Trata más bien de reforzar conceptos y suplementar la programación para que los estudiantes desarrollen modelos mentales más robustos
- b. PC a través de herramientas de programación. Si bien escribir código no equivale a pensar computacionalmente, el PC puede ser el resultado de una práctica planeada de programación.
- c. PC para la reforma curricular.
- d. Evaluación del PC. Ha habido diversas aproximaciones para hacerlo y destacan Dr. Scratch que evalúa de forma automática proyectos y aporta retroalimentación para desarrollar habilidades computacionales) y el uso de CTSIM (entorno de aprendizaje científico basado en PC, por sus siglas en inglés).

Chen, Tian, Zhou y Huang (2018) presentan una revisión sistemática de la literatura sobre CT de 1900 a 2018, usando ISI Web of Science. Encontraron 395 artículos y utilizaron CiteSpace, una herramienta desarrollada por Chen en Drexell University para analizar dicha literatura. Las palabras clave prevalentes incluyen PC, educación, K-12, programación, scratch, educación de ciencias computacionales, solución de problemas, robótica y diseño, entre otras. Al generar clusters las categorías son: escribir código, aprendizaje basado en juegos, programación, lenguajes de programación, K-12, educación, computación, educación secundaria, reto bebras y bibliotecas digitales, que se consideran “hot topics”. De acuerdo con su búsqueda, PC se menciona por primera vez en 1979, pero crece a partir de 2009. Su etapa de mayor desarrollo se da a partir de 2014. EUA, China y España son quienes más publicaciones tienen. El foco actual es cómo cultivar y evaluar el PC.

Yaday, Good & Caeli (2018) citan el estudio de Bocconi y sus colegas de 2016 en que encuestan a ministros de educación europeos, de Israel y de Turquía, así como a 14 expertos. Señalan que ya se tiende a promover el PC en nivel secundario pero que se ve la necesidad de hacerlo en primaria. Indican que integran PC para promover pensamiento lógico y habilidades para resolver problemas. Hubo cinco que lo hacían para atraer estudiantes al área de ciencia computacional y siete para promover habilidades de programación. Su aportación metodológica fue la inclusión de viñetas de texto donde tenían que decidir si el aprendiz estaba haciendo o no PC. No lograron identificar cómo logran los profesores traducir su comprensión del PC a la integración en el aula.



VIII. Otros factores

Obligatoriedad

Una opción para promover el PC es incluir una asignatura relacionada con computación que sea obligatoria para todos los estudiantes, como lo hicieron Gran Bretaña desde 2014, Francia en 2016 o Finlandia en 2018, y, más recientemente, Colombia en una fase piloto (Rico y Basogain, 2018), así como Australia a través de un área de aprendizaje dentro de su nuevo currículum nacional (Falkner et al., 2017 en Hickmott & Prieto-Rodríguez, 2018).

Barcelos (2018) menciona algunos países europeos que han incorporado temas de código como actividades obligatorias en ciertos niveles educativos: Bulgaria, República Checa, Dinamarca, Reino Unido, Portugal, Eslovenia y España, Austria, Bulgaria, Estonia, Hungría, Irlanda, Lituania, Malta, Polonia, Francia y Finlandia.

Interdisciplina

Las actividades que se relacionan con habilidades de PC pueden encontrarse presentes en áreas como la computación o las matemáticas, y no necesariamente estar etiquetadas como PC. El estudio de Barcelos, Munoz, Villaruel, Merino & Silveira (2018) revisa estas similitudes conceptuales y observa tres aspectos de cruce entre temas del área de las matemáticas y

actividades de PC: alternar entre diferentes representaciones semánticas, establecer relaciones e identificar patrones y construir modelos descriptivos y representativos, como mencionan en la página 819.

Género y edad

Algunos de los estudios resaltan diferencias de género (Villalba-Condori, Castro Cuba-Sayco, Guillen Chavez, et al (2018), donde los **hombres parecen tener una ventaja en el aprendizaje de PC.**

En el estudio de Mesiti et al. (2019) que investigó efectos de educación informal en el PC, a través de unas exhibiciones de Pisar en un museo, se identificaron dos exhibiciones que tuvieron particular efecto en niñas. **Ellas empezaron con un menor interés en programación que los varones pero las exhibiciones donde se guía al visitante en la solución de un experto a un problema complejo les generaron gran interés. Además, las narrativas transmedia fueron un recurso en particular útil para ellas.**

Rijke y sus colegas (2018) investigaron cuál es la mejor edad para introducir tareas de descomposición. Para ello tomaron en cuenta la dificultad percibida, la carga cognitiva y el flujo. Los resultados de su estudio muestran que no todos los estudiantes tuvieron iguales niveles de abstracción en las distintas edades (los aprendices más jóvenes tuvieron peores desempeños que los mayores, lo que es consistente con investigadores previos como Marini y Case en 1994 y las conclusiones de Piaget). Descubrieron también que **después de la edad de 9.5 años, las niñas empiezan a sobrepasar a los niños en tareas de abstracción. Adicionalmente, no encontraron diferencias en edad en las variables de dificultad percibida, carga cognitiva y flujo, lo que implica que los niños menores a ocho años pueden involucrarse en el aprendizaje de habilidades de PC.**

Condiciones especiales

Munoz, Barcelos & Villarroel (2018) desarrollaron una intervención para el desarrollo de PC en **estudiantes adolescentes con trastorno de espectro autista.** Se trata de una población desatendida en este aspecto, por lo que concluyen de su revisión de la literatura. Generaron directrices para la construcción de herramientas que atiendan a esta población y que involucran la creación de artefactos digitales. Aún después de cinco meses de la intervención, el desarrollo del PC se mantuvo.

De ese mismo grupo, Munoz, Villarroel, Barcelos, Riquelme, Quezada y Bustos-Valenzuela (2018) presentan **el taller para construcción de juegos,**

adaptado a las necesidades de individuos con trastorno de espectro autista, así como una rúbrica para evaluar el impacto de la estrategia. En su estudio, los participantes desarrollaron habilidades de alto nivel relativas a PC. La colaboración involucrada también resulta en un entorno útil para esta población.

Gonzalez-Gonzalez, Herrera Gonzalez, Moreno Ruiz, Infante-Moro & Guzmán-Franco (2018) presentan la experiencia de **enseñanza de PC a siete estudiantes con síndrome de Down**. Demuestran con su estudio que fue posible usar el robot KIBO para tal fin de manera exitosa. Destacan la importancia de no solo introducir un robot, sino convertirlo, por ejemplo, en un personaje con el que el estudiante se involucre. García-Penalvo (2018b) también incluye una breve síntesis de un trabajo sobre PC y síndrome de Down.

Referencias

- Adell, J., Llopis Nebot, M. A., Esteve Mon, F. M., & Valdeolivas Novella, M. G. (2019). The discussion on computational thinking in education. *RIED-REVISTA IBEROAMERICANA DE EDUCACIÓN A DISTANCIA*, 22(1), 171–186. <https://doi.org/10.5944/ried.22.1.22303>
- Adler, R. F., & Kim, H. (2018). Enhancing future K-8 teachers' computational thinking skills through modeling and simulations. *EDUCATION AND INFORMATION TECHNOLOGIES*, 23(4), 1501–1514. <https://doi.org/10.1007/s10639-017-9675-1>
- Anitha, P., Babu, S. K., Unnikrishnan, R., & Bhavani, R. R. (2018). Scratching out Problems: Exploring the use of Computational Thinking for Social Work in Rural India. In Kumar, V and Murthy, S and Kinshuk and Iyer, S (Ed.), 2018 IEEE TENTH INTERNATIONAL CONFERENCE ON TECHNOLOGY FOR EDUCATION (T4E) (pp. 16–19). <https://doi.org/10.1109/T4E.2018.00011>
- Barcelos, T. S., Munoz, R., Villarroel, R., Merino, E., & Silveira, I. F. (2018). Mathematics Learning through Computational Thinking Activities: A Systematic Literature Review. *JOURNAL OF UNIVERSAL COMPUTER SCIENCE*, 24(7), 815–845.
- Basogain, X., Angel Olabe, M., Carlos Olabe, J., & Rico, M.J. (2018). Computational Thinking in pre-university Blended Learning classrooms. *COMPUTERS IN HUMAN BEHAVIOR*, 80, 412–419. <https://doi.org/10.1016/j.chb.2017.04.058>
- Bati, K., Yetisir, M. I., Caliskan, I., Gunes, G., & Sacan, E. G. (2018). Teaching the concept of time: A steam-based program on computational thinking in science education. *COGNATE EDUCATION*, 5(1). <https://doi.org/10.1080/2331186X.2018.1507306>
- Bell, J., & Bell, T. (2018). Integrating Computational Thinking with a Music Education Context. *INFORMATICS IN EDUCATION*, 17(2), 151–166. <https://doi.org/10.15388/infedu.2018.09>
- Boticki, I., Kovacevic, P., Pivalica, D., & Seow, P. (2018). Identifying Patterns in Computational Thinking Problem Solving in Early Primary Education. In Yang, JC and Chang, M and Wong, LH and Rodrigo, MMT (Ed.), 26TH International Conference on Computers in Education (ICCE 2018) (pp. 470–475).
- Burbaitte, R., Drasute, V., & Stuikeys, V. (2018). Integration of Computational Thinking Skills in STEM-Driven Computer Science Education. In PROCEEDINGS OF 2018 IEEE GLOBAL ENGINEERING EDUCATION CONFERENCE (EDUCON) - EMERGING TRENDS AND CHALLENGES OF ENGINEERING EDUCATION (pp. 1824–1832).
- Caballero Gonzalez, Y. A., & Garcia-Valcarcel Munoz-Repiso, A. (2018). A robotics-based approach to foster programming skills and computational thinking: Pilot experience in the classroom of early childhood education. In GarciaPenalvo, FJ (Ed.), SIXTH INTERNATIONAL CONFERENCE ON TECHNOLOGICAL ECOSYSTEMS FOR ENHANCING MULTICULTURALITY (TEEM'18) (pp. 41–45). <https://doi.org/10.1145/3284179.3284188>
- Cai, J., Yang, H. H., Gong, D., MacLeod, J., & Jin, Y. (2018). A Case Study to Promote Computational Thinking: The Lab Rotation Approach. In Cheung, SKS and Kwok, LF and Kubota, K and Lee, LK and Tokito, J (Ed.), BLENDED LEARNING: ENHANCING LEARNING SUCCESS (Vol. 10949, pp. 393–403). https://doi.org/10.1007/978-3-319-94505-7_32
- Catete, V., Lytle, N., Dong, Y., Boulden, D., Akram, B., Houchins, J., ... Boyer, K. (2018). Infusing Computational Thinking into Middle Grade Science Classrooms: Lessons Learned. In Muhling, A and Cutts, Q (Ed.), WIPSC'18: PROCEEDINGS OF THE 13TH WORKSHOP IN PRIMARY AND SECONDARY COMPUTING EDUCATION (pp. 109–114). <https://doi.org/10.1145/3265757.3265778>
- Chang, Z., Sun, Y., Wu, T.-Y., & Guizani, M. (2018). Scratch Analysis Tool(SAT): A Modern Scratch Project Analysis Tool based on ANTLR to Assess Computational Thinking Skills. In 2018 14TH INTERNATIONAL WIRELESS COMMUNICATIONS & MOBILE COMPUTING CONFERENCE (IWCMC) (pp. 950–955).
- Chen, P., Tian, Y., Zhou, W., & Huang, R. (2018). A systematic review of computational thinking: Analysing research hot spots and trends by CiteSpace. In Yang, JC and Chang, M and Wong, LH and Rodrigo, MMT (Ed.), 26TH International Conference on Computers in Education (ICCE 2018) (pp. 211–213).
- Chiazzeze, G., Arrigo, M., Chifari, A., Lonati, V., & Tosto, C. (2018). Exploring the Effect of a Robotics Laboratory on Computational Thinking Skills in Primary School Children Using the Bebras Tasks. In GarciaPenalvo, FJ (Ed.), SIXTH INTERNATIONAL CONFERENCE ON TECHNOLOGICAL ECOSYSTEMS FOR ENHANCING MULTICULTURALITY (TEEM'18) (pp. 25–30). <https://doi.org/10.1145/3284179.3284186>

- Chiazzeze, G., Fulantelli, G., Pipitone, V., & Taibi, D. (2018). Engaging Primary School Children in Computational Thinking: Designing and Developing Videogames. *EDUCATION IN THE KNOWLEDGE SOCIETY*, 19(2), 63–81. <https://doi.org/10.14201/eks20181926381>
- Ching, Y.-H., Hsu, Y.-C., & Baldwin, S. (2018). Developing Computational Thinking with Educational Technologies for Young Learners. *TECHTRENDS*, 62(6), 563–573. <https://doi.org/10.1007/s11528-018-0292-7>
- Colombi, A., Fronza, I., Pahl, C., & Basso, D. (2018). COCONATS: Combining Computational Thinking Didactics and Software Engineering in K-12. In *SIGITE'18: PROCEEDINGS OF THE 19TH ANNUAL SIG CONFERENCE ON INFORMATION TECHNOLOGY EDUCATION* (p. 162). <https://doi.org/10.1145/3241815.3241886>
- de Staey, Z., Verlinde, T., Demoen, B., & Martens, B. (2018). Co-De: an Online Learning Platform for Computational Thinking. In Muhling, A and Cutts, Q (Ed.), *WIPSC'E'18: PROCEEDINGS OF THE 13TH WORKSHOP IN PRIMARY AND SECONDARY COMPUTING EDUCATION* (pp. 155–156). <https://doi.org/10.1145/3265757.3265790>
- Dengel, A., & Heuer, U. (2018). A Curriculum of Computational Thinking as a Central Idea of Information & Media Literacy. In Muhling, A and Cutts, Q (Ed.), *WIPSC'E'18: PROCEEDINGS OF THE 13TH WORKSHOP IN PRIMARY AND SECONDARY COMPUTING EDUCATION* (pp. 103–108). <https://doi.org/10.1145/3265757.3265777>
- Durak, H. Y., & Saritepeci, M. (2018). Analysis of the relation between computational thinking skills and various variables with the structural equation model. *COMPUTERS & EDUCATION*, 116, 191–202. <https://doi.org/10.1016/j.compedu.2017.09.004>
- English, L. (2018). On MTL's Second Milestone: Exploring Computational Thinking and Mathematics Learning. *MATHEMATICAL THINKING AND LEARNING*, 20(1, SI), 1–2. <https://doi.org/10.1080/10986065.2018.1405615> [Presentación de un número de este journal, no es un artículo]
- Fernandez, J. M., Zuniga, M. E., Rosas, M. V., & Guerrero, R. A. (2018). Experiences in Learning Problem-Solving through Computational Thinking. *JOURNAL OF COMPUTER SCIENCE & TECHNOLOGY*, 18(2), 136–142. <https://doi.org/10.24215/16666038.18.e15>
- Fuentes, A. D., & Miranda, G. (2018). Development and assessment of computational thinking A methodological proposal and a support tool. In *PROCEEDINGS OF 2018 IEEE GLOBAL ENGINEERING EDUCATION CONFERENCE (EDUCON) - EMERGING TRENDS AND CHALLENGES OF ENGINEERING EDUCATION* (pp. 787–795).
- Gadanidis, G., Clements, E., & Yiu, C. (2018). Group Theory, Computational Thinking, and Young Mathematicians. *MATHEMATICAL THINKING AND LEARNING*, 20(1, SI), 32–53. <https://doi.org/10.1080/10986065.2018.1403542>
- Garneli, V., & Chorianopoulos, K. (2018). Programming video games and simulations in science education: exploring computational thinking through code analysis. *INTERACTIVE LEARNING ENVIRONMENTS*, 26(3), 386–401. <https://doi.org/10.1080/10494820.2017.1337036>
- Giblin, P. (2018). The power of computational thinking. *MATHEMATICAL GAZETTE*, 102(554), 382–383. <https://doi.org/10.1017/mag.2018.100>
- Gonzalez, J., Estebanell, M., & Peracaula, M. (2018). Robotics or Coding? The Concept of Computational Thinking in Pre-service Teachers. *EDUCATION IN THE KNOWLEDGE SOCIETY*, 19(2), 29–45. <https://doi.org/10.14201/eks20181922945>
- Gonzalez-Gonzalez, C., Herrera Gonzalez, E., Moreno Ruiz, L., Infante-Moro, A., & Guzman-Franco, M. D. (2018). Teaching computational thinking to Down syndrome students. In GarciaPenalvo, FJ (Ed.), *SIXTH INTERNATIONAL CONFERENCE ON TECHNOLOGICAL ECOSYSTEMS FOR ENHANCING MULTICULTURALITY (TEEM'18)* (pp. 18–24). <https://doi.org/10.1145/3284179.3284191>
- Hickmott, D., & Prieto-Rodriguez, E. (2018). To assess or not to assess: Tensions negotiated in six years of teaching teachers about computational thinking. *INFORMATICS IN EDUCATION*, 17(2), 229–244. <https://doi.org/10.15388/infedu.2018.12>
- Hsu, T.-C., Chang, S.-C., & Hung, Y.-T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *COMPUTERS & EDUCATION*, 126, 296–310. <https://doi.org/10.1016/j.compedu.2018.07.004>
- Hutchins, N., Biswas, G., Conlin, L., Emara, M., Grover, S., Basu, S., & McElhaney, K. (2018). Studying synergistic learning of physics and computational thinking in a learning by modeling environment. In Yang, JC and Chang, M and Wong, LH and Rodrigo, MMT (Ed.), *26TH International Conference on Computers in Education (ICCE 2018)* (pp. 153–162).

- Ioannou, A., & Makridou, E. (2018). Exploring the potentials of educational robotics in the development of computational thinking: A summary of current research and practical proposal for future work. *EDUCATION AND INFORMATION TECHNOLOGIES*, 23(6), 2531–2544. <https://doi.org/10.1007/s10639-018-9729-z>
- García-Penalvo, F.J. (2018a). Computational Thinking. *IEEE REVISTA IBEROAMERICANA DE TECNOLOGÍAS DEL APRENDIZAJE-IEEE RITA*, 13(1), 17–19. <https://doi.org/10.1109/RITA.2018.2809939>
- García-Penalvo, F.J. (2018b). Computational thinking and programming education principles. In García-Penalvo, F.J. (Ed.), *SIXTH INTERNATIONAL CONFERENCE ON TECHNOLOGICAL ECOSYSTEMS FOR ENHANCING MULTICULTURALITY (TEEM'18)* (pp. 14–17). <https://doi.org/10.1145/3284179.3284184>
- García-Penalvo, F., & José Mendes, A. (2018). Exploring the computational thinking effects in pre-university education. *COMPUTERS IN HUMAN BEHAVIOR*, 80, 407–411. <https://doi.org/10.1016/j.chb.2017.12.005>
- Marcelino, M.J., Pessoa, T., Vieira, C., Salvador, T., & José Mendes, A. (2018). Learning Computational Thinking and Skolcraft at distance. *COMPUTERS IN HUMAN BEHAVIOR*, 80, 470–477. <https://doi.org/10.1016/j.chb.2017.09.025>
- Juskevičienė, A., & Dagienė, V. (2018). Computational Thinking Relationship with Digital Competence. *INFORMATICS IN EDUCATION*, 17(2), 265–284. <https://doi.org/10.15388/infedu.2018.14>
- Kale, U., Akcaoglu, M., Cullen, T., & Goh, D. (2018). Contextual Factors Influencing Access to Teaching Computational Thinking. *COMPUTERS IN THE SCHOOLS*, 35(2), 69–87. <https://doi.org/10.1080/07380569.2018.1462630>
- Kale, U., Akcaoglu, M., Cullen, T., Goh, D., Devine, L., Calvert, N., & Grise, K. (2018). Computational What? Relating Computational Thinking to Teaching. *TECHTRENDS*, 62(6), 574–584. <https://doi.org/10.1007/s11528-018-0290-9>
- Katchapakirin, K., & Anutariya, C. (2018). An Architectural Design of ScratchThAI A conversational agent for Computational Thinking Development using Scratch. In *PROCEEDINGS OF THE 10TH INTERNATIONAL CONFERENCE ON ADVANCES IN INFORMATION TECHNOLOGY (IAIT2018)*. <https://doi.org/10.1145/3291280.3291787>
- 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. <https://doi.org/10.1016/j.compedu.2018.08.026>
- Kusnendar, J., & Prabawa, H. W. (2018). Using NCLab-karel to improve computational thinking skill of junior high school students. In Aisyah, S and Samsudin, A and AlJupri and Kusumawaty, D and Nuraeni, E and Yulianti, K and Hasanah, L and Rusyati, L and Megasari, R and Rosjanuardi, R and Wiji (Ed.), *4TH INTERNATIONAL SEMINAR OF MATHEMATICS, SCIENCE AND COMPUTER SCIENCE EDUCATION (Vol. 1013)*. <https://doi.org/10.1088/1742-6596/1013/1/012104>
- Kwon, J., & Kim, J. (2018). A Study on the Design and Effect of Computational Thinking and Software Education. *KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS*, 12(8), 4057–4071. <https://doi.org/10.3837/tiis.2018.08.028>
- Kynigos, C., & Grizioti, M. (2018). Programming Approaches to Computational Thinking: Integrating Turtle Geometry, Dynamic Manipulation and 3D Space. *INFORMATICS IN EDUCATION*, 17(2), 321–340. <https://doi.org/10.15388/infedu.2018.17>
- Lee, C.-S., & Wong, K. D. (2018). Design - computational thinking, transfer and flavors of reuse: Scaffolds to Information and Data Science for sustainable systems in Smart Cities. In *2018 IEEE INTERNATIONAL CONFERENCE ON INFORMATION REUSE AND INTEGRATION (IRI)* (pp. 225–228). <https://doi.org/10.1109/IRI.2018.00040>
- Lee, V. R., & Recker, M. (2018). Paper Circuits: A Tangible, Low Threshold, Low Cost Entry to Computational Thinking. *TECHTRENDS*, 62(2), 197–203. <https://doi.org/10.1007/s11528-017-0248-3>
- Leifheit, L., Jabs, J., Ninaus, M., Moeller, K., & Ostermann, K. (2018). Programming Unplugged: An Evaluation of Game-Based Methods for Teaching Computational Thinking in Primary School. In Ciussi, M (Ed.), *PROCEEDINGS OF THE 12TH EUROPEAN CONFERENCE ON GAMES BASED LEARNING (ECGBL 2018)* (pp. 344–353).
- Leonard, J., Mitchell, M., Barnes-Johnson, J., Unertl, A., Outka-Hill, J., Robinson, R., & Hester-Croff, C. (2018). Preparing Teachers to Engage Rural Students in Computational Thinking Through Robotics, Game Design, and Culturally Responsive Teaching. *JOURNAL OF TEACHER EDUCATION*, 69(4), 386–407. <https://doi.org/10.1177/0022487117732317>

- Ling, U. L., Saibin, T. C., Naharu, N., Labadin, J., & Aziz, N. A. (2018). AN EVALUATION TOOL TO MEASURE COMPUTATIONAL THINKING SKILLS: PILOT INVESTIGATION. NATIONAL ACADEMY OF MANAGERIAL STAFF OF CULTURE AND ARTS HERALD, (1), 606–614.
- Looi, C.-K., How, M.-L., Longkai, W., Seow, P., & Liu, L. (2018). Analysis of linkages between an unplugged activity and the development of computational thinking. COMPUTER SCIENCE EDUCATION, 28(3), 255–270. <https://doi.org/10.1080/08993408.2018.1533297>
- Lübberts, T., & Jansen, M. (2018). Application of Microcontrollers for Fostering Computational Thinking by Using the Calliope System in School. In Yang, JC and Chang, M and Wong, LH and Rodrigo, MMT (Ed.), 26TH International Conference on Computers in Education (ICCE 2018) (pp. 500–505).
- Mesiti, L. A., Parkes, A., Paneto, S. C., & Cahill, C. (2019). Building Capacity for Computational Thinking in Youth through Informal Education. JOURNAL OF MUSEUM EDUCATION, 44(1, SI), 108–121. <https://doi.org/10.1080/10598650.2018.1558656>
- Mohanty, R., & Das, S. B. (2018). A Proposed What-Why-How (WWH) Learning Model for Students and Strengthening Learning Skills Through Computational Thinking. In Sa, PK and Sahoo, MN and Murugappan, M and Wu, Y and Majhi, B (Ed.), PROGRESS IN INTELLIGENT COMPUTING TECHNIQUES: THEORY, PRACTICE, AND APPLICATIONS, VOL 2 (Vol. 719, pp. 135–141). https://doi.org/10.1007/978-981-10-3376-6_15
- Moreno-Leon, J., Roman-Gonzalez, M., & Robles, G. (2018). On Computational Thinking as a Universal Skill A review of the latest research on this ability. In PROCEEDINGS OF 2018 IEEE GLOBAL ENGINEERING EDUCATION CONFERENCE (EDUCON) - EMERGING TRENDS AND CHALLENGES OF ENGINEERING EDUCATION (pp. 1684–1689).
- Munoz, R., Barcelos, T. S., & Villarroel, R. (2018). CT4All: Enhancing Computational Thinking Skills in Adolescents with Autism Spectrum Disorders. IEEE LATIN AMERICA TRANSACTIONS, 16(3), 909–917.
- Munoz, R., Villarroel, R., Barcelos, T. S., Riquelme, F., Quezada, A., & Bustos-Valenzuela, P. (2018). Developing Computational Thinking Skills in Adolescents With Autism Spectrum Disorder Through Digital Game Programming. IEEE ACCESS, 6, 63880–63889. <https://doi.org/10.1109/ACCESS.2018.2877417>
- Nardelli, E. (2019). Viewpoint Do We Really Need Computational Thinking? COMMUNICATIONS OF THE ACM, 62(2), 32–35. <https://doi.org/10.1145/3231587>
- Omata, K., & Imai, S. (2019). Development of Cultivate Computational Thinking using Finger Robot. In Sugisaka, M and Jia, Y and Ito, T and Lee, JJ (Ed.), ICAROB 2019: PROCEEDINGS OF THE 2019 INTERNATIONAL CONFERENCE ON ARTIFICIAL LIFE AND ROBOTICS (pp. 435–438).
- Ortega-Ruiperez, B., & Asensio Brouard, M. M. (2018). DIY robotics: computational thinking based patterns to improve problem solving. REVISTA LATINOAMERICANA DE TECNOLOGÍA EDUCATIVA-RELATEC, 17(2), 129–143. <https://doi.org/10.17398/1695-288X.17.2.129>
- Ozcan, R., Aktamis, H., & Higde, E. (2018). Computational Thinking and Integrative Education (STEAM) in Science Education. PAMUKKALE UNIVERSITESI EGITIM FAKULTESI DERGISI-PAMUKKALE UNIVERSITY JOURNAL OF EDUCATION, (43), 93–106. <https://doi.org/10.9779/PUJ857>
- Park, Y.-S., & Park, M. (2018). Exploring Students Competencies to be Creative Problem Solvers With Computational Thinking Practices. JOURNAL OF THE KOREAN EARTH SCIENCE SOCIETY, 39(4), 388–400. <https://doi.org/10.5467/JKESS.2018.39.4.388>
- Pedersen, B. K. M. K., Andersen, K. E., Jorgensen, A., Koslich, S., Sherzai, F., & Nielsen, J. (2018). Towards Playful Learning and Computational Thinking - Developing the Educational Robot BRICKO. In PROCEEDINGS OF THE 8TH IEEE INTEGRATED STEM EDUCATION CONFERENCE (ISEC 2018) (pp. 37–44).
- Peel, A., & Friedrichsen, P. (2018). Algorithms, Abstractions, and Iterations: Teaching Computational Thinking Using Protein Synthesis Translation. AMERICAN BIOLOGY TEACHER, 80(1), 21–28. <https://doi.org/10.1525/abt.2018.80.1.21>
- Pei, C. (Yu), Weintrop, D., & Wilensky, U. (2018). Cultivating Computational Thinking Practices and Mathematical Habits of Mind in Lattice Land. MATHEMATICAL THINKING AND LEARNING, 20(1, SI), 75–89. <https://doi.org/10.1080/10986065.2018.1403543>
- Rico, M. J., & Basogain, X. (2018). Computational thinking: breaking digital and educational gaps. EDMETIC, 7(1), 26–42. <https://doi.org/10.21071/edmetic.v7i1.10039>
- Rico, M. J., Basogain, X., & Moreno, N. (2018). "Evolution": Design and implementation of digital educational material to strengthen computational thinking skills. Ieee Revista Iberoamericana

- de Tecnologías del Aprendizaje-IEEE RITA, 13(1), 37–45. <https://doi.org/10.1109/RITA.2018.2809943>
- Rijke, W. J., Bollen, L., Eysink, T. H. S., & Tolboom, J. L. J. (2018). Computational Thinking in Primary School: An Examination of Abstraction and Decomposition in Different Age Groups. *INFORMATICS IN EDUCATION*, 17(1), 77–92. <https://doi.org/10.15388/infedu.2018.05>
- Rojas-Lopez, A., & Garcia-Penalvo, F.J. (2018a). Learning Scenarios for the Subject Methodology of Programming From Evaluating the Computational Thinking of New Students. *IEEE REVISTA IBEROAMERICANA DE TECNOLOGÍAS DEL APRENDIZAJE-IEEE RITA*, 13(1), 30–36. <https://doi.org/10.1109/RITA.2018.2809941>
- Rojas-Lopez, A., & Garcia-Penalvo, F.J. (2018b). Increase of confidence for the solution of problems in pre-university students through Computational Thinking. In GarciaPenalvo, FJ (Ed.), *SIXTH INTERNATIONAL CONFERENCE ON TECHNOLOGICAL ECOSYSTEMS FOR ENHANCING MULTICULTURALITY (TEEM'18)* (pp. 31–35). <https://doi.org/10.1145/3284179.3284187>
- Roman-Gonzalez, M., Perez-Gonzalez, J.-C., Moreno-Leon, J., & Robles, G. (2018). Extending the nomological network of computational thinking with non-cognitive factors. *COMPUTERS IN HUMAN BEHAVIOR*, 80, 441–459. <https://doi.org/10.1016/j.chb.2017.09.030>
- Rosamond, F. (2018). Computational Thinking Enrichment: Public-Key Cryptography. *INFORMATICS IN EDUCATION*, 17(1), 93–103. <https://doi.org/10.15388/infedu.2018.06>
- Rose, S., Habgood, J., & Jay, T. (2018). Pirate Plunder: Game-Based Computational Thinking Using Scratch Blocks. In Ciussi, M (Ed.), *PROCEEDINGS OF THE 12TH EUROPEAN CONFERENCE ON GAMES BASED LEARNING (ECGBL 2018)* (pp. 556–564).
- Rozali, N. F., Zaid, N. M., Noor, N. M., & Ibrahim, N. H. (2018). Developing A Unified Model of Teaching Computational Thinking. In 2018 IEEE 10TH INTERNATIONAL CONFERENCE ON ENGINEERING EDUCATION (ICEED) (pp. 208–213).
- Sabitzer, B., & Demarle-Meusel, H. (2018). A Congress for Children and Computational Thinking for Everyone. In Muhling, A and Cutts, Q (Ed.), *WIPSCÉ'18: PROCEEDINGS OF THE 13TH WORKSHOP IN PRIMARY AND SECONDARY COMPUTING EDUCATION* (pp. 133–138). <https://doi.org/10.1145/3265757.3265782>
- Sabitzer, B., Demarle-Meusel, H., & Jamig, M. (2018). Computational Thinking Through Modeling In Language Lessons. In *PROCEEDINGS OF 2018 IEEE GLOBAL ENGINEERING EDUCATION CONFERENCE (EDUCON) - EMERGING TRENDS AND CHALLENGES OF ENGINEERING EDUCATION* (pp. 1913–1918).
- Seoane, A. M. (2018). Computational Thinking Between Philosophy and STEM-Programming Decision Making Applied to the Behavior of “Moral Machines” in Ethical Values Classroom. *IEEE REVISTA IBEROAMERICANA DE TECNOLOGÍAS DEL APRENDIZAJE-IEEE RITA*, 13(1), 20–29. <https://doi.org/10.1109/RITA.2018.2809940>
- Tissenbaum, M., Sheldon, J., & Abelson, H. (2019). From Computational Thinking to Computational Action. *COMMUNICATIONS OF THE ACM*, 62(3), 34–36. <https://doi.org/10.1145/3265747>
- Tramonti, M., & Dochshanov, A. (2018). Students' Engagement through Computational Thinking and Robotics. *DIGITAL PRESENTATION AND PRESERVATION OF CULTURAL AND SCIENTIFIC HERITAGE*, 8, 213–219.
- Tran, Y. (2019). Computational Thinking Equity in Elementary Classrooms: What Third-Grade Students Know and Can Do. *JOURNAL OF EDUCATIONAL COMPUTING RESEARCH*, 57(1), 3–31. <https://doi.org/10.1177/0735633117743918>
- Tsai, M.-C., & Tsai, C.-W. (2018). Applying online externally-facilitated regulated learning and computational thinking to improve students' learning. *UNIVERSAL ACCESS IN THE INFORMATION SOCIETY*, 17(4), 811–820. <https://doi.org/10.1007/s10209-017-0542-z>
- Tsarava, K., Moeller, K., & Ninaus, M. (2018). Training Computational Thinking through board games: The case of Crabs & Turtles. *INTERNATIONAL JOURNAL OF SERIOUS GAMES*, 5(2), 25–44. <https://doi.org/10.17083/ijsg.v5i2.248>
- Valls, A., Albo-Canals, J., & Canaleta, X. (2018). Creativity and Contextualization Activities in Educational Robotics to Improve Engineering and Computational Thinking. In Lepuschitz, W and Merdan, M and Koppensteiner, G and Balogh, R and Obdrzalek, D (Ed.), *ROBOTICS IN EDUCATION: LATEST RESULTS AND DEVELOPMENTS* (Vol. 630, pp. 100–112). https://doi.org/10.1007/978-3-319-62875-2_9
- Velazquez-Iturbide, J. A. (2018). Towards an Analysis of Computational Thinking. In 2018 INTERNATIONAL SYMPOSIUM ON COMPUTERS IN EDUCATION (SIIE).

- Villalba-Condori KO, Castro Cuba-Sayco SE, Guillen Chavez EP, et al (2018) Approaches of Learning and Computational Thinking in Students that get into the Computer Sciences Career. In: GarciaPenalvo F (ed) SIXTH INTERNATIONAL CONFERENCE ON TECHNOLOGICAL ECOSYSTEMS FOR ENHANCING MULTICULTURALITY (TEEM'18). pp 36–40 <https://doi.org/10.1145/3284179.3284185>
- Vinayakumar, R., Soman, K. P., & Menon, P. (2018). Fractal Geometry: Enhancing Computational thinking with MIT Scratch. In 2018 9TH INTERNATIONAL CONFERENCE ON COMPUTING, COMMUNICATION AND NETWORKING TECHNOLOGIES (ICCCNT).
- Vinayakumar, R., Soman, K. P., & Menon, P. (2018). CT-Blocks: learning Computational thinking by snapping blocks. In 2018 9TH INTERNATIONAL CONFERENCE ON COMPUTING, COMMUNICATION AND NETWORKING TECHNOLOGIES (ICCCNT).
- Wang, R., & Wang, L. (2018). The Nature of Computational Thinking Used in Educational Research. In Hauer, T and Mujani, WK (Ed.), PROCEEDINGS OF THE 2018 INTERNATIONAL SEMINAR ON EDUCATION RESEARCH AND SOCIAL SCIENCE (ISERSS 2018) (Vol. 195, pp. 453–456).
- Werneburg, S., Manske, S., & Hoppe, H. U. (2018). ctGameStudio - A Game-Based Learning Environment to Foster Computational Thinking. In Yang, JC and Chang, M and Wong, LH and Rodrigo, MMT (Ed.), 26TH International Conference on Computers in Education (ICCE 2018) (pp. 543–552).
- Wu, L., Looi, C.-K., Liu, L., & How, M.-L. (2018). Understanding and Developing In-Service Teachers' Perceptions towards Teaching in Computational Thinking: Two Studies. In Yang, JC and Chang, M and Wong, LH and Rodrigo, MMT (Ed.), 26TH International Conference on Computers in Education (ICCE 2018) (pp. 735–742).
- Yadav, A., Krist, C., Good, J., & Caeli, E. N. (2018). Computational thinking in elementary classrooms: measuring teacher understanding of computational ideas for teaching science. COMPUTER SCIENCE EDUCATION, 28(4), 371–400. <https://doi.org/10.1080/08993408.2018.1560550>
- Yagci, M. (2019). A valid and reliable tool for examining computational thinking skills. EDUCATION AND INFORMATION TECHNOLOGIES, 24(1), 929–951. <https://doi.org/10.1007/s10639-018-9801-8>
- Yasar, O. (2018). Viewpoint A New Perspective on Computational Thinking. COMMUNICATIONS OF THE ACM, 61(7), 33–39. <https://doi.org/10.1145/3214354>
- Vera, J., Villalba-Condori, K., & Castro Cuba-Sayco, S. (2018). Model of a Recommendation System Based on the Context from the Analysis of Static Code for the Development of Computational Thinking: A Web Programming Case. EDUCATION IN THE KNOWLEDGE SOCIETY, 19(2), 103–126. <https://doi.org/10.14201/eks2018192103126>
- Referencias no disponibles hasta el término del análisis:**
- Chen, L., Xia, J., & Tao, J. (2018). Cultivating Computational Thinking Among Students Of Liberal Art In Basic Computer Courses. In 2018 5TH INTERNATIONAL CONFERENCE ON SYSTEMS AND INFORMATICS (ICSAI) (pp. 544–548).
- Feng, Z. (2018). Study on Computational Thinking Education in Professional Art Academies. In Kuek, M and Zhao, R (Ed.), PROCEEDINGS OF INTERNATIONAL SYMPOSIUM - EDUCATIONAL RESEARCH AND EDUCATIONAL TECHNOLOGY (pp. 38–41).
- Ferrari, A., Poggi, A., & Tomaiuolo, M. (2018). Computational Thinking in Innovative Computational Environments and Coding. In ENCYCLOPEDIA OF INFORMATION SCIENCE AND TECHNOLOGY, 4TH EDITION (pp. 2392–2401). <https://doi.org/10.4018/978-1-5225-2255-3.ch208>
- Huiming, S., & Li, W. (2018). Empirical Research on Classroom Teaching Based on Computational Thinking. In Kuek, M and Zhao, R (Ed.), PROCEEDINGS OF INTERNATIONAL SYMPOSIUM - EDUCATIONAL RESEARCH AND EDUCATIONAL TECHNOLOGY (pp. 28–32).
- Kong, S. C., Looi, C. K., Grover, S., Mirlad, M., Hoppe, U., & Biswas, G. (2018). An International Forum on Computational Thinking Education in K-12. In Yang, JC and Chang, M and Wong, LH and Rodrigo, MMT (Ed.), 26TH International Conference on Computers in Education (ICCE 2018) (p. 149).
- Lo, G. (2018). Using Excel VBA to promote computational thinking in physical chemistry. ABSTRACTS OF PAPERS OF THE AMERICAN CHEMICAL SOCIETY, 255.
- Perez, A. (2018). A Framework for Computational Thinking Dispositions in Mathematics Education. JOURNAL FOR RESEARCH IN MATHEMATICS EDUCATION, 49(4), 424–461.



El propósito de esta separata es presentar el estado del arte del concepto pensamiento computacional (PC) durante el periodo enero 2018 a febrero 2019. El texto incluye el método utilizado para la extracción de datos, las especificaciones de los términos de búsqueda, así como la síntesis de la búsqueda, dividida en las categorías resultantes: concepto de PC, herramientas para evaluarlo, propuestas para potenciarlo, formación docente en PC, resultados de investigación, retos y tendencias, y otros factores. El último apartado presenta las referencias correspondientes.

Este estado del arte forma parte del proyecto de investigación “Creación y validación de un conjunto de instrumentos para evaluar el desarrollo del pensamiento computacional en niveles de primaria y media superior para México, Paraguay y Uruguay”, proyecto financiado por el Fondo Sectorial “Inclusión Digital: Educación con Nuevos Horizontes” - 2018 de la Agencia Nacional de Investigación e Innovación de Uruguay. En él participan investigadores de tres países (Uruguay, Paraguay y México) que representan a cuatro instituciones.