

10 preguntas frecuentes (y urgentes) sobre Pensamiento Computacional

María Belén Bonello
Fernando Schapachnik

Este documento pretende contribuir con explicaciones simples y sin tecnicismos, pero con información actualizada basada en la bibliografía nacional e internacional, a varias preguntas que surgen en el ámbito educativo como consecuencia de la importancia que diversos actores (entre ellos el Consejo Federal de Educación argentino a través de sus resoluciones 263/15 y 343/18) otorgan al Pensamiento Computacional, la Programación y la Robótica. Si bien el hilo conductor es la rigurosidad fundamentada en fuentes verificables, algunos puntos aún suscitan polémica en el mundo científico y educativo.

1. ¿Es el Pensamiento Computacional la nueva piedra filosofal educativa?

La respuesta breve es que no, aunque urge avanzar en la incorporación de la programación y sus áreas conexas en la escuela (ver [pregunta 6](#)). La definición de pensamiento computacional no es del todo clara en la literatura especializada. Si bien algunos autores¹ abogan por una definición amplia del término, otros consideran que esto es problemático porque no permite clarificar de qué se está hablando cuando las propuestas curriculares hacen referencia a su incorporación, y, más adelante, cómo se mide si los estudiantes lo han incorporado o no.

En la propuesta original de Wing² se hace referencia a aquellas formas de pensar que utilizan los programadores en su labor cotidiana y que tienen que ver con cuestiones específicas sobre cómo funcionan las computadoras (por ejemplo, los *trade-offs* entre tiempo y espacio o entre energía y almacenamiento).

Otra definición posible tiene que ver con pensar en aquellas habilidades que los profesionales de la Informática adquieren durante su formación, pero que son de utilidad más allá de la Informática e incluso más allá de la interacción con sistemas digitales. Por ejemplo, la capacidad de descomponer un problema en subproblemas, la capacidad de abstracción, la generalización, etc. Según esta visión se trataría más bien de incorporar un conjunto de herramientas intelectuales³.

¹ Por ejemplo, Guzdial, M., 2011. A Definition of Computational Thinking from Jeannette Wing. In Computing Education Blog, Atlanta.

² Wing, J., 2006. Computational thinking. Commun. ACM 49, 3 (March 2006), 33-35.
DOI=<http://dx.doi.org/10.1145/1118178.1118215>

³ Ver por ejemplo National Research Council (US). (2010). Report of a Workshop on the Scope and Nature of Computational Thinking. National Academies Press.

Por último, en otros casos se utiliza el término pensamiento computacional para referirse a las habilidades cognitivas que se adquirirían cuando se ejercita la programación. Se trata de pensamiento de alto orden y se propone que la adquisición de las habilidades computacionales lo favorece⁴. Este aspecto es interesante porque no cualquier enseñanza de la algorítmica y de la programación favorecería este desarrollo cognitivo. Se señala que para lograrlo hay que lograr profundizar en la temática y abordar problemas de cierto nivel de complejidad.

Como se ve, no hay una definición única. Es por esto que resulta fundamental explicitar a qué nos estamos refiriendo cuando desarrollamos propuestas educativas cuyo objetivo es desarrollar el pensamiento computacional. Solo así podremos diseñar secuencias adecuadas y podremos, oportunamente, evaluar el éxito de las mismas.

2. ¿Hay Pensamiento Computacional sin Computación?

Como mencionamos anteriormente hay evidencia científica incipiente que permite asociar al entrenamiento en algunas áreas de las Ciencias de la Computación (en especial, la Algoritmia y la Programación) con la adquisición de herramientas intelectuales útiles en otros ámbitos o con la adquisición de habilidades cognitivas⁵. Lo que no existe es ningún tipo de evidencia de que estas habilidades puedan "destilarse" y dissociarse de la práctica informática.

Eso no significa que estas habilidades y herramientas sean atribución específica de la Informática, ni que no puedan fomentarse de otras formas, pero sí debe entenderse que si se apunta a obtener los beneficios que la literatura científica atribuye al Pensamiento Computacional, la práctica Informática, y en particular el desarrollo de algoritmos y programas en dispositivos digitales programables no puede evitarse.

⁴ La evidencia científica que relaciona la enseñanza de la programación con la adquisición de habilidades cognitivas está aún en construcción. Esto significa que existe cierta evidencia que habla de mejoras en el rendimiento escolar (ver por ejemplo Doran, K., Boyce, A., Finkelstein, S., & Barnes, T. (2012, July). Outreach for improved student performance: a game design and development curriculum. In Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education (pp. 209-214) o en algunas funciones ejecutivas (ver por ejemplo Kazakoff, Elizabeth R., Amanda Sullivan, y Marina U. Bers. "The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood." *Early Childhood Education Journal* 41.4 (2013): 245-255 o Clements, Douglas H., y Dominic F. Gullo. "Effects of computer programming on young children's cognition." *Journal of educational psychology* 76.6 (1984): 1051) pero no puede considerarse aún concluyente. Para una aproximación a la relación entre Pensamiento Computación y el pensamiento de alto orden, ver por ejemplo Cynthia C. Selby. 2015. Relationships: computational thinking, pedagogy of programming, and Bloom's Taxonomy. In Proceedings of the Workshop in Primary and Secondary Computing Education (WiPSCE '15). ACM, New York, NY, USA, 80-87. DOI: <http://dx.doi.org/10.1145/2818314.2818315>. Ver también la nota al pie siguiente.

⁵ En particular, un reciente meta análisis (Scherer, R., Siddiq, F., & Sánchez Viveros, B. (2018). The cognitive benefits of learning computer programming: A meta-analysis of transfer effects. *Journal of Educational Psychology*) documenta efectos de transferencia cercana y (de forma más moderada, lejana) atribuibles a la programación basado en el análisis riguroso de más de 400 trabajos científicos. Todos los estudios consignados se trataban de evaluar mejora en capacidades cognitivas luego de suministrar clases de programación, versus las que se obtenían luego de realizar otras actividades.

Por ende, así como se dificulta querer conseguir los beneficios que el deporte le brinda a la salud sin mover el cuerpo, cuesta terminar de delimitar un constructo consistente si la definición se inclina hacia un "Pensamiento Computacional sin Computación".

3. ¿Cualquier tipo de problema puede resolverse mediante el Pensamiento Computacional?

Desde nuestra perspectiva y tomando cualquiera de las definiciones consignadas del término Pensamiento Computacional, no cualquier problema se resuelve mediante la aplicación de Pensamiento Computacional.

Eso se debe a que si equiparamos Pensamiento Computacional a pensamiento, a secas, el concepto se disuelve sin aportar ni novedad, ni entendimiento, ni cambio educativo (ya que sin lugar a dudas la escuela es uno de los ámbitos de la sociedad que más estimula el pensamiento, desde mucho antes de la introducción de los dispositivos digitales a la misma). Para que el concepto cobre sentido hay que asociarlo a la resolución de problemas computacionales.

4. ¿Qué es un problema computacional?

No cualquier tipo de problema es un problema computacional. Los humanos tenemos la sorprendente capacidad de realizar tareas complejas mediante instrucciones imprecisas que desambiguamos por contexto. Hay momentos en que para procesar grandes volúmenes de información recurrimos a las computadoras, que ofrecen los beneficios ya por todos conocidos (automatización, velocidad de transmisión, facilidad para la visualización, etc.) al precio de requerir tomar decisiones sobre cómo representar esa información, y poder formular instrucciones muy precisas sobre cómo debe procesarse esa información.

Tomemos el ejemplo de ordenar a 10 niños por altura, de menor a mayor. Con niños de carne y hueso esa tarea puede resolverse de múltiples maneras. Se le puede pedir a los niños que se ordenen de menor a mayor y los niños se moverán por su cuenta. Se puede realizar un paneo con la vista para detectar a los que están desordenados y pedirles que se muevan a donde corresponda (incluso se les puede pedir a ellos mismos que encuentren su lugar). De hecho, lo más probable es que nos decidamos por una estrategia distinta cada vez que tengamos que ordenar 10 niños por altura, dependiendo de las particularidades del caso.

Imaginemos ahora que tenemos 1000 niños y por ende decidimos recurrir a una computadora para que nos asista en la tarea. Supongamos también que representamos a cada niño simplemente mediante el número que consigna su altura. Los números de una computadora no tiene voluntad y no pueden moverse por su cuenta. Las computadoras no pueden hacer "un paneo" para ver qué números están fuera de lugar: en todo caso deberemos pedirle que recorra todos los números, uno por uno. Si decidimos permutar dos números A y B que están desordenados, hay que explicarle a la computadora dónde almacenar a B mientras en su lugar se

almacena a A. Además, y principalmente, es necesario darle instrucciones a la computadora para que pueda ordenar *cualquier* listado de 1000 alturas y no solamente uno en particular.

Darle instrucciones a una computadora para ordenar un listado de alturas es un problema computacional, ordenar las alturas de 10 niños sin una computadora no lo es.

Un problema computacional es una situación problemática (es decir, la formulación de un objetivo a lograr cuya consecución no es obvia y por lo tanto requiere de un proceso de pensamiento para elaborar un plan efectivo) donde los elementos a manipular son abstracciones que se describen a través de un modelo computacional⁶. Dicho de otra forma, donde los elementos a manipular se describen en términos de qué cosas puede hacer con ellos una computadora.

5. ¿Por qué tratamos de no usar el término Pensamiento Computacional?

Preferimos utilizar Ciencias de la Computación, Informática, Computación o cualquiera de sus sinónimos. Porque hacen referencia a un área del conocimiento perfectamente delimitada y reconocida, con prácticas también bien definidas y queda claro de qué se está hablando. Además, los estudios rigurosos sobre el tema se centran en prácticas bien definidas (por ejemplo, la programación, u otras claramente identificadas) y analizan el efecto de esas para atribuir o no mejoras en habilidades cognitivas concretas.

Además, porque independientemente de su aspecto cognitivo, es la disciplina que permite entender una parte creciente de la realidad del siglo XXI.

No es lo mismo usar un buscador que preguntarse (y saber responder) cómo hace para encontrar en fracciones de segundo esas "agujas entre pajares", esas pocas páginas relevantes entre las miles de millones existentes. ¿Qué significa un virus informático y qué pueden y qué no pueden hacer los hackers? A la hora de elegir tecnología celular, ¿qué es y por qué es tan importante el sistema operativo que utilice? Cuando entramos a una página "segura", de esas que tienen un candadito en el navegador, ¿son realmente seguras? ¿Por qué? ¿Cómo hace una computadora para reconocer el habla y responder a una pregunta? ¿Cómo hace para poder recomendarnos cosas en base a nuestros gustos y preferencias? Cuando mandamos un mail, ¿cómo llega hasta la otra punta del planeta en segundos? Si pensamos que eso sucede únicamente porque hay una red de transmisión de datos global, no estamos entendiendo realmente qué es Internet, cuyo éxito se debe principalmente a sus protocolos. ¿Cómo hacen las redes sociales para sugerirnos nuevos amigos? Esa cosa que está en todos lados, esa computadora, ¿cómo funciona? Esa memoria que tiene, ¿qué relación guarda con la memoria de los humanos? ¿Por qué es cierto que una computadora de 1 GHz puede ser más rápida que otra de 2 GHz? ¿Y qué es eso de los GHz? ¿Qué le pasa a una computadora cuando se "cuelga"? ¿Qué podemos decir de los robots? ¿Con qué mecanismos van a proteger los estudiantes sus datos personales y su privacidad online?

⁶ Ver por ejemplo Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM*, 60(6), 33-39 y Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal*, 55(7), 832-835.

¿Esperamos que sigan recetas que no pueden analizar críticamente? ¿Cómo tomarán posición sobre el voto electrónico? ¿Sabrán deconstruir las opiniones de los "expertos"? ¿Y ante un nuevo "Y2K"? Cuando hablamos de sistemas inteligentes ¿eso quiere decir que las máquinas poseen inteligencia? ¿Qué significa que mis archivos estén "en la nube"? ¿Dónde están realmente?

No es necesario un trayecto escolar para saber que un objeto que se suelta cae al piso, que existen ríos y llanuras, o que en determinadas fechas se conmemoran diferentes acontecimientos históricos. Sin embargo, es la escuela la que problematiza estos "datos" y los transforma en conocimiento profundo, mostrándolos como parte de entramados más complejos, con causas y efectos, actores, intenciones, apropiaciones, disputas de significado, hipótesis, refutaciones y justificaciones. Que el conocimiento tenga todos estos componentes tiene la virtud adicional de que además ellos también se vuelven apropiables, con características específicas en cada disciplina (no se justifica igual en Historia que en Matemática o en Química), y pueden aplicarse más allá de la escuela, cuando se nos presenta nueva información. Por esos motivos consideramos fundamental la incorporación de las Ciencias de la Computación en la escuela.

6. ¿Qué relación hay entre Programación y Ciencias de la Computación?

La Programación es una parte (muy importante) de las Ciencias de la Computación, así como el Álgebra lo es de la Matemática.

Ciencias de la Computación es el nombre que recibe la disciplina que se ocupa de saberes tales como:

- Los necesarios para poder formular soluciones efectivas y sistemáticas a diversos tipos de problemas. Por ejemplo: pensemos en un GPS: ¿cuál camino debe sugerir a un usuario, entre todos los posibles, en un momento determinado y teniendo en cuenta las condiciones de tránsito? A esta área de la Computación se la conoce como algoritmia.
- La programación. Es decir, los conocimientos necesarios para poder volcar esas soluciones algorítmicas a los diversos lenguajes que utilizan las computadoras.
- Las estructuras de datos y las bases de datos, las dos áreas temáticas que se encargan de la forma de almacenar la información de manera que pueda ser recuperada más adelante y que se pueda buscar velozmente un dato entre miles o millones de otros, como hacen por ejemplo los buscadores de Internet.
- Las arquitecturas de computadoras. Nos referimos al entendimiento de los componentes que definen los distintos tipos de computadoras. También al entendimiento de cómo estos componentes se construyen a partir de la combinación de manipulaciones sencillas de voltaje eléctrico.
- Las redes de computadoras. Es decir, la forma en que las computadoras intercambian información permitiendo el funcionamiento de Internet y todas las aplicaciones que funcionan gracias a Internet, como la web, la mensajería instantánea, los juegos en línea, las transmisiones de audio y video, etc.
- Los fundamentos teóricos que marcan las diferencias entre los distintos lenguajes, sus posibilidades e imposibilidades, ventajas y desventajas, así como también otras áreas más

específicas entre las que se encuentran ciertas áreas de la Matemática Discreta, la Teoría de la Complejidad y otras como la Computabilidad, que estudia qué problemas son efectivamente computables y cuáles no.

- La inteligencia artificial, que se ocupa de la combinación de varias de las áreas previamente mencionadas para abordar problemas muy complejos mediante mecanismos que tienen puntos en común con la cognición humana. Incluye temas como aprendizaje automático, síntesis de información, reconocimiento de voz y de imágenes, etc.

Este listado no pretende ser exhaustivo sino dar cuenta de algunas de sus áreas fundamentales. Podría seguir detallándose con la Ingeniería del Software, los Métodos Formales, la Computación Gráfica, etc.

En algunas oportunidades y en algunos países se usa el término Informática para referirse a la disciplina que utiliza los conceptos y herramientas de las Ciencias de la Computación en el ámbito productivo (por ejemplo, se habla de Ingeniería Informática). Otras veces se utiliza el término Informática casi como sinónimo de Ciencias de la Computación. A fines educativos esta diferenciación puede considerarse un detalle menor.

7. ¿Y la robótica?

El término "robótica" hace referencia a al menos tres campos de estudio distintos:

- La robótica industrial, dedicada al diseño, confección y operación de maquinaria que realiza tareas repetitivas en contextos fabriles o similares. Se trata en general de dispositivos electromecánicos controlados digitalmente mediante un ciclo de control relativamente sencillo, más allá de que realicen tareas complejas.
- La robótica general se dedica al estudio, diseño y confección de entidades más bien autónomas, capaces de operar en ambientes no controlados y realizar planificaciones complejas para lograr su objetivo. Es decir, un brazo mecánico que suelda partes en una fábrica tiene muy poco que ver con un robot de rescate que se mete entre escombros para buscar y auxiliar víctimas.
- La robótica educativa, área donde se utilizan robots para reforzar aprendizajes de diversas disciplinas. Por ejemplo, para calcular ángulos de giro en Matemática, temas de Física en el manejo de sensores, etc.

Vivimos en una sociedad donde la robótica, en el segundo sentido del término cobra relevancia porque está presente en cada vez más áreas de la vida cotidiana y por ende adherimos a la idea de incluir la comprensión de su funcionamiento como un objetivo educativo. Esa comprensión requiere a su vez entender dos conceptos distintos: por un lado, cómo funcionan los sensores y actuadores que interactúan con el mundo físico, y por otro lado, cómo funcionan las computadoras que los controlan. Si consideramos que entre los objetivos de aprendizaje de la materia Física ya se encuentran presenten los conceptos de mecánica y de instrumental necesarios para entender el funcionamiento de un motor o la detección de un obstáculo por el rebote de una onda electromagnética, lo que resta incorporar es el entendimiento de los programas que controlan esos dispositivos.

El entendimiento de tales programas no escapa al que se logra cubriendo los temas de programación, Inteligencia Artificial, y otras áreas de la disciplina informática. Es decir, teniendo conocimientos de Computación es posible comprender la forma en que funcionan los robots, aún sin una formación específica en robótica. Dicho de otro modo, los profesionales de la informática comprenden cómo funcionan los robots sin necesidad de haberlos estudiado específicamente, porque se trata de un caso particular de una disciplina más general. Estas apreciaciones deben entenderse no en desmedro de la inclusión de la robótica en la escuela, sino en el marco de las tensiones siempre existentes por la asignación del escaso tiempo escolar.

Cabe considerar, sin embargo, que el uso de robots para el aprendizaje de otros temas (es decir, la robótica educativa) ha mostrado ser efectivo⁷, y lo mismo puede decirse de su utilización con niños pequeños para iniciar el aprendizaje de la programación⁸ ya que son elementos altamente motivadores. Es decir, valoramos su inclusión como un medio más que como un fin.

8. ¿La enseñanza de Ciencias de las Ciencias de la Computación va a resolver los problemas de la escuela?

Si bien sostenemos que las Ciencias de la Computación aportan una serie de saberes y habilidades que resultan socialmente pertinentes en el presente, de ninguna manera creemos que los problemas educativos que deben afrontar los países latinoamericanos puedan resolverse en forma exclusiva incorporando esta disciplina a la escuela.

La enseñanza de Ciencias de la Computación puede realizar una contribución valiosa en la actualización de los contenidos que se trabajan. A través del enfoque adecuado, también puede mejorar habilidades cognitivas que redundarán en un mejor desempeño general de los estudiantes. Sin embargo, y a pesar de la importancia que estos temas tienen hoy, no reside aquí la clave de soluciones que, por la naturaleza misma de los problemas a abordar, deberán ser multidimensionales.

9. TIC y alfabetización digital

La introducción de “computación” o “informática” en las escuelas argentinas comenzó a mediados de los años 80 y principios de los 90. Al margen de algunas experiencias basadas en el uso de LOGO, que buscaban introducir conceptos ligados a la programación, los contenidos

⁷ Ver por ejemplo Fabiane Barreto Vavassori Benitti, Exploring the educational potential of robotics in schools: A systematic review, *Computers & Education*, Volume 58, Issue 3, April 2012, Pages 978-988, ISSN 0360-1315, <http://dx.doi.org/10.1016/j.compedu.2011.10.006>.

⁸ Ver por ejemplo Cecilia Martinez, Marcos J. Gomez, and Luciana Benotti. 2015. A Comparison of Preschool and Elementary School Children Learning Computer Science Concepts through a Multilanguage Robot Programming Platform. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education (ITICSE '15)*. ACM, New York, NY, USA, 159-164. DOI: <http://dx.doi.org/10.1145/2729094.2742599>

predominantes en el espacio de informática estuvieron orientados a la formación de usuarios. Esto probablemente haya sido una decisión acertada en aquel momento, ya que eran muy pocas las personas que poseían una computadora en su casa y a escuela era el primer lugar de contacto con esos dispositivos. La incipiente difusión de la computación a nivel de la ciudadanía hacía que la alfabetización digital (formación de usuarios) fuera vista prácticamente como el único camino.

A su vez, la enseñanza estaba centrada en el uso de programas de oficina como procesadores de texto y planillas de cálculo⁹. Dados los altos índices de desempleo que exhibía nuestro país, el hecho de fortalecer la empleabilidad de los estudiantes a través de estas herramientas era percibido como una prioridad.

A comienzos del nuevo milenio, la tecnología educativa comienza a estar cada vez más presente en las escuelas. Entendemos a la tecnología educativa como el uso de programas informáticos desarrollados con fines didácticos. Es decir, la computadora comenzaba a utilizarse para fortalecer propuestas pedagógicas de campos disciplinares tradicionales. Estas propuestas crecieron a partir de la implementación de políticas 1 a 1, como Conectar Igualdad o Plan Sarmiento. Cabe tener en cuenta que la tecnología educativa utiliza la computadora como una herramienta para mejorar aprendizajes de diversas asignaturas, pero no la toma como objeto de estudio.

Las Ciencias de la Computación como área del conocimiento científico, se encuentran así ausentes en la mayoría de las currículas escolares. Bajo el nombre de “informática” o “computación” se dictan programas orientados al *uso* de computadoras y no a la comprensión del funcionamiento de esa tecnología.

10. ¿Se pueden incorporar el Pensamiento Computacional en forma transversal?

Existen hoy en día propuestas centradas en la integración curricular que buscan avanzar en el abordaje multidisciplinario del conocimiento en desmedro de la división en materias estancas que no dialogan entre sí. Estos proyectos buscan transversalizar una serie de saberes que se considera podrían abordarse mejor desde miradas disciplinares diversas. En algunos casos existe un cuerpo de conocimiento maduro que permite avanzar en ese sentido. En otros, las cosas no son tan claras. Tal es el caso de las Ciencias de la Computación.

Debemos separar la información con la que se cuenta por nivel educativo. El consenso internacional sobre los conceptos a trabajar en nivel inicial, así como los contenidos mínimos para dicho nivel, permiten pensar en que estos sean abordados dentro de la enseñanza general, sin necesidad de espacios específicos ni docentes especializados (aunque sí es necesario que el cuerpo docente de nivel inicial sea formado en la temática). Lo mismo puede decirse del primer ciclo de primaria. Debemos notar que estamos hablando de una enseñanza sin un espacio

⁹ Un conjunto reducido de empresas internacionales prácticamente monopolizaron tanto la provisión de programas como la capacitación docente.

específico, lo que no necesariamente significa que tenga que ser transversal a otras áreas del conocimiento.

A partir del segundo ciclo de primaria el escenario se complejiza. Si pensamos en los temas a trabajar con estudiantes que vienen transitando una formación en Ciencias de la Computación desde temprana edad, el peso de la especificidad temática aumenta y su importancia gana terreno, acercándonos a un escenario similar a lo que sucede en secundaria.

Si nos referimos a los conceptos que deberían abordarse y a las habilidades que deberían ejercitarse en la enseñanza secundaria, no existen casos documentados hoy en día en la literatura internacional sobre el tema que permitan dar cuenta de un abordaje transversal y exitoso de los diversos conceptos, saberes, prácticas y habilidades comprendidos por las Ciencias de la Computación¹⁰.

Más aún, gran parte del cuerpo de conocimiento a enseñar requiere, además de bases teóricas, del desarrollo de habilidades de orden práctico (por ejemplo, desarrollar programas, depurar y entender programas de otros, lidiar con aspectos prácticos de la tecnología y/o la interacción de múltiples tecnologías, estar al tanto de la evolución tecnológica y las herramientas específicas de programación para las diversas plataformas, etc.). Por ende, es necesario que el cuerpo docente cuente con habilidades muy específicas (puede pensarse al aula de Computación como un espacio similar al laboratorio de Química, en el que hay que tener experiencia práctica con los materiales y métodos para conducir una experiencia educativa exitosa y guiar a los alumnos a través de las dificultades que se les puedan presentar).

Para dar un ejemplo más familiar, cabe preguntarse por qué no se plantea la enseñanza transversal de Matemática, considerando, por ejemplo, que algunos de los procesos económicos que marcaron hitos en la Historia de la humanidad pueden explicarse mediante ecuaciones diferenciales. ¿Realmente imaginamos al cuerpo docente de Historia resolviendo ecuaciones diferenciales en clase? ¿Ayudando al alumnado cuando no puedan resolver las propias? ¿Es ésta la mejor estrategia para aprovechar el talento humano y la diversidad de preferencias y habilidades del cuerpo docente?

Estas características hacen que en el nivel secundario resulte muy difícil pensar en una enseñanza transversal de la Computación.

Cabe señalar también que muchos de los ejemplos que circulan como propuestas de enseñanza transversal de programación en realidad no lo son cuando se analizan en detalle. Como ejemplo, pensemos en el caso muy frecuente donde se realiza una animación computarizada de un proceso histórico. Típicamente la evaluación de ese trabajo se realizará en base al apego de la narración a los objetivos de aprendizaje de Historia. ¿Quién analiza la calidad del programa subyacente? El mero hecho de que la animación "funcione" no da cuenta de un programa que

¹⁰ Para ser precisos, la bibliografía solamente da cuenta de un conjunto muy acotado de conceptos que han logrado tratarse de manera transversal en situaciones muy específicas. Para un relevamiento sobre el tema se recomienda el capítulo "Implementation Guidance: Curriculum, Course Pathways, and Teacher Development" del K-12 Computer Science Framework Steering Committee. 2016. K-12 Computer Science Framework. Technical Report. ACM, New York, NY, USA.

cumpla con los objetivos de aprendizaje de Informática. ¿Realmente imaginamos que la evaluación de ese trabajo en la materia Historia tendrá en cuenta la aplicación o no de los correspondientes conceptos informáticos?

Bibliografía

Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal*, 55(7), 832-835.

Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education*, 58(3), 978-988.

Boyce, A. K., Campbell, A., Pickford, S., Culler, D., & Barnes, T. (2011, June). Experimental evaluation of BeadLoom game: how adding game elements to an educational tool improves motivation and learning. In *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education* (pp. 243-247). ACM.

Clements, D. H., & Gullo, D. F. (1984). Effects of computer programming on young children's cognition. *Journal of Educational psychology*, 76(6), 1051.

Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM*, 60(6), 33-39.

Guzdial, M., 2011. A Definition of Computational Thinking from Jeannette Wing. In Computing Education Blog, Atlanta.

K-12 Computer Science Framework. Technical Report. ACM, New York, NY, USA.

Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal*, 41(4), 245-255.

Kazimoglu, C., Kiernan, M., Bacon, L., & MacKinnon, L. (2011). Understanding computational thinking before programming: developing guidelines for the design of games to learn introductory programming through game-play. *International Journal of Game-Based Learning (IJGBL)*, 1(3), 30-52.

Martinez, C., Gomez, M. J., & Benotti, L. (2015, June). A comparison of preschool and elementary school children learning computer science concepts through a multilanguage robot programming platform. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education* (pp. 159-164). ACM.

Nardelli, E. (2019). Do we really need computational thinking?. *Communications of the ACM*, 62(2), 32-35.

National Research Council. (2010). Report of a workshop on the scope and nature of computational thinking. National Academies Press.

Selby, C. C. (2015, November). Relationships: computational thinking, pedagogy of programming, and Bloom's Taxonomy. In *Proceedings of the workshop in primary and secondary computing education* (pp. 80-87). ACM.

Scherer, R., Siddiq, F., & Sánchez Viveros, B. (2018). The cognitive benefits of learning computer programming: A meta-analysis of transfer effects. *Journal of Educational Psychology*.

Wing, J., 2006. Computational thinking. *Commun. ACM* 49, 3 (March 2006), 33-35.