

Diez preguntas frecuentes (y urgentes) sobre pensamiento computacional

Ten frequent (and urgent) questions about computational thinking

Resumen

En este artículo abordamos algunas preguntas frecuentes relacionadas con el pensamiento computacional y las ciencias de la computación en la escuela desde una perspectiva que busca indagar detrás de las modas que imperan en lo respectivo a los usos de la tecnología en el ámbito escolar. En este sentido, se vuelve necesario involucrarnos en otras discusiones conexas, como la de la robótica escolar o la enseñanza transversal de la disciplina Informática. Se titula “urgente” porque cierta claridad epistemológica es precisa para acompañar los procesos de inserción de estos conceptos en las escuelas argentinas.

Palabras clave: *pensamiento computacional; ciencias de la computación en la escuela; robótica escolar; transversalidad.*

Abstract

In this article we address some frequently asked questions related to Computational Thinking and Computer Science in schools from a perspective that seeks to investigate behind the fashions that prevail regarding the uses of technology in the school environment. In this sense, it becomes necessary to get involved in other related matters, such as school robotics or transversal teaching of the Computer discipline. The word “urgent” in the title accounts for the need of some epistemological clarity necessary to accompany the processes of inclusion of these concepts in Argentine schools.

Keywords: *computational thinking; computer science in school; school robotics; transversality.*

María Belén Bonello

Fundación Dr. Manuel Sadosky,
Argentina.
-
mibbonello@fundacionsadosky.org.ar

Fernando Schapachnik

Fundación Dr. Manuel Sadosky,
CONICET, Argentina.
-
fschapachnik@fundacionsadosky.org.ar

Fecha de recepción:

Octubre 2019

Aceptado:

Diciembre 2019

BONELLO, M.B. Y SCHAPACHNIK, F.
(2020). Diez preguntas frecuentes
(y urgentes) sobre pensamiento
computacional

Virtualidad, Educación y Ciencia, 20 (11),
pp. 156-167.-

■ Introducción

Los términos programación y pensamiento computacional aparecen hoy en día en el podio discursivo de la vanguardia tecnoeducativa. Es difícil para la comunidad docente distinguir si se trata de una nueva ola de tecno-optimismo o si estos términos traen consigo aportes a la compleja realidad del aula, en particular, en Argentina.

Es frecuente encontrar propuestas (a las que genéricamente llamamos tecno-optimistas) que, pletóricas de reduccionismo, confían en resolver los problemas educativos mediante algún tipo de introducción de la tecnología. Entre ellas encontramos versiones simplificadas del aula invertida, los cursos masivos online (MOOC) o trivializaciones del uso educativo de las TIC. Se trata de discursos que anclan en propuestas interesantes e, incluso, de mucho valor, pero que sus promesas de resultados desmedidos ignoran las complejidades tanto las que conlleva cualquier cambio educativo como las intrínsecas a cada propuesta. Los discursos tecno-optimistas son los que consideran que la mera incorporación de MOOC, aula invertida o TIC en la escuela, de manera casi milagrosa genera un efecto positivo en el aula.

¿Es el pensamiento computacional un fenómeno distinto o se inscribe en la larga lista de estas modas? ¿Aporta algún valor a la escuela? ¿Qué problemas soluciona y cuáles son sus complejidades intrínsecas? Como es usual en estos casos, parte de la dificultad para responder a estas preguntas se relaciona con la propia opacidad de los términos en cuestión. ¿Qué significa pensamiento computacional? ¿Qué significa exactamente programación? ¿Qué relación hay entre ambos términos? Para complicar más la cuestión, varios actores de relevancia educativa en nuestro país (como el Consejo Federal de Educación a través de sus resoluciones 263/15 y 343/18) le otorgan un rol al pensamiento computacional, a la programación y a la robótica que sobreentiende algunas concepciones y sentidos que aún están en disputa.

Este artículo pretende responder varias preguntas surgidas en el ámbito educativo respecto de estos términos con explicaciones simples y sin tecnicismos, pero utilizando información actualizada basada en la bibliografía nacional e internacional.

Si bien el hilo conductor es la rigurosidad fundamentada en fuentes verificables, algunos puntos aún suscitan polémica en el mundo científico y educativo. En esos casos este documento recoge las opiniones del equipo interdisciplinario de expertas y expertos de la Iniciativa Program.AR, de la Fundación Sadosky, con casi una década de experiencia en el tema¹.

1. ¿Qué es el pensamiento computacional? ¿Es 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 autores (Guzdial, 2011) abogan por una definición amplia del término, otros consideran que esto es problemático porque no clarifican 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.

1 Más información sobre la Iniciativa Program.AR, así como material didáctico e información sobre cursos de formación docente, puede encontrarse en www.program.ar.

En la propuesta original de Wing (2006) se hace referencia a aquellas formas de pensar utilizadas por los programadores en su labor cotidiana que se relacionan con cuestiones específicas sobre cómo funcionan las computadoras (por ejemplo, los trade-offs entre tiempo y espacio o entre energía y almacenamiento). Serían útiles porque, según Wing, la mayoría de los empleos futuros requerirían algún tipo de forma de programación de computadoras.

Otra definición posible es pensar en aquellas habilidades que los profesionales de la informática adquieren durante su formación, pero útiles más allá de su campo y 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 intelectuales (ver National Research Council, 2010) necesarias también para la “vida analógica”.

Por último, en otros casos, se utiliza el término pensamiento computacional para referirse a las habilidades cognitivas adquiridas cuando se ejercita la programación. Se trata de un pensamiento de alto orden y de desarrollo de funciones intelectuales (como la inteligencia fluida y la capacidad de planificación), y su adquisición es favorecida por las habilidades computacionales². Este aspecto es interesante porque no cualquier enseñanza de la algorítmica y de la programación favorece este desarrollo cognitivo sino se profundiza en la temática y se abordan problemas de cierto nivel de complejidad.

Como se muestra, no hay una definición única. Por esto, resulta fundamental explicitar a qué nos referimos cuando planteamos propuestas educativas cuyo objetivo es desarrollar el pensamiento computacional. Solo así diseñaremos secuencias adecuadas y evaluaremos sus éxitos.

Nuestra postura es que el pensamiento computacional no es ni más ni menos que otro nombre para la enseñanza escolar de las ciencias de la computación, es decir, una disciplina específica y no competencias generales ni metaoperaciones cognitivas. En la pregunta 6 se aborda esto con mayor detalle.

2. ¿Hay pensamiento computacional sin computación?

Como mencionamos anteriormente, hay evidencia científica 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 de habilidades cognitivas. En particular, un reciente metaanálisis (Scherer et al., 2018) 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 evaluaban una mejora en las capacidades cognitivas luego de suministrar clases de programación. Lo que no existe es algún tipo de evidencia de que estas habilidades puedan “destilarse” y disociarse de la práctica informática.

² 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 et al, 2012) o en algunas funciones ejecutivas (ver por ejemplo Kazakoff et al., 2013 o Clements et al., 1998) pero no puede considerarse aún concluyente. Para una aproximación a la relación entre pensamiento de computación y el pensamiento de alto orden, ver por ejemplo (Selby, 2015). Ver también la nota al pie siguiente.

Eso no significa que estas habilidades y herramientas sean atribución específica de la informática ni que no sean fomentadas por 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.

Por ende, así como se dificulta conseguir los beneficios que el deporte le brinda a la salud sin mover el cuerpo, cuesta delimitar un constructo consistente si la definición se inclina hacia un “pensamiento computacional sin computación”. Asimismo, no existe en la literatura científica ninguna evidencia de que esto se haya llevado adelante de forma exitosa.

3. ¿Cualquier tipo de problema puede resolverse mediante el pensamiento computacional?

Desde nuestra perspectiva, tomando cualquiera de las definiciones razonables respecto de pensamiento computacional (ver preguntas anteriores), no cualquier problema se resuelve mediante su aplicación.

Eso se debe a que si equiparamos pensamiento computacional con pensamiento en términos generales, 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, incluso, mucho antes de la introducción de los dispositivos digitales a la educación). Para que el concepto cobre sentido hay que asociarlo a la resolución de problemas computacionales que describimos en la pregunta 4.

4. ¿Qué es un problema computacional?

No cualquier tipo de problema es un problema computacional. Los humanos tenemos la capacidad de realizar tareas complejas mediante instrucciones imprecisas que desambiguamos por contexto. Hay momentos en los cuales para procesar grandes volúmenes de información recurrimos a las computadoras porque ofrecen los beneficios ya conocidos (automatización, velocidad de transmisión, facilidad para la visualización, etc.) y solo requieren que las personas tomen decisiones sobre cómo representar esa información y formular instrucciones muy precisas sobre cómo se la debe procesar. Tomemos el ejemplo de ordenar a 10 niños por altura de menor a mayor. Con niños de carne y hueso esa tarea se resuelve de múltiples maneras. Se puede pedir a los niños que se ordenen de menor a mayor y ellos se moverán por su cuenta. También, 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 pide 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 tienen voluntad y no pueden moverse por su cuenta y 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 que están desordenados, hay que explicarle a la computadora dónde almacenar a uno mientras que en su lugar se almacena al otro.

Además, y principalmente, es necesario darle instrucciones a la computadora para que ordene 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.

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 descritas a través de un modelo computacional³. Dicho de otra forma, donde los elementos a trabajar se describen en términos de lo que 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 u otros términos relacionados porque hacen referencia a un área del conocimiento perfectamente delimitada y reconocida con prácticas definidas; de esa manera, queda claro de qué se está hablando. Además, los estudios rigurosos sobre el tema (como los citados en el resto de este documento) se centran en prácticas bien definidas (por ejemplo, la programación u otras claramente identificadas) y analizan su efecto para atribuir o no mejoras en habilidades cognitivas concretas.

Tal como señala Nardelli (2019), en la escuela enseñamos matemática y no pensamiento matemático, literatura y no pensamiento literario. De la misma manera, deberíamos enseñar computación y no pensamiento computacional. Contribuir a la idea de que estos dos últimos conceptos son cosas diferenciadas aporta confusión y habilita a enseñar bajo el título de pensamiento computacional contenidos que no están relacionados con la disciplina informática.

Por otro lado, independientemente de su aspecto cognitivo, la disciplina informática permite entender una parte creciente de la realidad del siglo XXI y es por esto que su inclusión durante la escolaridad formal resulta socialmente relevante.

No es lo mismo usar un buscador que preguntarse (y saber responder) cómo hace para encontrar, en fracciones de segundo, esas pocas páginas relevantes entre las miles de millones existentes. ¿Qué significa un virus informático y qué hacen y no hacen los hackers? A la hora de elegir tecnología celular, ¿qué es y por qué es tan importante el sistema operativo utilizado? 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 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 entendemos 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? La 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? ¿Qué es eso de los GHz? ¿Qué le pasa a una computadora cuando se “cuelga”? ¿Qué podemos decir de los robots? ¿Con qué mecanismos protegerán los estudiantes

³ Ver, por ejemplo: Denning, 2017 y Aho, 2012.

sus datos personales y su privacidad online? ¿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 (el llamado “problema del año 2000”)? 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 sin sostén cae al piso, que existen ríos y llanuras, y que el 9 de julio se conmemora una fecha patria. Sin embargo, es la escuela quien 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 estos componentes le da la virtud adicional de que además pueden adquirir características específicas al ser apropiados por distintas disciplinas (no se justifica igual en Historia que en Matemática o en Química), y se aplican 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 argentina.

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 la aritmética lo es de la matemática. No hay problema en usar ambos términos como sinónimos en una charla coloquial, pero en un ámbito formal es importante distinguirlos.

Ciencias de la computación es el nombre que recibe la disciplina que se ocupa, entre otros, de los siguientes saberes:

- Los necesarios para formular soluciones efectivas y sistemáticas ante diversos tipos de problemas. Por ejemplo, pensemos en un sistema de posicionamiento global (GPS, por su escritura en inglés) ¿Cuál camino debe sugerir a un usuario entre todos los posibles para un momento determinado, 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 armar soluciones algorítmicas en los diversos lenguajes que utilizan las computadoras. Muchas veces, y en particular cuando se habla de “llevar la programación a la escuela”, se engloba a la algoritmia dentro de la programación.
- Las estructuras de datos y las bases de datos son las áreas temáticas encargadas de la forma de almacenar la información de manera que sea recuperada más adelante y se busque velozmente un dato entre miles o millones de otros, como hacen los buscadores de internet.
- Las arquitecturas de computadoras son el conocimiento sobre los componentes que definen a los distintos tipos de computadoras y 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 de las aplicaciones que funcionan gracias a ella, 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 con 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 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. Se podría seguir detallando 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. ¿Es la robótica la nueva piedra filosofal de la educación?

El término robótica hace referencia a, como mínimo, tres campos de estudio distintos:

- La robótica industrial dedicada al diseño, confección y operación de maquinaria para realizar 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 poco que ver con un robot de rescate que se mete entre escombros para buscar y auxiliar víctimas.
- La robótica educativa es el 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 presentes 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 logro 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 dentro 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 efectivo (Benitti, 2012), lo mismo puede decirse de su utilización con niños pequeños para iniciar el aprendizaje de la programación (Martínez, 2015) ya que son elementos altamente motivadores. Por eso, valoramos su inclusión como un medio más que como un fin.

8. ¿Puede la enseñanza de las ciencias de la computación contribuir a resolver los problemas de la escuela?

La escuela, tanto en Argentina como a nivel internacional, enfrenta en la actualidad una multiplicidad de desafíos. Algunos de ellos están vinculados a la inclusión de distintos sectores de la población, la mejora de los aprendizajes, la fragmentación del sistema y la consolidación de circuitos diferenciados de acuerdo al nivel socioeconómico de los estudiantes y los docentes, entre tantos otros.

Si bien desde la Iniciativa Program.AR 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 a afrontar puedan resolverse incorporando esta disciplina a la escuela.

La enseñanza de ciencias de la computación puede realizar una contribución valiosa en la actualización de los contenidos trabajados a través del enfoque adecuado, incluso, puede mejorar habilidades cognitivas que redundan 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. ¿Qué diferencia hay entre enseñar computación y enseñar con TIC?

La introducción de la materia 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 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 pocas las personas poseedoras de una computadora en su casa, por ende, era la escuela 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 cálculo⁴. 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.

4 Un conjunto reducido de empresas internacionales prácticamente monopolizó tanto la provisión de programas como la capacitación docente.

A comienzos del nuevo milenio la tecnología educativa comenzó 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 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. ¿Es la transversalidad la mejor manera de incluir las ciencias de la computación?

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 transversalizan una serie de saberes que consideramos que podrían ser abordados mejor desde miradas disciplinares diversas. En algunos casos existe un cuerpo de conocimiento que, por sus características intrínsecamente multidisciplinarias, ha permitido un abordaje transversal (por ejemplo, la educación sexual integral). Existen también competencias generales, como la producción de textos, o habilidades, como el trabajo en equipo, que (sin desmedro de ser tratadas por disciplinas específicas) es conveniente reforzarlas en todas las asignaturas (Kysilka, 1998). Distinto 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 para trabajar en nivel inicial, así como los Núcleos de Aprendizaje Prioritarios (NAP) para dicho nivel, establecidos por la resolución 343/18 del Consejo Federal de Educación, permite 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 específico en el currículum, 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 para 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 que abarcan a las ciencias de la computación⁵. Por caso, un reciente informe de la Organización de las Naciones Unidas para la Educación, la Ciencia

5 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.

y la Cultura (UNESCO, por su escritura internacional) en conjunto con el comité especializado en la enseñanza de computación afirma que: “La integración de Ciencias de la Computación en otras asignaturas no fue efectivo” (UNESCO, 2019).

Más aún, gran parte del cuerpo de conocimiento para 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 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. En ese sentido, pensamos al aula de computación como un espacio similar al laboratorio de química, en el cual 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.

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. Por ejemplo, pensemos en el caso 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 para 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 cumpla con los objetivos de aprendizaje de Informática. Resulta difícil imaginar que la evaluación de ese trabajo en la materia Historia tendrá en cuenta la aplicación o no de los correspondientes conceptos informáticos.

■ Conclusión

Dado que el término de pensamiento computacional parece haber llegado para quedarse, invitamos a pensarlo como un sinónimo escolar de ciencias de la computación, que es una disciplina del conocimiento científico cuyo corpus de contenido está claramente definido. Esto permite tomar decisiones no arbitrarias al momento de diseñar programas de estudios para abordar estos temas. Partir de definiciones más lábiles habilita la incorporación de contenidos no relacionados con el campo bajo del nombre de pensamiento computacional.

Adicionalmente, consideramos pertinente avanzar hacia incorporaciones de las ciencias de la computación en la escuela que transiten por el sendero del espacio curricular específico ya que, como detallamos anteriormente, las implementaciones transversales no resultan aconsejables para esta disciplina.

Como cierre expresamos nuestro convencimiento acerca de la importancia y urgencia que reviste la incorporación de estos contenidos a las currículas escolares. Esto se fundamenta, principalmente, en la relevancia social adquirida por el dominio fluido de estos saberes. Este conocimiento contribuye a garantizar el pleno ejercicio de los derechos ciudadanos en el siglo XXI.

Referencias Bibliográficas

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*, 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.

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*, 209-214. ACM.

GUZDIAL, M., (2011). A Definition of Computational Thinking from Jeannette Wing. In *Computing Education Blog*, Atlanta.

K-12 Computer Science Framework (2016). 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.

KYSILKA, M. L. (1998). Understanding integrated curriculum. *Curriculum journal*, 9(2), 197-209.

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*, 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*, 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*.

UNESCO/IFIP TC3 MEETING AT OCCE. Coding, Programming and the Changing Curriculum for Computing in Schools Report – Date of report: 4 February 2019.

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