

Dossier: "Aportes del pensamiento computacional a la educación en ciencias y tecnologías"

Pensar y computar: hacia una crítica conceptual al pensamiento computacional

IRICE
REVISTA
INSTITUTO ROSARIO DE INVESTIGACIONES EN CIENCIAS DE LA EDUCACIÓN

Thinking and computing: toward a conceptual critique of computational thinking

 **Alejandro Chmiel ***

Universidad de la República, Uruguay
alejandro.chmiel@gmail.com

Revista IRICE

núm. 49, e2093 2025
Consejo Nacional de Investigaciones Científicas y Técnicas,
Argentina
ISSN-E: 2618-4052
Periodicidad: Frecuencia continua
revista@irice-conicet.gov.ar

Recepción: 06 junio 2025
Aprobación: 07 julio 2025

DOI: <https://doi.org/10.35305/revistairice.vi49.2093>

URL: <https://portal.amelica.org/ameli/journal/746/7465386012/>

Resumen: Este artículo propone una crítica conceptual al pensamiento computacional (PC), entendido como una noción que oscila entre una concepción informal –asociada a habilidades cognitivas generales como la abstracción, el análisis y la resolución estructurada de problemas– y una concepción técnica, vinculada a la automatización y al uso de formalismos computacionales. A partir de esta ambigüedad, se desarrolla un marco gradualista que permite distinguir entre PC informal y PC técnico en función de tres componentes fundamentales: el problema que se aborda, los medios expresivos empleados para su representación y resolución, y el agente encargado de ejecutar la solución. Se argumenta que el nivel de formalización requerido depende del contexto y que las capacidades cognitivas generales implicadas en la formulación y evaluación de problemas deben ser reconocidas como parte constitutiva del PC. Esto permite preservar su función epistémica y evitar una reducción puramente instrumental. La propuesta busca superar una concepción excesivamente técnica del PC y contribuir a su comprensión como una práctica compleja que articula dimensiones cognitivas, técnicas y formativas, desplegándose progresivamente a lo largo de diferentes niveles educativos. El trabajo concluye que una definición del PC centrada exclusivamente en la automatización rompe con la continuidad histórica del pensamiento algorítmico y debilita su articulación pedagógica entre etapas formativas, si bien reconoce que la orientación hacia la automatización constituye un rasgo distintivo y transversal del PC.

Palabras clave: pensamiento computacional, formalización, automatización, crítica conceptual, resolución de problemas.

Notas de autor

*

Magíster en Ciencias Humanas, opción Filosofía Contemporánea, por la Facultad de Humanidades y Ciencias de la Educación de la Universidad de la República, Uruguay (FHCE-UDELAR).

Abstract: This article offers a conceptual critique of computational thinking (CT), understood as a notion that oscillates between an informal conception –associated with general cognitive abilities such as abstraction, analysis, and structured problem-solving– and a technical conception, tied to automation and the use of computational formalisms. Based on this ambiguity, a gradualist framework is developed to distinguish between informal and technical CT according to three key components: the problem being addressed, the expressive means used for its representation and resolution, and the agent responsible for executing the solution. It is argued that the level of formalization required depends on context, and that the general cognitive abilities involved in the formulation and evaluation of problems should be acknowledged as constitutive elements of CT. This approach aims to preserve its epistemic function and avoid a purely instrumental reduction. The proposal seeks to move beyond an overly technical view of CT and to support its understanding as a complex practice that interweaves cognitive, technical, and educational dimensions, progressively unfolding across different educational stages. The article concludes that a definition of CT focused exclusively on automation disrupts the historical continuity of algorithmic thinking and weakens its pedagogical articulation across learning levels, although it also acknowledges that an orientation toward automation remains a distinctive and transversal feature of CT.

Keywords: computational thinking, formalization, automation, conceptual critique, problem solving.

Introducción: entre la transversalidad del pensamiento computacional y la especificidad de las ciencias de la computación

Desde hace al menos dos décadas, la expresión *pensamiento computacional* (PC) circula con creciente intensidad para referirse a una serie de actividades intelectuales asociadas paradigmáticamente con las ciencias de la computación –aunque no exclusivamente con ellas–, las cuales se han vuelto progresivamente indispensables en el trabajo científico, e incluso, en algunas disciplinas humanísticas. Como señala Jeannette Wing en un influyente artículo que contribuyó a popularizar dicha expresión, el PC constituye una habilidad fundamental para todas las personas, no solo para quienes se dedican a la informática, y debería incorporarse, junto con la lectura, la escritura y la aritmética como parte del desarrollo de la capacidad analítica de todos los niños (Wing, 2006).

De este modo, el PC se presenta como una forma básica de pensar que debería incorporarse desde edades tempranas en la educación, no solo por su notable fertilidad para abordar problemas prácticos en diversos dominios, sino porque desarrolla capacidades analíticas sumamente generales. Por ejemplo, Grover y Pea (2018) sostienen que el PC abarca un conjunto de habilidades específicas orientadas a la resolución de problemas, que incluyen la abstracción, la descomposición en subproblemas, la evaluación, el reconocimiento de patrones, la lógica y el diseño de algoritmos. En este sentido, el PC puede entenderse como un conjunto de formas de razonamiento transversales a la resolución de problemas, que se manifiestan también en otras disciplinas.^[1]

Sin embargo, dado que estas habilidades son transversales, es decir, que aparecen involucradas en la resolución de problemas de matemáticas, física, o en el diseño de un artefacto –entre infinidad de otros ejemplos–, surge la pregunta: ¿en qué sentido este tipo de habilidades generales se vinculan *específicamente* con las ciencias de la computación? La pregunta no es tan obvia y la respuesta no es inmediata. Por tanto, esta cuestión requiere un análisis más detenido, como veremos a través de este trabajo.

Las habilidades más generales vinculadas al PC, como aquellas que consisten en la construcción de modelos mediante procesos de abstracción y el desarrollo de procedimientos de resolución de problemas a partir de dichos modelos, en principio no están necesariamente vinculadas con la implementación o automatización de la resolución. En particular, existe un enfoque sobre la enseñanza de las ciencias de la computación que trabaja con la idea de que no es

necesario introducir computadoras para enseñar algunos conceptos básicos.^[2]

Sin embargo, ¿la ciencia de la computación podría existir sin computadoras? Si bien muchos conceptos centrales para la disciplina, como puede ser el concepto de *algoritmo*, aparecen mucho antes que las computadoras, la noción de *programa*, que es central para la ciencia de la computación, suele definirse atendiendo a la dualidad entre el algoritmo y su implementación física.^[3] Por tanto, aunque los algoritmos pueden entenderse como entidades simbólicas, es difícil sostener que los programas puedan existir sin un soporte físico en el cual se implementan y desde el cual pueden computar la solución a un problema.

Desde luego, si bien es posible abordar muchos conceptos que son parte del PC sin necesidad de automatización, una de las grandes ventajas de la resolución de problemas mediante un abordaje computacional está en la enorme capacidad de cómputo de las computadoras, es decir, en la posibilidad de automatizar una solución. Como señalan Denning y Tedre (2021), este es un problema central al momento de entender el PC. Para los autores, algunos defensores del concepto han pasado por alto la distinción entre realizar tareas pequeñas y tareas de gran escala. Sostienen que los algoritmos son ejecutados por “agentes de información” y que los seres humanos son tales agentes. Sin embargo, esta afirmación –que ha sido incorporada de manera equívoca en algunas definiciones de PC para los niveles de educación primaria y secundaria– solo resulta aplicable a tareas pequeñas que pueden completarse en poco tiempo, y no a tareas de mayor complejidad.

De este modo, el PC se inscribe en una red de tensiones conceptuales que pueden organizarse a lo largo de un eje que va desde una concepción amplia, asociada a habilidades generales de resolución de problemas, hasta una concepción más técnica y específica, propia del ámbito de las ciencias de la computación. En este espectro, el lugar que ocupa la automatización sigue siendo objeto de debate: si bien parece un componente indispensable en las formas más acotadas de PC, no resulta evidente que lo sea en sus versiones más generales.

Así, el problema de la automatización está en el centro de esta tensión entre un PC amplio y un PC acotado. Por otra parte, la automatización se articula de manera natural con la trayectoria formativa de un profesional en ciencias de la computación. En esta línea, Denning y Tedre (2021) afirman que las prácticas asociadas al PC pueden ubicarse a lo largo de un espectro que va desde los niveles iniciales hasta los profesionales.

Si asumimos que las habilidades generales se encuentran en el inicio de ese recorrido –es decir, que en el comienzo se encuentra el PC en

sentido amplio, especialmente si consideramos que debería ser estimulado desde edades tempranas–, entonces podríamos argumentar que, en ciertos tramos de ese continuo hacia la profesionalización, comienzan a incorporarse progresivamente conceptos teóricos más complejos, así como también crece la exigencia de automatización en la resolución de problemas. Dicho de otro modo, en ese continuo se observa una transición gradual hacia formas más técnicas del PC, donde la automatización gana centralidad. Esta posición moderada respecto del rol de la automatización será reevaluada hacia el final del artículo, a la luz del marco elaborado.

En cualquier caso, el PC se manifiesta en distintos niveles de rigor conceptual, formalización y exigencia de automatización. De allí surge una pregunta que estructura la tensión subyacente en el debate: ¿la automatización forma parte constitutiva del PC? Esta cuestión recorre transversalmente el trabajo y será retomada en las conclusiones, donde propondremos una respuesta matizada a partir del análisis conceptual desarrollado.

Ahora bien, en la medida en que relacionamos el PC con la capacidad de producir un programa que resuelva un problema, lo estamos asociando con la automatización y con los problemas técnicos que esta implica. Por otra parte, el PC podría desligarse, al menos en ciertos contextos, de la exigencia de la automatización, sobre todo en la medida en que el objetivo en dicho contexto sea una aproximación más informal a los conceptos y habilidades generales asociados a él.

En síntesis, el PC abarca, al menos, dos sentidos que conviene distinguir para evitar equívocos. En particular, oscila entre una concepción amplia, que lo entiende como una forma general de pensamiento –casi como un nuevo lenguaje transversal, incorporado tanto en las ciencias como en la vida cotidiana–, y una concepción más restringida, centrada en la comprensión y uso de ciertos conceptos técnicos. Por ejemplo, aunque la noción de algoritmo es una noción central para el PC –tanto en un sentido amplio como restringido–, la noción de programa forma parte del PC en un sentido técnico. El problema, en cierto modo, es comprender la relación entre una y otra. Dicho de forma más general: ¿cuál es la relación entre el PC en sentido amplio y en sentido técnico?

A continuación presentaremos con mayor detalle la distinción entre los dos sentidos que tiene el PC; luego argumentaremos que el PC, en ambos casos, debería vincularse con los contextos y prácticas en los que se despliega. Posteriormente, ofreceremos un marco elemental que permita precisar dicha distinción con mayor claridad conceptual. Por último, sostendremos que el PC debería comprenderse en el marco de procesos generales de análisis y

resolución de problemas, los cuales requieren –de forma irreductible– de una comprensión profunda de los mismos, la cual está mediada por herramientas de análisis tradicionales, como la descripción adecuada de un problema en lenguaje natural, así como la deliberación individual y colectiva respecto de lo que es relevante en su resolución. Este punto es crucial para mantener una distancia crítica que permita repensar el PC integrado con nuestras capacidades de comunicación, comprensión y análisis.

Finalmente, en las conclusiones, integraremos estas observaciones dentro de una perspectiva crítica sobre el PC, que incorpora algunos de estos elementos. Desde luego, el objetivo del trabajo no es cuestionar los avances alcanzados en la incorporación educativa del PC, sino proponer un marco más robusto para evitar errores metodológicos y conceptuales que pueden surgir de una comprensión simplificada o excesivamente entusiasta de su papel.

Dos sentidos del pensamiento computacional

Desde luego, existe una larga y compleja discusión sobre la relación entre el pensamiento y los medios mediante los cuales el mismo se puede realizar. El lenguaje y la escritura son, en cierto sentido, herramientas conceptuales tan básicas que parecen indiferenciables de la actividad misma de pensar. Por otra parte, hay una clara diferencia entre ser capaz de lograr una comunicación oral y escrita exitosa, y ser un escritor profesional. Con el PC parece ocurrir una ambigüedad semejante, es decir, se lo entiende como una familia de habilidades cognitivas básicas, así como también un tipo de pensamiento técnico sofisticado.

Veamos esta diferencia en dos pasajes del clásico artículo de Wing (2006). Al introducir lo que entiende por PC, la autora lo describe mediante ejemplos que remiten al razonamiento recursivo, el procesamiento en paralelo, la interpretación del código como dato y del dato como código, la verificación de tipos, la identificación de alias o nombres múltiples, y la valoración de programas no solo por su corrección y eficiencia, sino también por su simplicidad y elegancia. En este pasaje, el PC se identifica con habilidades cognitivas altamente técnicas, características de la manera de pensar de los científicos de la computación. Más adelante, sin embargo, Wing extiende su alcance al afirmar que el PC constituye una forma general del pensamiento humano para resolver problemas; es decir, un conjunto de capacidades amplias aplicables a diversas tareas, como la representación abstracta de un problema, su descomposición en partes y su resolución mediante un procedimiento algorítmico.

En este punto, vale la pena detenerse en lo que subyace a esta ambigüedad. En un trabajo posterior, Wing (2011) retoma el

problema y define el PC como el conjunto de procesos de pensamiento involucrados en la formulación de problemas y sus soluciones, de modo que estos puedan representarse en una forma que permita su ejecución efectiva por un agente que procesa información. Esta sugerente definición permite aislar el problema central –ya señalado en la introducción–, a saber, el rol de la automatización en relación con el PC.

El PC involucra la capacidad de representar un problema y desarrollar su solución tal que la misma sea susceptible de ser ejecutada por un agente. Sin embargo, Wing destaca que dicho agente puede ser humano o artificial; es decir, se trataría de un agente capaz de ejecutar la solución, aunque no necesariamente una computadora. Por tanto, al menos parte de la ambigüedad surge del grado de refinamiento requerido para lo que entendemos por una solución a un problema y, desde luego, del vínculo de la misma con la automatización.

Así, la definición de Wing es amplia; por tanto, permite entender que el grado de refinamiento requerido dependerá de las características del agente que vaya a ejecutar la solución. En este sentido, el lenguaje en el que representemos el problema y escribamos la solución estará relacionado con las características del agente que la ejecutará.

Sin embargo, no hay consenso sobre esta perspectiva; en particular, Denning y Tedre (2021) sostienen que el PC implica un conjunto de habilidades y prácticas orientadas al diseño de procesos computacionales capaces de hacer que las computadoras realicen tareas por nosotros. Por tanto, en su opinión, la automatización es constitutiva del PC.

Así, la ambigüedad en torno al PC puede comprenderse con mayor claridad como una diferencia en torno al papel que cumple la automatización en la definición misma del PC; en última instancia, al lugar que ocupa la automatización en su conceptualización.

Por otra parte, podemos considerar una gradación que va desde un extremo, en el que escribimos una solución en lenguaje natural para que sea ejecutada por una persona, hasta el otro, en el que codificamos dicha solución en un lenguaje de programación para que sea ejecutada automáticamente por una computadora. En la actualidad, el surgimiento de inteligencias artificiales capaces de procesar instrucciones en lenguaje natural ha dado lugar a casos intermedios que, por el momento, no se encuentran claramente conceptualizados y que merecerían un análisis más detenido en otra oportunidad.

Dicho esto, el PC involucraría capacidades cognitivas altamente generales, como representar un problema de un modo abstracto y escribir una solución detallada para que sea ejecutada por un agente. Sin embargo, hay una enorme variedad tanto en el nivel de

refinamiento que se espera de la solución como en el lenguaje (o medios de representación) en el que ésta debería escribirse, en función de los requerimientos del agente.

De este modo, si entendemos como parte del PC capacidades como representar un problema, dividirlo en partes, escribir una solución detallada, etc., las mismas podrían situarse dentro de una gradualidad que va de lo informal a lo estrictamente formal, es decir, desde la representación del problema mediante herramientas conceptuales informales, como una tabla o un diagrama, hasta la escritura de la solución en un pseudocódigo y, finalmente, una solución que involucre estructuras de datos y un lenguaje de programación. Ahora bien, un mismo problema podría resolverse en distintos niveles de abstracción y especificidad, aunque involucrando las mismas habilidades conceptuales generales, las cuales son parte del PC en distintos niveles de formalidad y rigor.

De este modo, los dos sentidos que encontramos en el uso de la expresión PC se podrían distinguir mediante el nivel de formalidad que el mismo involucra, tanto en la representación del problema como en la formulación de su solución.

En síntesis, vamos a decir que cuando hablamos de *PC en un sentido informal*, estamos pensando en las capacidades generales que aplicamos en la representación de un problema y su solución, sin que esto implique el uso de conceptos técnicos ni, en particular, una resolución automatizada del problema. Cuando nos referimos al *PC en un sentido técnico*, estamos pensando, por ejemplo, en la capacidad para representar mediante estructuras de datos un problema y ofrecer soluciones codificadas en un lenguaje de programación, es decir, estamos considerando una solución automatizada al problema. La diferencia entre ambos no solo se plantea en términos del rol de la automatización, sino que también está en el nivel de formalidad o rigor con el que estamos trabajando, es decir, en el nivel de formalidad requerido para la solución a un problema. ¿De qué depende este nivel de rigor? Por un lado, como hemos dicho, está vinculado al agente que ejecutará la solución. Por otro lado, depende del contexto en el que se plantea el problema, a partir del cual se delimita lo que se entiende por una solución adecuada en términos computacionales.

Contextos, niveles de formalización y variaciones del pensamiento computacional

Como señalamos al final de la sección anterior, el nivel de formalidad o rigor requerido para formular una solución computacional dependerá en gran medida del contexto en el que se plantea el problema y del rol que tenga la automatización en su resolución. Ahora bien, para defender la transversalidad del PC –es

decir, que aparece en cualquier contexto—, es necesario sostener que lo aplicamos incluso en actividades cotidianas, donde no utilizamos conceptos técnicos como lenguajes de programación o estructuras de datos. De este modo, podemos decir que aplicamos el PC —por ejemplo— cuando buscamos el camino más corto que pasa por tres puntos de una ciudad, así como también en un laboratorio científico, donde precisamos construir un modelo computacional sofisticado para una molécula.

Incluso dentro de una misma actividad —como puede ser la enseñanza de algoritmos básicos—, el tipo de representación utilizada (lenguaje natural, pseudocódigo, diagramas de flujo, lenguaje de programación) varía de acuerdo a los objetivos de la misma en un momento dado. Por ejemplo, comprender un algoritmo escrito en lenguaje natural o en un pseudocódigo básico podría ser suficiente en un determinado momento, para luego pasar a los problemas que se introducen al codificar el mismo en un lenguaje de programación. Es por este tipo de razones que la automatización se suele excluir de la definición de PC, sobre todo cuando se busca ajustarla a las prácticas educativas en edades tempranas.

En esta perspectiva, Grover y Pea (2018) reconocen que una parte esencial del PC, tanto en la informática como en otros ámbitos, consiste en trabajar hacia soluciones que puedan ser ejecutadas por una máquina. Sin embargo, los autores matizan el rol de la automatización en edades tempranas al señalar que, en el nivel escolar, si bien el objetivo final del PC no siempre es una solución computacional implementada en un dispositivo, es importante que los estudiantes comprendan cuándo la automatización constituye una respuesta adecuada al problema y qué aspectos conviene resolver mediante la intervención humana o mediante las máquinas.

Así, en el ámbito educativo podría ser suficiente —al menos en ciertos casos— con comprender los conceptos clave, como la elaboración de un modelo del problema o la construcción de un algoritmo que ofrezca una resolución estructurada, dejando planteada una orientación hacia la automatización y haciendo visible el papel fundamental que esta adquiere cuando los problemas escalan en complejidad.

Por supuesto, en la medida en que los problemas requieren una resolución más técnica, surgen también los desafíos propios de la automatización, como el tiempo que una computadora demoraría en resolverlos, lo que da lugar a una discusión teórica de nivel más alto, vinculada con la complejidad algorítmica.

Siguiendo esta idea, diremos que un problema puede resolverse aplicando métodos computacionales según distintos requerimientos de rigor o formalidad, los cuales están determinados, al menos parcialmente, por el contexto y por el rol que tiene la automatización

en dicho contexto. Por ejemplo, si consideramos el problema (P): encontrar el recorrido más corto que pasa por tres puntos en una ciudad; podríamos decir que el mismo puede resolverse aplicando el PC en distintos niveles de rigor o formalidad, que van desde un abordaje que podría ser adecuado a nivel escolar, hasta un enfoque que podría involucrar conceptos sofisticados propios de la teoría de la complejidad computacional.

Este fenómeno, a saber, la variación en lo que se entiende por una solución rigurosa a un problema, tampoco es exclusivo del PC, ya que también podemos encontrar situaciones similares en otras prácticas científicas. En particular, existe un amplio debate en matemáticas en relación con la noción de prueba matemática, que ha dado lugar a una profunda reflexión sobre la relación entre pruebas informales y pruebas formales, en sentido estricto. De esta manera, podemos considerar la diferencia que existe entre ofrecer una demostración rigurosa en lenguaje natural y ofrecer la misma demostración codificada en un sistema formal. Ahora bien, desde luego los matemáticos aplican el pensamiento matemático, o el razonamiento matemático, tanto cuando ofrecen una prueba informal como cuando ofrecen una prueba absolutamente rigurosa en un sistema formal.

Esta diferencia en términos de rigor puede encontrarse en la práctica matemática actual y también en la comparación entre distintos momentos históricos de la disciplina. De este modo, una de las motivaciones del pensamiento matemático ha sido la de lograr mayores niveles de rigor y control en relación con los conceptos que se utilizan. A este respecto, Seoane (2017) señala que la exigencia de mayor rigor es relativa al contexto histórico, dado que, según la época, dicho rigor puede manifestarse de diferentes maneras.

Ahora bien, en relación con el PC se presenta un problema propio: lo que motiva la exigencia del rigor y el llegar a un alto nivel de formalización del problema no es meramente una exigencia de rigor radical sustentada en algún ideal epistémico, como puede ocurrir en las matemáticas, sino que dicha exigencia se sustenta en la posibilidad de computar la solución, es decir, en la posibilidad misma de ofrecer una solución que pueda ser ejecutada por una computadora. Esto coloca en primer plano la relación entre niveles de formalización y el agente que ejecuta la solución.

En síntesis, el mismo problema computacional podría abordarse con distintos niveles de rigor. Sin embargo, deberíamos tener en cuenta que en todo ese espectro, que va de una solución informal que recurre a ideas intuitivas, hasta un programa sofisticado que se desarrolla sobre la base de una teoría matemática, estamos aplicando el PC. Por tanto, desde luego que el PC en sentido informal puede aparecer y estimularse desde una edad temprana; no obstante, cuando el problema adquiere una formalización precisa capaz de ser ejecutada

por una computadora, estamos frente al PC en un sentido técnico. De este modo, podemos considerar un cierto tipo de gradualidad entre el PC informal y técnico.

En este último caso también podemos encontrar matices. Por ejemplo, una solución técnica a un problema computacional puede estar sustentada en sofisticadas teorías matemáticas. Sin embargo, podemos ser capaces de implementar una solución compleja, que involucra conceptos matemáticos sofisticados, sin ser capaces de entender en profundidad la teoría matemática que está en juego. Por tanto, el PC, en este sentido técnico, podría requerir de mayores niveles de implicación con las teorías matemáticas de fondo. De este modo, la dimensión técnica de un problema se puede volver cada vez más teórica.

Por otra parte, en la medida en que un problema es abordado mediante un enfoque computacional de alto nivel, es decir, mediante un pensamiento técnico que involucra aspectos teóricos, aparecen nuevos problemas que no estaban en el horizonte del problema inicial. Así pues, el PC en sentido técnico introduce problemas que le son propios, o dicho de otro modo, en cierto nivel se discute sobre los problemas en abstracto, perdiendo paulatinamente los elementos intuitivos que aparecen en las formulaciones más básicas. Por ejemplo, el recorrido sobre un mapa se convierte en un problema de optimización o de complejidad computacional sobre grafos, y las soluciones algorítmicas están íntimamente relacionadas con teorías de fondo que ofrecen el soporte conceptual de las posibles soluciones.

Un marco para establecer la distinción entre pensamiento computacional informal y pensamiento computacional técnico

En esta sección ofreceremos el esbozo de un marco a partir del cual podremos distinguir con más cuidado la diferencia entre PC informal y PC técnico. El punto central mediante el cual se establece la diferencia entre ambos está en los medios (en sentido amplio) a partir de los cuales se representa y desarrolla la solución a un problema planteado. Esta diferencia, por otra parte, está vinculada con el tipo de agente que va a ejecutar la solución, a saber, si se trata de un agente artificial o humano.

Siguiendo esta idea, podemos plantear un esquema con tres componentes:

1. La descripción de un problema (P) que requiere de una solución computacional,

2. los medios (M) mediante los cuales representamos el problema y su solución, y
3. el agente (A) que va a ejecutar la misma.

Estos tres componentes permitirán enfatizar el tipo de PC que ponemos en obra en un contexto determinado. Por otra parte, cada uno de estos componentes es entendido en un sentido amplio, lo cual permite admitir distintas variantes dependiendo del caso. Por ejemplo, el problema (P) puede describirse en lenguaje natural, junto con una serie de elementos como diagramas, tablas, etc., donde está codificada información relevante. Por otra parte, no se excluye una formulación más técnica de un problema. Respecto de los medios (M) para representar y desarrollar la solución al problema, podríamos transitar una gradación que va desde el lenguaje natural, junto con diagramas y otros dispositivos para representar información, hasta un lenguaje de programación de alto nivel. Desde luego, en la transición entre esos dos extremos se encuentra la diferencia entre el PC informal y técnico.

Por último, el agente (A) puede ser entendido como una computadora, como una inteligencia artificial o como una persona. Desde luego, si el agente es una computadora en sentido lato, no podremos ofrecer una solución al problema en lenguaje natural. Por lo tanto, existe una estrecha relación entre el lenguaje de la solución y el agente que la ejecutará.

De este modo, en un contexto determinado, un problema (P) estará asociado al medio (M) de su representación y solución, y al agente (A) que ejecutará la misma. Así pues, las diferencias en los contextos impactarán en los tres componentes, dando lugar a distintas variantes de PC, en particular, las que hemos denominado como informal y técnica. De este modo, las diferencias graduales que hay entre el PC más general e informal –naturalmente asociado con etapas educativas tempranas– y el PC más técnico y avanzado, propio del ámbito profesional, se plantean dentro de un marco que permite enfatizar la continuidad, en lugar del hiato entre ambos, en función de los objetivos del contexto.

Ahora bien, considerando los tres componentes, la diferencia fundamental entre el PC informal y el PC técnico se encuentra en M, es decir, en los medios representacionales que utilizamos para especificar el problema y ofrecer una solución; paradigmáticamente en los lenguajes que utilizamos. Desde luego, el medio puede consistir en el lenguaje natural, quizás agregando algunos símbolos matemáticos, o puede involucrar diagramas, etc.; o también puede tratarse de un lenguaje de programación. En otras palabras, y siendo

un poco más acotados, podemos representar y resolver el problema en un lenguaje informal o en un lenguaje formal.

En relación con este problema, Rescorla (2014) señala que, cuando Knuth presenta el algoritmo que permite obtener el máximo común divisor de dos números naturales, lo hace en lenguaje natural, complementado con símbolos matemáticos. El autor observa que, durante la mayor parte de la historia humana, esta fue prácticamente la única manera de formular instrucciones mecánicas. Con la revolución informática, en cambio, surgieron los *formalismos computacionales* rigurosos, que permiten expresar dichas instrucciones de manera precisa, no ambigua y en una forma canónica.

En este sentido, los formalismos computacionales, como los llama Rescorla, ofrecen un nivel de rigor más estricto que el mero lenguaje natural, aunque no se trata solo de rigor. El otro problema que está en juego es la capacidad que tiene el lenguaje de vincularse con una ejecución mecánica. De este modo, los formalismos computacionales hacen posible codificar una solución a un problema y, al mismo tiempo, ejecutar dicha solución en una computadora. Por tanto, el lenguaje en que representamos y damos la solución a un problema está vinculado con el agente que la ejecutará (humano o artificial).

Finalmente, siguiendo la terminología de Rescorla, vamos a sintetizar el punto en relación con los medios (M) mediante los cuales resolvemos un problema, distinguiendo entre medios expresivos de propósito general (lenguaje natural, diagramas, tablas, etc.) y formalismos computacionales que permiten vincular la codificación del problema y su solución a una ejecución mecánica.

Por tanto, dicho en otras palabras, cuando la solución al problema se implementa en un medio expresivo de propósito general estamos frente al PC informal. En este caso, el agente que ejecute la solución deberá ser capaz de convertir un lenguaje informal en acciones específicas. Por otra parte, cuando la solución al problema se implementa en un formalismo computacional estamos frente al PC técnico. En este caso, el agente que ejecuta la solución podrá ser una computadora, capaz de convertir las instrucciones en acciones mecánicas.

Desde luego, este sintético marco es tan solo el esbozo de una estructura que nos permitiría distinguir con más cuidado dos sentidos del PC que, o bien están en disputa –como vimos a partir de las diferencias del enfoque de Denning y Tedre con el de Wing–, o bien coexisten acríticamente, lo cual puede propiciar confusiones.

Evaluación, interpretación y el rol del pensamiento informal

Como desarrollamos en la sección anterior, un problema susceptible de solución computacional aparece en un contexto determinado. El contexto delimitará lo que se entiende como una descripción del problema, el lenguaje en el que se codifica su representación y solución y el tipo de agente que va a ejecutar la misma; es decir, el rol de la automatización en relación con la solución que se proponga.

Ahora bien, en este apartado vamos a destacar un problema fundamental dentro del PC, tanto en su variante informal como en su variante técnica, aunque especialmente en esta última.

Al considerar la transición entre una manera informal de pensar un problema y una solución técnica sofisticada, no deberíamos perder de vista su articulación, es decir, la relación que existe entre ambas. En este sentido, Wing (2008) sostiene que la esencia del PC radica en la abstracción y subraya que, al trabajar con abstracciones complejas, resulta fundamental definir la “abstracción adecuada”, puesto que el proceso mismo de abstracción –decidir qué detalles destacar y cuáles dejar de lado– constituye el núcleo del PC.

Siguiendo a Wing, los procesos de abstracción subyacen a lo que entendemos por una solución adecuada a un determinado problema. Entonces, ¿por qué una solución es correcta o incorrecta?

En primer lugar, la transición entre un modo informal de pensar computacionalmente y un modo técnico también se plantea en términos del grado de abstracción con el que representamos el problema y su solución.^[4] Por otra parte, en la medida en que el problema se representa con mayor rigor debemos tomar decisiones que están articuladas con los niveles de menor abstracción, donde el problema está siendo representado a nivel informal.

Planteando la situación esquemáticamente y en un contexto dado, nos enfrentamos a un problema sobre el cual podemos aplicar una solución computacional (P). Por ejemplo, encontrar un recorrido en auto por una ciudad, saliendo de un punto y llegando a otro, pasando por tres puntos específicos, en el menor tiempo posible.

En un primer momento, el problema se plantea de manera informal: contamos con una descripción en lenguaje natural de lo que buscamos resolver y, probablemente, con diagramas que representan cierta información, entre otros recursos. En esta instancia discutimos el problema, esbozamos posibles algoritmos en pseudocódigo y lo subdividimos en partes, etc. Podemos decir que estamos aplicando el PC en sentido informal y que los medios representacionales empleados también lo son; es decir, aún no se trata de una solución implementada en un lenguaje de programación, aunque exista una orientación general hacia la automatización.

En una segunda etapa, ya con el objetivo de ofrecer una solución computacional capaz de ser ejecutada por una computadora,

podríamos pensar en el lenguaje de programación que vamos a utilizar para resolver el problema, en las estructuras de datos que son requeridas para representar el mismo y en los programas específicos que codifican los distintos algoritmos que son necesarios en la solución. Aquí, el lenguaje en el que representamos el problema y ofrecemos su solución es un lenguaje de programación, es decir, se trata de un formalismo computacional.

Ahora bien, la articulación entre ambas etapas, es decir, entre estos distintos niveles de abstracción, rigor y formalización, tiene un rol epistémico central en la comprensión de la solución que se propone al problema. Destaquemos dos roles fundamentales.

Por un lado, cuando se resuelve un problema a nivel técnico es importante recuperar el vínculo con la descripción informal del problema, e incluso, con ciertos esbozos de la solución que se construyeron en etapas anteriores a su codificación, para reconocer en la solución técnica (es decir, en las estructuras y programas) la solución al problema planteado originalmente. Sin este vínculo no podríamos determinar si la solución técnica resuelve el problema planteado.

Por otro lado, los procedimientos mediante los cuales corregimos la solución a un problema están vinculados a la comprensión ordinaria del problema que pretendemos resolver, al menos en dos dimensiones fundamentales: la descripción del problema y el ajuste del modelo respecto de dicha descripción. En este sentido, los procesos de abstracción operan constantemente sobre la brecha que separa el problema real de su representación y resolución.

Dicho de otro modo, tanto la formulación inicial del problema como la evaluación de la adecuación del modelo forman parte de prácticas interpretativas en las que deliberamos qué aspectos del problema deben ser destacados y capturados. Lo que queremos subrayar es que estos procesos de ajuste dependen, de manera irreductible, de formas de análisis informal tales como la descripción y la comprensión del problema que buscamos resolver, así como de procedimientos para evaluar la solución.

Ahora bien, en la medida en que consideramos que los procesos de abstracción subyacen al PC, tanto en su variante informal como técnica, debemos concluir que, desde esta perspectiva, los procesos de análisis informal que involucran habilidades generales –como la capacidad de describir con precisión un problema en el lenguaje natural–, son tan importantes como la representación de un problema mediante un modelo formal.

Aunque esto último puede parecer obvio, lo que no es tan evidente es que sin este tipo de herramientas de análisis no seríamos capaces de evaluar propiamente una solución, o dicho de otra manera, no tendríamos una robusta articulación entre los problemas reales y las

soluciones que se proponen. Y precisamente de esta articulación depende que el PC supere un rol meramente instrumental para adquirir una función epistémica significativa.

Conclusiones: hacia una perspectiva crítica sobre el pensamiento computacional

En este trabajo hemos mostrado que lo que se entiende por PC oscila entre un sentido general e informal –asociado a habilidades cognitivas transversales como la abstracción o la resolución estructurada de problemas– y un sentido técnico, vinculado al uso de formalismos computacionales y a la automatización de soluciones. Esta ambigüedad, si no se analiza críticamente, puede generar confusiones tanto en el plano teórico como en el educativo.

Una de las razones que explica esta oscilación se encuentra, al menos parcialmente, en que los conceptos propios de este tipo de pensamiento operan en distintos niveles de abstracción y con criterios de rigor variables, que dependen del contexto y del rol de la automatización. En este sentido, un mismo problema puede recibir una solución computacional en distintos niveles de abstracción, formalidad y elaboración técnica. Como sugerimos anteriormente, el problema de determinar el recorrido más corto que pasa por tres puntos en una ciudad podría dar lugar a una solución computacional de nivel escolar o a una solución técnica altamente sofisticada, sobre todo si integramos en el problema más información, restricciones, la generalización de la solución, etc.

Esta característica del PC conserva cierta analogía con lo que ocurre en otras disciplinas. Si pensamos en la física, podríamos considerar que ocurre un fenómeno similar cuando se plantean soluciones a un problema con distintos niveles de sofisticación y refinamiento, en la medida en que el problema se plantea de un modo más realista.

Ahora bien, hay dos características que hacen del PC un caso especial. Por un lado, su enorme versatilidad, es decir, su capacidad para abordar problemas de contextos muy distintos, además de su capacidad para elaborarlos en distintos niveles de complejidad. Esto se debe, fundamentalmente, a que en esencia el PC está vinculado a capacidades muy generales de abstracción, representación y resolución estructurada, que son aplicables en cualquier dominio.

Por otra parte, los esfuerzos por resolver un problema desde un enfoque computacional están orientados a la automatización, incluso desde su expresión más informal. Esto significa que, si bien no es necesario en una determinada etapa llegar a utilizar formalismos computacionales para desarrollar el PC, la orientación a la automatización marca un interés creciente por incrementar el nivel de

formalidad de las soluciones que se proponen. Este vector hacia la formalización y hacia el rigor está justificado por una *necesidad práctica* que, de algún modo, dio origen a la propia disciplina, a saber, la necesidad de automatización. En este sentido, dicha necesidad produce un movimiento desde el PC informal hacia formas más técnicas de PC.

En esta posición gradualista se plantea el rol de la automatización según el contexto, reafirmando la continuidad entre formas de PC más generales y formas más técnicas, lo cual puede contribuir a reflejar el PC en etapas educativas distintas, además de tener implicaciones de carácter histórico. Algunos autores –como Denning y Tedre– reconocen que muchas ideas asociadas al PC existían antes de que fuera posible una ejecución automática, sin embargo, lo definen como vinculado a la automatización. Esta delimitación técnica del PC introduce, de forma implícita, una ruptura en la continuidad histórica que lo relaciona con formas de pensamiento algorítmico que aparecieron antes del surgimiento de las computadoras.

Por último, como hemos destacado, los procesos de análisis mediante los cuales somos capaces de evaluar una solución siguen estando inmersos en las capacidades cognitivas más generales posibles: interpretar, comprender, deliberar qué es lo relevante, etc. Desde luego, estas capacidades también se pueden ver beneficiadas al exponerse a la dificultad de intentar resolver un problema de un modo riguroso, mediante conceptos que forman parte del repertorio de las ciencias de la computación. Sin embargo, se trata de capacidades que, de algún modo, se sustraen a la construcción de los modelos y a las posibles soluciones, ya que viven en la interacción entre los problemas reales que pretendemos resolver y sus soluciones. Se trata, propiamente, de las capacidades cognitivas que nos permiten evaluar críticamente las herramientas, las soluciones propuestas, comprender realmente los problemas y su articulación con las soluciones. Dicho de otro modo, es lo que nos permite elaborar criterios de corrección respecto de lo que hacemos. Estas capacidades no deberían omitirse de lo que se entiende por PC.

Referencias

- Denning, P. J., & Tedre, M. (2021). Computational thinking: A disciplinary perspective. *Informatics in Education*, 20(1), 361-390. <https://doi.org/gmvbk4>
- Floridi, L. (2008). The method of levels of abstraction. *Minds and Machines*, 18(3), 303-329. <https://doi.org/dk3pwf>
- Grover, S., & Pea, R. (2018). Computational thinking: A competency whose time has come. En S. Sentance, E. Barendsen, & C. Schulte (Eds.), *Computer science education: Perspectives on teaching and learning in school* (pp. 19-37). Bloomsbury Academic.
- Moor, J. H. (1978). Three myths of computer science. *The British Journal for the Philosophy of Science*, 29(3), 213-222. <https://doi.org/ctbpqk>
- Primiero, G. (2016). Information in the philosophy of computer science. En L. Floridi (Ed.), *The Routledge handbook of philosophy of information* (pp. 90-106). Routledge.
- Rescorla, M. (2014). The computational theory of mind. *Philosophy Compass*, 9(2), 127-138.
- Seoane, J. (2017). On mathematical elucidation. *Revista Portuguesa de Filosofia*, 73(3-4), 1405-1422. <https://doi.org/pzc3>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. <https://doi.org/fd3h5w>
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717-3725. <https://doi.org/c24v3c>
- Wing, J. M. (2011). Research notebook: computational thinking-What and why? *The Link Magazine*, 6, 20-23.

Contribución de las/os autoras/es (CRediT)

Nombre de las/os autoras/es	Colaboración Académica													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Alejandro Chmiel	x	x	x	x	x	x	x	x	x	x	x	x	x	x

1- Administración del proyecto; 2- Adquisición de fondos; 3- Análisis formal; 4- Conceptualización; 5- Curaduría de datos; 6- Escritura – revisión/edición; 7- Investigación; 8- Metodología; 9- Recursos; 10- Redacción – borrador original; 11- Software; 12- Supervisión; 13- Validación; 14- Visualización.

Notas

- [1] Especialmente en aquellas que integran el ámbito de la Ciencia, Tecnología, Ingeniería y Matemáticas (*Science, Technology, Engineering and Mathematics* [STEM]).
- [2] Véase <https://www.csunplugged.org/en/>
- [3] Véase Moor (1978), donde enfatiza que los programas requieren de una realización física.
- [4] La discusión sobre niveles de abstracción en computación es sumamente importante. Véase Floridi (2008) y Primiero (2016).

Información adicional

Cómo citar: Chmiel, A. (2025). Pensar y computar: hacia una crítica conceptual al pensamiento computacional. *Revista IRICE*, 49, e2093. <https://doi.org/10.35305/revistairice.vi49.2093>

AmeliCA

Disponible en:

<https://portal.amelica.org/amelia/ameli/journal/746/7465386012/7465386012.pdf>

Cómo citar el artículo

Número completo

Más información del artículo

Página de la revista en portal.amelica.org

AmeliCA

Ciencia Abierta para el Bien Común

Alejandro Chmiel

Pensar y computar: hacia una crítica conceptual al pensamiento computacional
Thinking and computing: toward a conceptual critique of computational thinking

Revista IRICE

núm. 49, e2093, 2025

Consejo Nacional de Investigaciones Científicas y Técnicas,
Argentina
revista@irice-conicet.gov.ar

ISSN-E: 2618-4052

DOI: <https://doi.org/10.35305/revistairice.vi49.2093>