



## LA COMPUTACIÓN EN EL MARCO DE UN NUEVO PROYECTO HUMANISTA: RELEYENDO A DIJKSTRA Y SIMONDON

*Computing in the framework of a new humanist project: Rereading to Dijkstra and Simondon*

**Aldana D'Andrea**  
UNRC  
**Emiliano Campoamor**  
UPC/UNRC

**Resumen:** En este trabajo intentamos explorar el vínculo entre la noción de computación de Edsger Dijkstra y la propuesta ontológica relacional de Gilbert Simondon. Proponemos abordar la noción de computación de Dijkstra a partir de la revisión de su artículo “On the cruelty of really teaching computing science” con una clave de lectura particular; sugerimos que el enfoque de Dijkstra puede comprenderse a partir de un planteo relacional de la computación que implica la disolución de una serie de pares de opuestos clásicos (humana/máquina, particular/general, concreto/abstracto) y que bajo esta interpretación su enfoque puede ser leído en el marco de la filosofía de Simondon como un camino posible hacia la desalienación de la relación entre humanas y técnica y una invitación a pensar a partir de una ontología relacional y dinámica. Finalizamos con el planteo de la hipótesis de que la computación tal como la entendemos a partir de Dijkstra podría significar una forma de individuación colectiva en la ontología simondoniana.

**Palabras claves:** Computación, Humanismo, Dijkstra, Simondon, Individuación

**Abstract:** In this work we try to explore the link between Edsger Dijkstra's notion of computing and Gilbert Simondon's relational ontological proposal. We propose to approach Dijkstra's notion of computing from the revision of his paper “On the cruelty of really teaching computing science” with a particular reading key; we suggest that Dijkstra's approach can be understood from a relational approach to computing that involves the dissolution of a series of pairs of classical opposites (human/machine, particular/general, concrete/abstract) and that under this interpretation his approach can be read within the framework of Simondon's philosophy as a possible path towards the de-alienation of the relationship between the human and the technical and an invitation to think from a relational and dynamic ontology. We end with the hypothesis that computing as we understand it from Dijkstra could mean a form of collective individuation in simondonian ontology.

**Keywords:** Computing, Humanism, Dijkstra, Simondon, Individuation.

### 1. Introduciendo a Simondon

Este trabajo intenta enmarcarse en una apuesta integral que propuso Simondon a mediados del siglo pasado buscando reintegrar la tecnología a la cultura. La apuesta simondoniana es expresada claramente en sus dos tesis doctorales. En *La Individuación a la Luz de las nociones de Forma y de Información*<sup>1</sup>, su tesis principal, Simondon hace una crítica a la metafísica tradicional y plantea que cualquier principio ontológico que intente sostenerse no puede nunca basarse en el abordaje de individuos (biológicos,

<sup>1</sup> SIMONDON, Gilbert. *La individuación a la luz de las nociones de forma e información*. Buenos Aires: Cactus, 2015.

físicos, psíquicos, sociales) ya definidos sino que debe tomar su base en los procesos que llevan a constituir esas individuaciones. La ontología nunca es un proceso que determina un estado definitivo, sino que es un conjunto de relaciones inestables en constante evolución y adaptación. Luego, en *El Modo de Existencia de los Objetos Técnicos*<sup>2</sup>, su tesis auxiliar, define a los objetos técnicos como parte de esa ontología y como parte también de una interacción compleja entre los entornos naturales y psico-sociales, sosteniendo que la técnica no es un mero abordaje instrumental ni una forma de dominación sino una co-constitución de lo humano con esos entornos. A diferencia de cómo la tradición filosófica ha tratado a la técnica y sus productos, esto es como una simple exteriorización del pensamiento, y de cómo la cultura se ha convertido en un sistema de defensa ante esto, Simondon sostiene que cultura y técnica se retroalimentan co-evolutivamente y que resulta perjudicial para una sociedad suponerlas separadas: la cultura es una técnica. Este problema conlleva consecuencias políticas e ideológicas, el propio Simondon considera que nuestra relación con los entornos técnicos es propia de un vínculo de alienación, de extrañeza, hasta tal punto que si la propuesta marxista de una colectivización de los medios producción apareciera en el horizonte de posibilidades nunca la alcanzaríamos si primero no superamos nuestra alienación con la máquina. En sociedades pre-industriales la rivalidad entre cultura y técnica se encuentra soslayada porque la técnica tiene un campo de acción intra-cultural, se mantiene dentro de ciertos márgenes y la cultura dicta los valores y creencias, pero cuando una sociedad evoluciona hacia desarrollos industriales el poder de la técnica sobrepasa el ámbito de la cultura, entonces esta tiende a tomar una posición defensiva, conservadora. Una técnica que deviene en procesos transculturales amenaza a las culturas que coloniza y lo que Simondon intentó poner en relieve es que la cultura no es ni más ni menos que un tipo de técnica: la técnica de crianza del ser humano por el ser humano<sup>3</sup>.

Para buscar una analogía que sirva de introducción con el trabajo de Simondon, podríamos decir que este guarda una estrecha relación con el diagnóstico heideggeriano del lugar de la técnica en la historia de la humanidad pero, como nos sugieren Bardin y Menegalle<sup>4</sup>, Simondon en vez de abogar por un retorno a lo natural (rural) y a la figura soberana de un Filósofo-Poeta-Rey apuesta por un Filósofo-Ingeniero que asumiendo un rol de director de orquesta de los conjuntos técnicos concilie a la tecnología con la cultura.

Haciéndonos eco de este diagnóstico de Simondon, nuestra propuesta asume que el humanismo clásico ha pensado a la humana en términos de oposiciones fundamentales y esenciales, siendo claro hoy que una posición ontológica determinada supone una escala axiológica y política afín. En estos términos, la esencia (cualquier cosa que ella sea, cualquier propiedad que ella implique) es anterior a la existencia, y esa propiedad anterior e inmutable le confiere el poder de dominio y control de lo inferior: la naturaleza, los animales, otros seres humanos y, lo que nos interesa a nosotras rescatar: las máquinas, ese extremo opuesto y amenazador de la humanidad. Queremos decir que este humanismo constituido como rechazo a la técnica y sus productos sólo ha defendido a alguna humanidad; en efecto, como relata Canguilhem, retomando a Schuhl, en la filosofía antigua, la oposición entre ciencia y técnica, teoría y práctica, naturaleza y artificio, “es paralela a una jerarquía económica y política, la jerarquía en la ciudad del hombre libre y de los esclavos”<sup>5</sup>.

En este sentido, Simondon denuncia que este *humanismo fácil*, ha opuesto la máquina a la humana, ha marcado su discontinuidad, reduciendo la primera a la dimensión puramente utilitaria y servil, concibiéndola sólo como medio y negándole, por lo tanto, el acceso al mundo de los valores y de la libertad y la cultura. La máquina, en tanto esclava, se constituye en el extremo opuesto y amenazador de la humanidad. El *humanismo fácil* ofrece como corolario la oposición al progreso técnico y un pedido de

<sup>2</sup> SIMONDON, Gilbert. *El modo de existencia de los objetos técnicos*. Buenos Aires: Prometeo, 2008.

<sup>3</sup> SIMONDON, Gilbert. *Sobre la técnica*. Buenos Aires: Cactus, 2017.

<sup>4</sup> BARDIN, Andrea; MENEGALLE, Giovanni. “Introduction to Simondon”. En: *Radical Philosophy*, n. 189, 2015, pp. 15-16.

<sup>5</sup> CANGUILHEM, Georges. *El conocimiento de la vida*. Barcelona: Anagrama, 1976, pág. 124.

regreso a la naturaleza y, por ende, a la esencia, moderna manifestación de la antigua oposición y discontinuidad entre vida activa y vida contemplativa, entre lo servil y lo liberal.

Ante esta denuncia, la apuesta de Simondon aboga por una renovación del proyecto humanista, un humanismo que dé cuenta de los procesos de alienación propios de cada época; en este sentido el humanismo debe ser para Simondon un proyecto de emancipación y redescubrimiento de lo humano, debe suponer “la voluntad de llevar a un estatuto de libertad lo que fue alienado del ser humano, para que nada humano sea extraño al hombre” y dado que este “redescubrimiento de la realidad humana se puede operar en sentidos diferentes, y que cada época recrea un humanismo que siempre es, en cierta medida, apropiado para las circunstancias, porque apunta al aspecto más grave de la alienación que acarrea o produce una civilización”<sup>6</sup> nuestro trabajo intenta ser un aporte en dicha dirección.

### 1.1. Los procesos ontogenéticos de individuación

En el pensamiento de Simondon los procesos de individuación son la piedra angular de su ontología. Todo lo que llega a ser lo hace a través de un proceso de individuación, es la forma en que Simondon concibe cómo las cosas en el mundo adquieren realidad. Sin embargo, ésta no es una realidad fija y con aspectos esencialistas sino dinámica y relacional, el mismo proceso de individuación es en sí mismo una relación, es la relación. En efecto, su crítica a la tradición es que ésta siempre ha intentado concebir las nociones de individuación a partir de individuos ya formados en vez de poner el foco en los procesos que llevan a estas individualidades a ser lo que son. Vamos a intentar echar un poco más de luz sobre estas nociones y en la medida en que avancemos iremos vislumbrando algunas líneas fundamentales de la ontología simondoniana.

En principio, nos ceñiremos a dos tipos de procesos de individuación (hay más): la individuación física y la vital. Esto es la diferencia entre lo inerte y lo vivo. Simondon propone un monismo ontogenético por lo que los dos procesos son, en rigor, el mismo, luego veremos qué cambia en la individuación vital para conducir a un resultado diferente. Desde el comienzo se debe comprender que Simondon supone que en la realidad existen tres órdenes de magnitud diferentes. Un orden de magnitud superior, esto es la relación inter-elemental entre individuaciones ya alcanzadas, un orden de magnitud inferior, las relaciones intra-elementales de la materia y por último un orden de magnitud intermedio que es en donde deviene el individuo fruto de la comunicación entre los dos órdenes anteriores. En tanto que la individuación se da en el orden intermedio, el orden superior y el inferior pertenecen a una realidad pre-individual y la comunicación entre estos dos órdenes no tiene carácter necesario, por lo que todo proceso de individuación es un proceso contingente. Ahora bien, para poder continuar es pertinente introducir las nociones de energía potencial y metaestabilidad que Simondon retoma de la física y que servirán a las explicaciones subsiguientes. Que un sistema posea energía potencial significa que dicho sistema es pasible de ser transformado, y la propiedad de metaestabilidad está directamente relacionada con esta noción de energía potencial: un estado estable es propio de un sistema en equilibrio sin energía potencial, un sistema en estado inestable es aquel alejado del equilibrio y con cierta cantidad de energía potencial, y un estado metaestable es un estado en donde una mínima adición de energía externa al sistema desata un desequilibrio que luego deviene en un equilibrio.

Habiendo introducido estos elementos conceptuales podemos profundizar ahora sobre el ejemplo paradigmático<sup>7</sup> de Simondon sobre cómo se da la individuación física. El pensador francés nos sugiere remitirnos a la génesis de un cristal a partir de una materia amorfa (aguamadre), una materia no individuada dado que no posee una estructura

<sup>6</sup> SIMONDON, *op. cit.*, 2008, pág. 121.

<sup>7</sup> SIMONDON, *op. cit.*, 2015, pág. 110.

ordenada. Para que el proceso de individuación comience, la materia amorfa debe estar en un estado metaestable y en algún momento ser perturbada por un germen estructurante externo a ella. Dicho germen actuará de disparador para actualizar la energía potencial contenida en la materia amorfa, sin embargo, no cualquier perturbación externa puede actuar de disparador, debe existir entre materia y germen lo que Simondon llama una relación analógica. El germen comenzará a estructurar, a ordenar, una porción de la materia amorfa inmediata a él y las zonas que se van estructurando sirven de germen para estructurar a zonas contiguas no ordenadas en un proceso que continua hasta agotar toda la energía potencial contenida en la materia, o sea, el proceso de individuación es una amplificación del germen estructurante, operación que Simondon denomina transductiva:

Entendemos por transducción una operación física, biológica, mental, social, por la cual una actividad se propaga progresivamente en el interior de un dominio, fundando esta propagación sobre una estructuración del dominio operada aquí y allá: cada región de estructura constituida sirve de principio de constitución a la región siguiente, de modo que una modificación se extiende así progresivamente al mismo tiempo que dicha operación estructurante.<sup>8</sup>

En síntesis, el proceso de individuación física es la operación transductiva realizada por una singularidad (germen) en un dominio metaestable (materia amorfa) hasta agotar la energía potencial. Algo importante que se desprende de aquí es que para Simondon la individuación no se produce en el tiempo sino que crea la temporalidad misma: la actividad amplificante es el presente, la materia ya estructurada es el pasado y la materia por estructurar el futuro.

Ahora bien, así como Simondon no concibe a la individuación física a partir de individuos ya formados, tampoco seguirá ese camino con la individuación vital. Como habíamos mencionado anteriormente, el proceso de individuación física continuaba hasta alcanzar el equilibrio estable, esto es hasta agotar la energía potencial del dominio. Allí el ser ya individuado deviene en una degradación. El proceso de individuación vital seguirá un camino idéntico sólo que a diferencia de la individuación física, en vez de alcanzar estabilidad, alcanzará otro estado metaestable, similar al estado inicial, permitiéndole proseguir individuándose y habilitando una génesis continua de estructuraciones. El ser vivo sólo alcanza la estabilidad cuando perezca. Ciertamente, esto es posible porque en el proceso de individuación vital la comunicación entre los órdenes de magnitud que habíamos mencionado inicialmente es diferente, permitiendo estos que el ser vivo individuándose sea capaz de recibir energía de su medio asociado y así prolongar su metaestabilidad.

Como podemos observar, la propuesta de Simondon plantea un monismo ontogenético que resulta en una pluralidad ontológica, en otras palabras, los procesos de individuación son el mismo proceso sólo que de acuerdo a los diferentes estados que alcanzan generan diversos tipos de individuaciones. De allí podemos desprender, entonces, que para el filósofo francés existe una continuidad entre lo inerte y lo vivo.

## 2. Pensar la computación

Edsger Dijkstra es uno de los teóricos de la computación más importante del siglo XX. Sus trabajos han contribuido a desarrollar no solo el campo científico de la computación, sino también la reflexión sobre el estatus de la disciplina, su objeto de estudio, las metodologías empleadas y su enseñanza. En 1988 Dijkstra escribe el artículo *On the Cruelty of Really Teaching Computing Science*, plantea allí que la computación representa una *novedad radical* dentro de la ciencia, la técnica y la cultura.<sup>9</sup> La actitud

<sup>8</sup> *Ibid.*, pág. 38.

<sup>9</sup> DIJKSTRA, Edsger. On the cruelty of really teaching computing science. <http://www.cs.utexas.edu/users/EWD/ewd10xx/EWD1036.PDF> (último acceso 01/07/2020).

general frente a este tipo de novedades es, según este autor, intentar comprenderlas en su totalidad a partir de las experiencias y los conceptos pasados, con la consecuente supresión o negación de lo nuevo. Creemos necesario reconsiderar el reclamo y la propuesta que realiza Dijkstra en torno a la necesidad de avanzar en la comprensión de la computación como un fenómeno novedoso, ya que esta línea puede aportarnos algunas claves para pensar los desafíos y las potencialidades de la cultura técnica de nuestro momento histórico.

Un ámbito de pensamiento que ha evidenciado de manera paradigmática aquello que Dijkstra señala es el conformado por la discusión sobre el estatus de la ciencia de la computación, éste ha estado delineado en gran medida por el artículo pionero de Wegner<sup>10</sup> que plantea que la computación es una disciplina híbrida conformada a partir del encuentro de intereses y desarrollos pertenecientes a tres áreas de estudio: las ciencias empíricas, la lógica-matemática y la ingeniería. Esta línea tripartita sigue siendo sostenida y debatida en la actualidad por muchas científicas del área; por un lado se encuentran quienes afirman que la hibridez sigue siendo válida; por otro lado se encuentran quienes adscriben a una línea en particular y sostienen incluso que puede pensarse en una reducción de la computación a otra disciplina; finalmente, se encuentran quienes admiten que las tres tradiciones intelectuales originan el área de la ciencia de la computación, pero que a partir de ellas la disciplina no se conforma principalmente como un híbrido sino como una nueva ciencia con un carácter propio, el cual se constituye fundamentalmente a partir de las nociones de programa y programación.

Charles Hoare, en un artículo clásico en este debate, sostiene esta última línea de pensamiento:

¿cuál es el núcleo central de la disciplina? ¿Qué la distingue de las distintas disciplinas con las cuales se relaciona? (...) Mi respuesta a estas preguntas es simple - es el arte de programar una computadora. Es el arte de diseñar métodos eficientes y elegantes para conseguir que una computadora resuelva problemas, teóricos o prácticos, pequeños o grandes, simples o complejos.<sup>11</sup>

De acuerdo a este planteo, la identidad de la ciencia de la computación es irreductible a la de las disciplinas tradicionales porque la computación trata de la programación de una máquina –podemos pensar en una máquina abstracta o concreta- y este cometido central es propio de la ciencia de la computación. La línea propuesta por Hoare está en consonancia con el reclamo de Dijkstra, puesto que no intenta reducir la computación a sus antepasados -las máquinas calculadoras o a la noción de cálculo algorítmico- sino que comprende que a partir de ellos, pero al mismo tiempo superándolos, la computación se presenta como un nuevo fenómeno científico y técnico que desafía los marcos de pensamiento ontológicos, epistémicos y metodológicos tradicionales.

Esta forma de abordar el problema del estatus epistémico de la computación, centrado en la programación, es particularmente interesante para nuestro propósito de aportar a la recreación de un humanismo acorde a las exigencias de nuestro contexto. Es claro que la computación, en tanto concepto disciplinar y fenómeno técnico, es central para pensar la cultura de nuestro tiempo y el modo en que las humanas nos relacionamos con la técnica. Bajo la lógica del rechazo a las novedades que denuncia Dijkstra, la computación ha sido pensada bajo dos perspectivas excluyentes que han dominado tradicionalmente el pensamiento sobre la técnica: la computación como inteligencia superior que domina o limita la realidad humana, o la computación como instrumento dócil de la inteligencia humana. Ambas perspectivas, son, evidentemente, alienantes desde la perspectiva introducida por Simondon, por lo que elaborar un marco de comprensión

<sup>10</sup> WEGNER, Peter. "Research paradigms in computer science". En: Proceedings of the 2nd international Conference on Software Engineering. San Francisco: 1976, pp. 322-30.

<sup>11</sup> HOARE, Charles (1989). "Computer Science". En HOARE, Charles; JONES, Cliff (eds.). Essays in Computer Science. London: Prentice Hall, 1989, pág. 91. La traducción al español de este y los demás textos en inglés nos pertenece.

no alienante de la computación empieza a ser visualizado como fundamental para cualquier propuesta humanista que evite volverse contra su propio fin emancipatorio.

## 2.1. Dijkstra: la computación como relación

Proponemos abordar la noción de computación de Dijkstra a partir de la revisión del artículo mencionado con una clave de lectura particular; sugerimos que el enfoque de Dijkstra sobre la novedad que implica la computación puede comprenderse a partir de un planteo relacional que implica la disolución de una serie de pares de opuestos clásicos (humana/máquina, particular/general, concreto/abstracto) y que bajo esta interpretación, su planteo puede ser leído en el marco de la filosofía de Simondon como un camino posible hacia la desalienación de la relación entre humanas y técnica y una invitación a pensar a partir de una ontología relacional y dinámica.

### 2.1.1. Relación humana-máquina

Para Dijkstra la ciencia de la computación es una disciplina fundamentalmente formal o matemática, pues entiende que la actividad primordial es la manipulación simbólica; pero hay en la disciplina algo más que la matemática clásica ya que en este caso la manipulación “está relacionada con la interacción entre la manipulación mecánica y humana de símbolos, generalmente referidas como ‘computación’ y ‘programación’ respectivamente”<sup>12</sup>. La computación se comprende no solo como manipulación efectiva de símbolos o cálculo mecánico, sino como una manipulación mediada y constituida por la noción de programación. Así, mientras que la noción de cálculo efectivo o algoritmo parece ser suficiente para un paradigma lógico matemático de la computación, una perspectiva más acorde con la práctica de la disciplina y con el desarrollo del área, visualiza el vínculo de la noción de efectividad con la de programación como central ante las preguntas sobre qué es la computación y sobre qué trata la ciencia de la computación.

Dijkstra rechaza en primer lugar el paradigma ingenieril sobre la computación, esto es que la ciencia de la computación trate de máquinas computadoras o sistemas computacionales físicos, pues sostiene que “lo único que las computadoras pueden hacer por nosotros es manipular símbolos y producir los resultados de tales manipulaciones”<sup>13</sup>, en este sentido, la computadora funciona como una máquina calculadora tradicional, donde todo el proceso se explica según el mecanismo que produce la acción. Para que la máquina computadora posea alguna singularidad conceptual y técnica frente a la máquina calculadora –más allá de la potencia de cálculo- y pueda ser percibida como lo que Dijkstra llama una *novedad radical*, es necesaria además la noción de programación o manipulación humana de símbolos, de donde resulta la noción de código como aquello que posibilita la relación humana-máquina y que habilita un planteo relacional antes que dualista: “Y ahora el círculo está cerrado: construimos nuestras manipuladoras mecánicas de símbolos por medio de la manipulación humana de símbolos”<sup>14</sup>. Como se trata de un círculo, es importante apreciar que la manipulación mecánica permite, a su vez, el acceso humano a dominios no intuitivos, más abstractos y computacionalmente más complejos y, en este sentido, Dijkstra recuerda repetidamente que el desafío central de la ciencia de la computación consiste en no perderse en estas complejidades alcanzadas.<sup>15</sup>

Notemos que esta propuesta de Dijkstra solo se entiende superando la oposición tradicional humana/máquina en pos de alguna noción de acoplamiento o, en términos del

<sup>12</sup> *Ibíd.*, pp. 16 - 17.

<sup>13</sup> *Ibíd.*, pág. 15.

<sup>14</sup> *Ibíd.*, pág. 16.

<sup>15</sup> Véase: DIJKSTRA, Edsger. “The threats to computing science”. En: ACM 1984 South Central Regional Conference. Austin: 1984, pp., 16–18; DIJKSTRA, Edsger. “On a cultural gap”. En: *The Mathematical Intelligencer*, v. 8, n.1, 1986, pp. 48-58.

autor, de la interrelación entre las concepciones de proceso humano-proceso mecánico, de donde resulta el fenómeno de la computación y desde donde se evidencia, a su vez, la necesidad de una disciplina que tome a este fenómeno relacional como su objeto de estudio y desarrollo. Esta disciplina es para Dijkstra heredera de la lógica y de la matemática, pero las trasciende en tanto que mediante la interrelación humana-máquina ella tiene la posibilidad de realizar una parte del sueño de Leibniz:

Espero que a largo plazo la ciencia de la computación trascienda a sus disciplinas madres, matemática y lógica, haciendo realidad efectivamente una parte significativa del Sueño de Leibniz de proporcionar un cálculo simbólico como una alternativa al razonamiento humano (por favor, note la diferencia entre 'imitar' y 'proporcionar una alternativa': a las alternativas se les permite ser mejores).<sup>16</sup>

Hay otras autoras que señalan que la historia de la computación sigue esta línea leibniziana,<sup>17</sup> lo interesante de la propuesta de Dijkstra es que su lectura no sugiere la tradicional contraposición –ontológica, epistémica y política- entre el desarrollo de un cálculo mecánico y el razonamiento humano como procesos pre-definidos y estables que devienen de entes con las mismas características, sino que precisamente en la interrelación entre ellos la computación se vuelve, no autónoma ni automática, sino expresión de la relación co-constitutiva entre máquina y humana mediante la producción de un código que expresa y posibilita este vínculo. Lo que queremos notar es que Dijkstra evidencia que la computación no es simple ensamblaje humano y procesamiento maquínico de un código previamente producido, sino que el código se produce en la relación y, como en Simondon, la relación produce la temporalidad en la que la computación se desarrolla a partir de la relación código, humana y máquina. La relación, como veremos, no tienen la necesidad de responder a estados estables definidos previamente –no es necesario *imitar*, recuerda Dijkstra-, sino que la indeterminación, la contingencia y la metaestabilidad puede ser las características que mejor describan el despliegue de la potencialidad ofrecida por un proceso computacional.

### 2.1.2 Relación general-particular

La superación de la oposición humana/máquina en una noción de computación conformada mediante la noción de programación permite a Dijkstra replantear otros pares de opuestos clásicos en el pensamiento occidental mediante una apuesta conceptual sobre lo que es un programa: “Podemos ver a un programa como aquello que transforma la computadora de propósito general en un manipulador de símbolos de propósito particular, y hacerlo sin la necesidad de cambiar un solo cable”<sup>18</sup>.

Dijkstra entiende al programa como aquella entidad que determina qué tipo de máquina es una máquina computadora a partir de la definición de su propósito o finalidad, planteando así la posibilidad de repensar las categorías de propósito particular y de propósito general, empleadas tradicionalmente para caracterizar a la máquina y a la razón humana respectivamente, pero ahora propuestas para caracterizar una máquina singular: la máquina de programa almacenado. Podemos pensar, pues, en la computadora como una máquina de propósito general que logra cumplir propósitos particulares porque posee un programa intérprete –una máquina de Turing universal- que permite la interrelación efectiva entre lo universal y lo particular, y ello, a su vez, a partir de la interrelación no (necesariamente) efectiva entre humana y máquina.

Aquí el concepto de programa intérprete se vuelve fundamental para la programabilidad de los sistemas y la concepción relacional de la computación. Podemos

<sup>16</sup> DIJKSTRA, *op.cit.*, 1988, pág. 17.

<sup>17</sup> Véase, por ejemplo, DAVIS, Martin. *The universal computer: the Road from Leibniz to Turing*. New York: W. W. Norton & Company, 2000.

<sup>18</sup> *Ibíd.*, pp. 15.

entender a un intérprete como un programa que toma como dato a otro programa y su *input*  $p(i)$  y se comporta como se comporta ese programa para ese *input*,  $p(i)$  empieza siendo un dato, luego es una prescripción para el intérprete. A la programabilidad, por su parte, podemos expresarla como la capacidad de un sistema de comportarse de modos sustancialmente distintos de acuerdo a un *input* específico. Observamos que ambas nociones son, en el sentido aquí expuesto, interdependientes: todo sistema programable supone un intérprete, todo sistema que posee un intérprete es programable. Programabilidad e intérprete son conceptos que se co-implican. Una consecuencia interesante del concepto de computación entendido a partir de la propiedad de la programabilidad es que los límites que parecieran ser precisos entre las nociones de dato/programa empiezan a diluirse en el análisis del proceso llevado a cabo por el intérprete.

Colocando la interrelación como definitoria de la programabilidad de los sistemas computacionales podemos pensar en la relevancia práctica y técnica de la ciencia de la computación actual:

Sabemos *ahora* que una computadora programable no es ni más ni menos que un dispositivo extremadamente útil para realizar cualquier mecanismo concebible sin cambiar un solo cable y que el desafío central de la ciencia de la computación es entonces conceptual, concretamente, qué mecanismos (abstractos) podemos concebir sin perdernos en las complejidades de nuestra propia creación.<sup>19</sup>

El desafío no es centralmente diseñar mejores mecanismos físicos, sino saber cómo programar mecanismos abstractos para que el mecanismo físico sea verdaderamente un dispositivo útil y versátil en su medio. La potencialidad de la noción de mecanismo a la que nos enfrenta Dijkstra es la de mecanismos programables o universales, es decir, aquellos que para pasar de ser mecanismos de propósito general a mecanismos de propósito particular -y hacerlo *sin la necesidad de cambiar un solo cable*- utilizan la noción de programa intérprete -o máquina universal- ya sea que éste sea concebido como un mecanismo abstracto y formal o como un mecanismo concreto capaz de implementar aquella descripción abstracta.

### 2.1.3. Relación concreto-abstracto

Dijkstra ofrece una segunda conceptualización de lo que es un programa: “es un manipulador de símbolos abstractos, el cual puede ser transformado en uno concreto administrándole una computadora”<sup>20</sup>. Esta definición focaliza en lo que suele calificarse como la naturaleza dual de los programas, caracterizada en el área de la computación a partir de la oposición *hardware/software* y en la historia de la filosofía como la oposición ontológica entre lo abstracto/lo concreto. Dijkstra plantea el problema a partir de evidenciar cómo el procedimiento de manipulación de símbolos que efectúa el programa puede ser pensado en términos abstractos o concretos según sea comprendido como una manipulación formal de símbolos o como un proceso causal en un sistema físico que realiza aquella misma manipulación formal. Es entonces que surge el problema de la implementación: cómo se da ese vínculo entre la computación abstracta y la computación concreta, es decir, cómo un sistema físico realiza -*hace realidad*- una computación cuando ésta se define primordialmente en términos formales o abstractos.

En respuesta a ello, y valiéndonos de la propuesta de Dijkstra, podemos pensar la computación a partir de la noción abstracta de programa como una descripción -abstracta- del código prescriptivo y efectivo que vincula el programa físico y el resultado que éste produce en la máquina, excluyendo así una interpretación dualista, del tipo que ofrece Eden, por ejemplo, en donde el programa abstracto, que él llama el *program-script*,

<sup>19</sup> DIJKSTRA, *op. cit.*, 1986, pág. 49.

<sup>20</sup> DIJKSTRA, *op. cit.*, 1988, pp. 16.

se explica por las leyes formales de la lógica y la matemática, mientras que el programa concreto, que él llama *program-process*, se explica por las leyes causales de la física.<sup>21</sup> En Dijkstra parece haber, en cambio, un vínculo de implementación en donde la computación –y la computadora- resultan de la interacción efectiva entre los aspectos físicos y abstractos. Si bien hay que reconocer que en la propuesta de Dijkstra hay una preeminencia de la dimensión formal, al comprender que la relevancia técnica y práctica de la ciencia de la computación se halla en la implementación de una diversidad de mecanismos en un sistema físico -sin cambiar un solo cable-, la relación de implementación se vuelve central y dar una respuesta que no se fundamente en un dualismo ontológico es crucial (suponiendo que entendimos ya los problemas que el dualismo ontológico supone en relación a la eficacia causal de la realidad abstracta sobre la concreta). Podríamos entender, entonces, la relación de implementación como el ámbito en donde se *negocia* la relación entre los aspectos concretos que impone el mecanismo físico y los aspectos abstractos que impone la programadora humana; en este sentido, es bueno pensar en que programar una computadora es un modo de acción técnica en el mundo, como leímos en Hoare, un modo de diseñar métodos para conseguir que una computadora resuelva problemas o, como ha apuntado Moor, para que produzca otros programas o realice cualquier suerte de manipulación simbólica efectiva en orden a realizar alguna actividad.<sup>22</sup>

En base a las preguntas sobre qué es la computación y de qué trata la ciencia de la computación, Dijkstra propone la superación de los esquemas dualistas tradicionales en pos de una ontología relacional. La superación de la oposición humana/máquina mediante la noción de programación es la base que permite el replanteo y la disolución de la oposición entre lo universal/particular para pensar los objetos técnicos a partir de su propósito y la oposición abstracto/concreto para dar cuenta de la relación de implementación en la computación. Esta apuesta relacional ofrece un marco para abordar algunas de los problemas fundamentales que se plantean hoy en la filosofía de la ciencia de la computación, al mismo tiempo que sugiere cuál puede ser el vínculo de la computación con el proyecto humanista de nuestra época. Al visualizar la relación dinámica como el centro del fenómeno computacional, el enfoque de Dijkstra nos permite pensar que la posibilidad de realizar un nuevo ideal humanista no se halla en la insistencia en determinar un estado estable y definitivo de la técnica -ni de lo humano- sino en la puesta en evidencia de las relaciones inestables, no definidas y en constante evolución.

### 3. Computación e individuación

Si bien Simondon al formular sus tesis no logra visualizar la novedad radical que significa la computación en su tiempo, y en tal sentido emplea ejemplos de técnicas propias de un paradigma anterior y no muestra una clara concepción de la potencialidad de la programación, nuestro propósito es vincular su propuesta ontológica con la concepción de computación que elaboramos con Dijkstra.

La falta de claridad sobre la novedad y potencialidad de la computación en Simondon se debe quizá a que la computadora todavía es interpretada en su tiempo bajo aquella lógica negadora de la novedad que Dijkstra denunciaba: la computadora se concibe, pues, como una potente calculadora o como un autómatas cuyo comportamiento se explica según la finalidad asignada. Podemos observar esta interpretación, por ejemplo, cuando Simondon especifica que el autómatas es un ser técnico que, en cuanto tal, tiene un comportamiento teleológico y una estructura de funcionamiento definida: “el autómatas solo puede adaptarse de una manera convergente a un conjunto de condiciones

<sup>21</sup> EDEN, Amnon, “Three Paradigms of Computer Science”. En: *Minds & Machines*, n. 17, 2007, pp. 135–167.

<sup>22</sup> MOOR, James. “Three Myths of Computer Science”. En: *British Journal for the Philosophy of Science*, v. 29, n.3, 1978, pág. 215.

reduciendo cada vez más la distancia que existe entre su acción y el fin predeterminado<sup>23</sup>, es capaz de adaptarse modificando su relación con el medio pero no de modificarse a sí mismo e inventar nuevas estructuras internas como lo hacen los individuos vivientes,<sup>24</sup> y es por ello que Simondon termina afirmando que la máquina no inventa y no descubre fines en el curso de su acción porque no realiza un verdadero proceso de transducción, entendiendo transducción como la ampliación de un dominio inicial muy restringido que va creciendo en estructura y extensión.<sup>25</sup> Esta descripción puede resultar adecuada para determinado tipo de autómatas mecánicos, como los servomecanismos estudiados por la cibernética por ejemplo, pero no puede generalizarse para el caso de los autómatas programables. Ya Turing en su famoso artículo de 1950 planteaba la posibilidad de diseñar una máquina que aprende y se modifica sí misma: una máquina que transforme sus mismas reglas de operación, su mismo programa, en el proceso de intercambio de información con su contexto.<sup>26</sup> Podemos pensar, pues, que las máquinas niñas de Turing -que expresan quizá uno de los grados de tecnicidad más altos en términos de Simondon<sup>27</sup>- desafían la pretensión de diferenciar entre seres técnicos y otro tipo de entes y así plantean un nuevo problema para el concepto de individuación.

Introducimos anteriormente las dos primeras formas de individuación que propone Simondon: la vital y la física. Sin embargo, nunca queda completamente claro dónde ubicar la individuación técnica, aquella que da origen al objeto técnico, no solo porque Simondon no es completamente explícito sobre la cuestión, sino porque él escribe en un momento histórico en el que el objeto técnico paradigmático cambia completamente y él no es capaz de asimilar en sus aportes lo radical de la irrupción de la computadora.

Ahora bien, poniendo a la individuación técnica en suspenso es necesario mencionar que Simondon planteó al menos dos formas más de individuación: la psíquica y la colectiva, que muchas veces son consideradas como una sola: la psíquico-colectiva. Si el ser vivo, producto de una individuación vital, pudiera estar completamente aplacado y satisfecho por sí mismo, esto es sin conflictos tanto hacia el interior de sus límites somáticos como en su relación con el medio, no habría necesidad de recurrir al psiquismo; pero cuando la vida, se problematiza, no puede tomar de la mera naturaleza, de su entorno más próximo, potenciales que produzcan una nueva individuación sin entrar en un orden de realidad que sobrepasa los límites de lo viviente en términos de individuación vital, y esto supone, entonces, un tipo de individuación diferente: una individuación psíquica. En tal sentido, Simondon sostiene que a todo ser vivo se le plantean conflictos surgidos en la relación entre la percepción y la acción, y que estos conflictos en animales no cognitivos se resuelven mediante la afectividad; sin embargo, en ciertos animales la afectividad en vez de resolver los conflictos los genera, es allí donde aparece necesaria la individuación psíquica. Esta nueva individuación supondría en algunos seres vivos una porción de realidad pre-individual, incapaz de ser individuada por los procesos de individuación vital, a la que Simondon relacionaría con las cuestiones emotivas. Con todo, continuando con su monismo ontogenético, esta nueva individuación no plantea una diferencia radical con la individuación vital, en palabras de Simondon: “no hay una naturaleza, una esencia que permita fundar una antropología; simplemente, un umbral es atravesado: el animal está mejor equipado para vivir que para pensar, y el hombre para pensar que para vivir. Pero tanto el uno como el otro viven y piensan, de manera corriente o excepcional<sup>28</sup>”. Sin embargo, Simondon sostiene, que la realidad psíquica no puede estar cerrada sobre sí misma, no puede resolverse de manera intraindividual, como sí se resolvía la

<sup>23</sup> SIMONDON, *op. cit.*, 2015, pág. 236.

<sup>24</sup> *Ibid.*, pág. 31.

<sup>25</sup> *Ibid.*, pág. 236.

<sup>26</sup> TURING, Alan. “Computing machinery and intelligence”. En: *Mind*, v.. 59, n 236, 1950, pág. 433-460.

<sup>27</sup> “El verdadero perfeccionamiento de las máquinas, aquel del cual se puede decir que eleva el grado de tecnicidad, corresponde no a un acrecentamiento del automatismo, sino, por el contrario, al hecho de que el funcionamiento de una máquina preserve un cierto margen de indeterminación. Es este margen lo que permite a la máquina ser sensible a una información exterior” SIMONDON, *op. cit.*, 2008, pág.33.

<sup>28</sup> SIMONDON, *op. cit.*, 2015, pág. 165.

individuación vital, porque la realidad pre-individual a la que apela “comunica directamente con las otras realidades pre-individuales contenidas en los otros individuos, como las mallas de una red se comunican entre ellas al sobreponerse en la malla siguiente”<sup>29</sup>. Esto supone la génesis de un nuevo orden de magnitud, ya no intra-individual o inter-individual, sino transindividual. Cuando esta red transindividual sufre transformaciones transductivas aparece lo colectivo, estas transformaciones toman la forma de una individuación colectiva, que no es, sin embargo, identificable con lo social puro o lo inter-individual. Mientras que una sociedad de animales no cognitivos, o con grados bajos de cognición, agrupa diferentes heterogeneidades obteniendo ventajas basadas en un funcionamiento como grupo, lo colectivo apela a lo homogéneo de base que poseen ciertos individuos y así a la capacidad de desindividuarse para ser parte de una individuación más vasta. Un grupo de individuos que como grupo no sufre procesos de individuación es meramente algo social, inter-individual, “lo colectivo solo existe verdaderamente si una individuación lo instituye. Es histórico”<sup>30</sup>. Un ejemplo concreto de individuación colectiva que ofrece Simondon es el proceso acerca de cómo el Gran Miedo terminó derivando en la Revolución Burguesa<sup>31</sup>.

Retomando la cuestión sobre el lugar ontológico de la individuación técnica, en un trabajo previo<sup>32</sup> intentamos dar cuenta de que esta individuación podría encontrarse a mitad de camino entre la individuación física y la vital. Lo desarrollado en este trabajo arroja nueva luz sobre aquella hipótesis y nos permite avanzar un poco más en la relación con otras formas de individuación. Centrándonos en la computadora como individuo técnico, y a modo de esbozo de una nueva hipótesis a investigar, sugerimos que ciertas formas de individuación técnica están vinculadas no solo ya con individuaciones físicas o vitales, sino también con un proceso de individuación psíquico-colectivo. Como pudimos observar a partir de Dijkstra, la computadora sin la interacción con la humana es meramente una máquina calculadora, para tener una relevancia práctica y conceptual sobre esta requiere de la manipulación humana de símbolos que es la programación, la cual resulta irreductible a una existencia meramente física. Esto nos permite suponer que existe cierta realidad pre-individual en la computadora que no puede ser resuelta intra-individualmente al igual que sucedía con los seres vivos de orden psíquico y es necesario apelar a otro orden de magnitud análogo a un orden transindividual. Además, esta relación de co-constitución con la humana indicaría que en la computadora esta porción de realidad pre-individual se vincula, como las mallas de una red, con la realidad pre-individual de las humanas, lo cual permite engendrar un proceso análogo a lo colectivo en Simondon. Bajo esta hipótesis, entonces, la computación es interpretada como una forma de individuación colectiva.

A partir de esta hipótesis, podríamos pensar en la computación como aquel fenómeno técnico-cultural en donde el proceso transductivo también es posible en máquinas, no en todas, pero sí en aquellas que gracias a la potencialidad de la programación -que funda el juego de relaciones que señalamos con Dijkstra- desarrollan su funcionamiento no a partir de un individuo ya consumado, sino a partir de la dinámica de la individuación colectiva. Es en este punto en donde consideramos que el encuentro entre la filosofía de Simondon y la concepción relacional de computación desarrollada por Dijkstra permite visualizar los elementos conceptuales necesarios para hacer frente a la alienación de nuestro tiempo expresada bajo las fórmulas tecnofílicas o tecnofóbicas que se ofrecen para pensar la computación. A partir de las primeras, se abonan proyectos tecnocráticos, capitalistas y transhumanistas; a partir de las segundas se sueña con un afuera del mundo computacional, lo cual se expresa tanto en las denuncias de pérdida de lo *propriadamente humano* como en la búsqueda de lo que no pueda ser computado -el Santo Grial del *humanismo fácil* de nuestro tiempo-. Pensar la computación a partir de una

<sup>29</sup> *Ibíd.*, pág. 220.

<sup>30</sup> *Ibíd.*, pág. 245.

<sup>31</sup> *Ibíd.*, pág. 509.

<sup>32</sup> AUTORAS, 2017.

ontología relacional dinámica y contingente permite cuestionar, en cambio, los puntos de interés que han fijado aquellas lecturas reduccionistas de la técnica y, así, abre la posibilidad de ensayar nuevos imaginarios emancipatorios y desalienantes de lo humano:

Así, la primera condición de incorporación de los objetos técnicos a la cultura sería que el hombre no fuera ni inferior ni superior a los objetos técnicos, que pueda abordarlos y aprender a conocerlos manteniendo con ellos una relación de igualdad, de reciprocidad de intercambios: en cierta manera, una relación social.<sup>33</sup>

## Referencias

- BARDIN, Andrea; MENEGALLE, Giovanni. "Introduction to Simondon". *Radical Philosophy*, n. 189, 2015, p. 15-16.
- CANGUILHEM, Georges. *El conocimiento de la vida*. Barcelona: Anagrama, 1976.
- D'ANDREA, Aldana; CAMPOAMOR, Emiliano. "El humano y la técnica: individuación y máquina". En: OSELLA, Mario (comp.). *I Encuentro sobre técnicas del yo/Técnicas de la interioridad*. Río Cuarto: UniRío, 2017, p. 129-39.
- DAVIS, Martin. *The universal computer: The Road from Leibniz to Turing*. New York: W. W. Norton & Company, 2000.
- DIJKSTRA, Edsger. "On a cultural gap". *The Mathematical Intelligencer*, v. 8, n.1, 1986, p. 48-58.
- DIJKSTRA, Edsger. "On the cruelty of really teaching computing science". <http://www.cs.utexas.edu/users/EWD/ewd10xx/EWD1036.PDF> (último acceso 01/07/2020).
- DIJKSTRA, Edsger. The threats to computing science. *ACM 1984 South Central Regional Conference*. Austin: 1984, p., 16-18.
- EDEN, Amnon, "Three Paradigms of Computer Science". *Minds & Machines*, n. 17, 2007, p. 135-167.
- HOARE, Charles (1989). "Computer Science". En HOARE, Charles; JONES, Cliff (eds.). *Essays in Computer Science*. London: Prentice Hall, 1989, p.89-101.
- MOOR, James. "Three Myths of Computer Science". *British Journal for the Philosophy of Science*, v. 29, n.3, 1978, p. 215.
- SIMONDON, Gilbert. *El modo de existencia de los objetos técnicos*. Buenos Aires: Prometeo, 2008.
- SIMONDON, Gilbert. *La individuación a la luz de las nociones de forma e información*. Buenos Aires: Cactus, 2015.
- SIMONDON, Gilbert. *Sobre la técnica*. Buenos Aires: Cactus, 2017.
- TURING, Alan. "Computing machinery and intelligence". *Mind*, v. 59, n 236, 1950, p. 433-460.
- WEGNER, Peter. "Research paradigms in computer science". *Proceedings of the 2nd international Conference on Software Engineering*. San Francisco: 1976, p. 322-30.

---

Doutor em Filosofia (Universidad Nacional de Córdoba)

---

<sup>33</sup> SIMONDON, *op. cit.*, 2008, pág.108.

Professora da Universidad Nacional de Río Cuarto (UNRC); Instituto de Humanidades(IDH)- Universidad Nacional de Córdoba (UNC); Investigadora do CONICET  
E-mail: [aldana.dandrea@gmail.com](mailto:aldana.dandrea@gmail.com)

Doutorando em Filosofia (Universidad Nacional de Córdoba)  
Professor da Universidad Provinciana de Córdoba (UPC); Universidad Nacional de Río Cuarto (UNRC)  
E-mail: [emilianocampoamor@gmail.com](mailto:emilianocampoamor@gmail.com)