

Nuevas tecnologías: humanxs, algoritmos y programación

Aldana D'Andrea

Paula Sibila en su libro *El hombre postorgánico* indica que hay dos mitos fundamentales que “expresan, en la tradición occidental, la mezcla de fascinación y terror causada por las posibilidades de la tecnología y el conocimiento (y también por sus limitaciones)”¹. La autora se refiere al mito de Prometeo y al de Fausto que han iniciado la tradición prometeica y fáustica respectivamente. La tradición prometeica, con una firme confianza en el progreso, ha acentuado el rol liberador del conocimiento científico y el carácter instrumental de la técnica, pero al mismo tiempo ha advertido que existen límites sobre lo que podemos conocer, hacer y crear lxs humanxs; la naturaleza -sobre todo la naturaleza humana- es el límite de la técnica concebida como mera prótesis. La tradición fáustica, sustentada en la misma concepción instrumental, ha fomentado por el contrario el dominio, control y apropiación total de lo natural -tanto en el dominio de lo humano como de lo no-humano-, desde esta tradición no se visualizan límites ni riesgos, solo la posibilidad de progreso; los modelos tecnócratas capitalistas tienen una afinidad clara con esta perspectiva. A partir de la lectura que propone Sibila podemos observar que entre los límites de una tecnofobia de corte naturalista y humanista y una tecnofilia de corte capitalista, el pensamiento sobre la técnica queda atrapado en un dualismo que no parece ayudarnos a pensar la posibilidad de una relación entre la técnica y lxs humanxs más allá del horizonte reproductivista -naturalista o capitalista- que ofrecen los modelos clásicos.

Nuestro propósito es que este escrito contribuya a la constitución de una alternativa que escape a la polaridad prometeica-fáustica, para ello proponemos poner el foco en las tecnologías computacionales actuales y, en particular, en las propiedades que ellas tienen y que las diferencian de antiguas tecnologías; entendemos que esta distinción entre nuevas y antiguas tecnologías –aun no siendo absoluta- aporta dimensiones relevantes a la discusión sobre las potencialidades no reproductivas de la tecnología actual.²

Proponemos un esquema básico: en primer lugar introducir conceptos y resultados fundamentales de la teoría de la computabilidad poniendo énfasis en la noción de algoritmo, la insolubilidad algorítmica y la programación; en segundo lugar analizar cómo estos conceptos y resultados tensionan la noción clásica de máquina, para ello analizamos la noción clásica de máquina de Canguilhem y la ponemos en relación con la noción de máquina que surge de la teoría de la computabilidad; en tercer lugar, y estrechamente vinculado con lo anterior, evaluamos cómo la noción de programación puede desafiar la concepción instrumental de la técnica que ve en el funcionamiento de

¹ Sibila, P., *El hombre postorgánico: cuerpo subjetividad y tecnologías digitales*, FCE, Bs. As., p. 43

² No sostenemos con ello que las tecnologías clásicas no puedan escapar a la polaridad aludida, simplemente nuestro estudio se centra en las tecnologías computacionales y en la posibilidad de encontrar en ellas aspectos novedosos y relevantes para nuestro interés político.

la máquina la realización de la intencionalidad humana; por último, y a modo de cierre, esbozamos cuáles son los cambios que las nuevas tecnologías introducen en su relación con lxs humanxs.

1. Aportes de la teoría de la computabilidad

La teoría de la computabilidad tiene como cometido central establecer el alcance y el límite de lo que puede ser computado mediante un procedimiento efectivo o algoritmo. La computación puede ser abordada desde este punto de vista como una disciplina lógico matemática que define formalmente sus objetos –pueden ser funciones, algoritmos, máquinas de Turing- y prueba teoremas sobre ellos. En este sentido, puede plantearse que la noción de computación y la creación de modelos computacionales surgen en la década de 1930 con el trabajo de lógicos matemáticos como Alan Turing, Alonzo Church, Stephen Kleene, Kurt Gödel y Emil Post en torno a la resolución del *Entscheidungsproblem* (problema de la decisión) de Hilbert³ y, en cuanto tal, son anteriores a la existencia de la ciencia de la computación y a las máquinas computadoras. Se sostiene, además, que el trabajo de estos autores es el fundamento teórico de la actual ciencia de la computación y que sus aportes “presagiaron muchos aspectos de la práctica computacional que son ahora comunes y cuyos antecedentes intelectuales son normalmente desconocidos para los usuarios”⁴. Entre estos antecedentes se incluye la existencia en principio de computadoras digitales de propósito general (o universales), la idea de codificación, el concepto de programa como lista de instrucciones codificadas en un lenguaje formal y, por supuesto, la definición de lo que es computable.

A pesar de que la teoría de la computabilidad es abordada, en la mayoría de los casos, desde una perspectiva absolutamente formal y abstracta, es preciso entender el vínculo estrecho que ésta mantiene con la realidad tecnológica concreta y con la práctica real de lxs científicxs de la computación. Asimismo queremos señalar nuestro horizonte más amplio: los teoremas fundamentales de la computabilidad, en tanto

³ La formulación clásica del *Entscheidungsproblem* es de Hilbert y Ackermann: “El *Entscheidungsproblem* se resuelve si se conoce un procedimiento que permita decidir, usando un número finito de operaciones, sobre la validez, o respectivamente la satisfactibilidad, de una expresión lógica [de primer orden]” (Hilbert, D. y Ackerman, W. 1926; citado por Sieg, W., *Mechanical procedures and mathematical experience*, Carnegie Mellon University, Pittsburgh, 1991, p. 12. Disponible en: <http://repository.cmu.edu/cgi/viewcontent.cgi?article=1243&context=philosophy>. Último acceso 26/10/2019). Sin embargo, la formulación de Behmann del problema de la decisión, es anterior y filosóficamente más sugerente: “Ante una afirmación matemática, el procedimiento requerido debería aportar instrucciones completas para determinar si la sentencia es correcta o falsa por un cálculo determinista después de una cantidad finita de pasos(...) Es esencial para el carácter de este problema que, como método de prueba, solo se permita un cálculo completamente mecánico de acuerdo a instrucciones dadas, sin que sea permitida ninguna actividad del pensamiento en el sentido más limitado. Se podría hablar, si se quiere, de pensamiento mecánico o pensamiento maquinico (quizá un día pueda incluso ser desarrollado por una máquina)” (Behmann, H., “Contributions to the algebra of logic, in particular to the decision problem”, en *The Bulletin of Symbolic Logic*, vol. 21, 2015, n° 22, p. 176).

⁴ Davis, M., Sigal, R. & Weyuker, E., *Computability, complexity, and languages: fundamentals of theoretical computer science*, Academic Press, Florida, 1983, p. xiii.

expresan el alcance y límite de lo que puede ser computado, arrojan nueva luz sobre problemas filosóficos clásicos que tradicionalmente se han formulado en base a una idea apenas intuitiva y poco precisa de lo que puede ser mecanizado o maquinizado.

1.1. Algoritmos y problemas computables

La pregunta fundamental de la teoría de la computabilidad es: ¿existen problemas insolubles por cualquier algoritmo? Preguntas equivalentes son formuladas cuando se sustituye el término *cualquier algoritmo* por *cualquier procedimiento efectivo*, *cualquier programa en cualquier lenguaje de programación* o *cualquier computadora*.

Para dar respuesta a esta pregunta es necesario precisar qué se entiende por algoritmo. En este sentido, uno de los grandes logros de la teoría de la computabilidad es la demostración en aquel contexto fundacional de 1930 de que todas las formalizaciones propuestas de la noción de algoritmo arrojan, en un sentido extensional, una misma teoría y un mismo concepto de computabilidad. Esto es lo que conocemos como la tesis de Church-Turing:

(CT) Un procedimiento es algorítmico si y solo si es computable por una máquina de Turing o λ -definible o recursivo

Tomando por válida (CT) es posible responder de un modo positivo la pregunta planteada inicialmente: existen problemas que no pueden ser resueltos por ningún procedimiento algorítmico -o procedimiento efectivo, o programa o computadora-. Tal resultado, que en sentido estricto es un teorema obtenido al interior de cada uno de los modelos de computabilidad señalados, se conoce como el resultado de insolubilidad algorítmica:

(I) Existen problemas que no pueden ser resueltos mediante un procedimiento algorítmico.

Por *problemas* se entiende en general una clase de preguntas claramente formuladas para las cuales es posible, en principio, dar una respuesta precisa y finita. En este sentido, un problema distinguido de la teoría de la computabilidad es calcular una función.⁵ Por *procedimiento* se entiende un método general, esta noción tiene dos características fundamentales que pueden ser enfatizadas, el carácter teleológico y el normativo; respectivamente, un procedimiento puede ser entendido como un modo de hacer algo *para lograr una meta* –calcular una función–, o como una especificación de ciertos pasos que *se deben seguir* –en el cálculo de una función–. A su vez, un procedimiento que sea *algorítmico* puede ser caracterizado informalmente por las siguientes propiedades:

⁵ Jones, N. (1997). *Computability and complexity: from a programming perspective*. The MIT Press, Massachusetts, p. 10.

- Por su *ejecutabilidad*: el procedimiento consiste en un número finito de instrucciones determinísticas, dadas por especificaciones finitas y no ambiguas que prescriben la ejecución de un número finito de operaciones primitivas.
- Por su *automaticidad*: el desarrollo del procedimiento no requiere intuiciones sobre el dominio, ingenio, invención ni elaboración de conjeturas.
- Por su *uniformidad*: el procedimiento es el mismo para cada posible argumento de la función.
- Por su *fiabilidad*: cuando el procedimiento termina se obtiene el valor correcto de la función para cada argumento.⁶

Tradicionalmente, un procedimiento que goza de tales propiedades se dice algorítmico en el ámbito de la matemática y la lógica y mecánico en un ámbito más general. A partir de la teoría de la computabilidad, entonces, un problema que admite un tratamiento con un procedimiento tal se dice computable, de otra manera es no-computable o incomputable. En la línea de identificación de los diferentes contextos de uso de estos términos equivalentes, es particularmente destacable el uso extendido y generalizado del término *algoritmo* -y la propiedad de *ser algorítmico*- en la actualidad, extendiendo así su área de aplicación desde la lógica matemática hacia las ciencias en general, artes y humanidades. Volveremos a este punto más adelante.

Decimos que un problema es no computable o insoluble por un procedimiento algorítmico cuando el problema, estando bien definido y admitiendo en principio una respuesta, no admite, sin embargo, un tratamiento efectivo exitoso: el método puede concluir y arrojar el resultado buscado o puede no terminar nunca. Esto es el caso del problema de la parada [*halting problem*] inspirado en el trabajo de Turing de 1936 y planteado en términos de programación: dados un programa arbitrario p y un input i no es posible decidir por un método algorítmico si la computación resultante de aplicar p a i para o no. Dicho de otro modo, la función total *halt* que decide la parada de p no es computable.⁷

Un corolario interesante de la demostración de existencia de problemas no computables es que la noción de procedimiento algorítmico abarca funciones parciales, esto es, que el procedimiento aún contando con las propiedades de ejecutabilidad, automaticidad, uniformidad y fiabilidad que caracterizan a la noción informal de algoritmo puede resultar un procedimiento infructuoso para lograr una meta en un tiempo finito determinado. Con este resultado se evidencian dos caras del problema conceptual y epistémico del cálculo algorítmico y que quedan plasmadas en los debates actuales sobre la tesis (CT): por un lado un logro, se evalúa positivamente la precisión que se gana al pasar de la noción intuitiva, históricamente desarrollada, de algoritmo o procedimiento mecánico a la noción formalizada; por otro lado un desafío, se evalúa la

⁶ Piccinini, G., "The physical Church–Turing thesis: modest or bold?", en *The British Journal for the Philosophy of Science*, vol. 62, 2011, n° 4, pp. 737.

⁷ Jones, N., *op. cit.*, p. 17.

posibilidad de dar cuenta de que la noción así formalizada corresponde, en efecto, a la noción intuitiva que se desea definir: ¿la noción formal capta la noción intuitiva? ¿es válida la tesis?

1.1. La perspectiva del análisis de Turing: mecanismos programables

La teoría de la computabilidad se desarrolla desde una perspectiva abstracta o lógico matemática, esto es fundamentalmente patente en el caso de las propuestas fundacionales como el λ -cálculo de Church-Kleene y la teoría de funciones recursivas de Gödel-Herbrand en donde la noción de cálculo algorítmico es identificada con una clase de funciones definidas formalmente. Pero, por otro lado, el caso de la propuesta intensional de Turing⁸ aporta otra perspectiva sobre la noción lógico matemática de cálculo algorítmico, una ligada a la noción de máquina y de cálculo mecánico.

En primer lugar, Turing elabora un análisis de la computabilidad en términos del cálculo mecánico desarrollado por humanxs o máquinas aritméticas abstractas -máquinas de Turing-, con lo que permite un acercamiento intuitivo más directo al enfoque de la computación en términos de la noción de cálculo mecánico y máquinas que desarrollan procedimientos algorítmicos, de lo cual resulta una tesis fundamental: un procedimiento es algorítmico si y solo si es computable por una máquina de Turing.

En segundo lugar, Turing presenta la noción de máquina universal como una máquina capaz de computar cualquier función computable por cualquier otra máquina de Turing. Se reconoce en la actualidad que la noción de máquina universal es la base teórica detrás de las computadoras de propósito general, más precisamente la noción de programa intérprete o compilador es, en esencia, la noción de máquina universal expresada en términos del lenguaje de la ciencia de la computación actual.⁹ Así como podemos definir a la máquina universal como aquella que toma a cualquier otra máquina de Turing M y su input i y devuelve el resultado de aplicar M a i , del mismo modo podemos definir a un intérprete como aquel que toma a cualquier otro programa P y su input i y devuelve el resultado de aplicar P a i .¹⁰ La demostración de la existencia de un elemento universal dentro de la noción de procedimiento algorítmico es un resultado teórico capital de la ciencia de la computación actual, pues indica que una única computadora podría computar todo lo computable, basta con aportarle a ésta el programa correcto en el lenguaje adecuado.

⁸ Turing, A., "On computable numbers, with an application to the Entscheidungsproblem", en Davis, M. (ed), *The Undecidable: Basic Papers on Undecidable Propositions, Unsolvability Problems and Computable Functions*. Raven Press, New York, 1965, pp. 115-154.

⁹ En este sentido, Martin Davis sostiene: "Un programa escrito en un lenguaje de programación moderno es dato para el intérprete o compilador que lo manipula para que las instrucciones puedan ser realmente ejecutadas. De hecho, la máquina universal de Turing puede ser en sí misma considerada como un intérprete" Davis, M., *The universal computer: the Road from Leibniz to Turing*, W. W. Norton & Company, New York, 2000, p. 165.

¹⁰ Jones, N., *op. cit.*, p. 47.

La idea de máquina introducida por Turing para el tratamiento de un problema formal, el *Entscheidungsproblem*, junto al desarrollo de la idea de una máquina universal permiten vincular la noción intuitiva de algoritmo al desarrollo de mecanismos programables y así dar lugar a lo que entendemos actualmente por ciencia de la computación como una disciplina novedosa y singular que surge, no obstante, a partir de planteos tradicionales. En particular, el empleo de la noción de máquina para capturar la idea de cálculo algorítmico desafía la oposición clásica entre humanxs y máquinas que ha permeado la discusión sobre la mecanización de la inferencia;¹¹ por otro lado, el desarrollo de la noción de máquina universal aporta un marco teórico que posibilita el concepto actual de programación y permite comprender a la programabilidad como la capacidad que tienen ciertos mecanismos de arrojar diversos *outputs* o comportarse de modo diverso frente a un mismo *input* y, por ello, de ser mecanismos de propósito general, en abierta oposición a clásica caracterización teleológica e intencionalista de los mecanismos, es decir, a la idea de que un mecanismo responde a una finalidad establecida de antemano por lx humanx que lo diseña.

Finalmente, el desarrollo tecnológico en la construcción de computadoras digitales de propósito general y la demostración de la equivalencia computacional entre los programas codificados en lenguajes de programación de máquina y la noción abstracta de procedimiento algorítmico,¹² sugieren que los resultados de la teoría de la computabilidad no solo admiten ser entendidos en términos lógico matemáticos sino también en términos más amplios. Desde esta perspectiva la tesis central de la teoría de la computabilidad puede ser planteada en términos del desarrollo actual de la ciencia de la computación:

- (I') Existen problemas insolubles para cualquier programa en cualquier lenguaje de programación concebible en cualquier computadora.¹³

La diferencia entre las tesis (I) e (I') no pretende ser extensional, sin embargo la alusión en (I') a la noción de programación ligada a la de computadora –y posiblemente mediada por alguna noción de implementación- habilita a que esta tesis sea pensada no solo en términos abstractos sino también en términos concretos, materiales y tecnológicos, esto es, (I') ya no solo parece referir al ámbito de la lógica matemática sin más –aunque evidentemente se funda en ella-, sino al ámbito más amplio en el que se desarrolla la ciencia de la computación: entre especificaciones formales, programas e implementaciones en sistemas computacionales concretos. Un algoritmo es, pues, una máquina.

Desde la perspectiva introducida por Turing, la piedra angular de la teoría de la computabilidad (CT) y el resultado capital que ella supone (I) se vinculan con el ámbito de la tecnología a través de la noción de máquina que Turing emplea o, lo que es lo

¹¹ Véase D'Andrea, A. "Lo humano y lo mecánico en el análisis de Turing sobre la computación", en Osella, M. (Comp.) *Técnica y subjetividad: las técnicas del yo*, UniRío, Río Cuarto, 2016, pp.63-85.

¹² La equivalencia computacional implica que cualquier problema que puede ser resuelto de manera efectiva por una máquina de Turing puede ser resuelto por la ejecución de un programa codificado en un lenguaje de programación de máquina y su converso y contrapositivo.

¹³ Jones, N., *op. cit.*, p. 3.

mismo para nuestros fines, con la noción de máquina computadora de programa almacenado. Es este vínculo, precisamente, el que nos permite plantear que la tesis de Church-Turing entendida en el marco del desarrollo y construcción de sistemas computacionales concretos posibilita el surgimiento de la categoría de *nuevas tecnologías* en tanto diferentes de las *tecnologías clásicas* a partir de una redefinición de la noción de máquina. Esta diferenciación, a su vez, introduce la posibilidad de analizar cuál es la novedad que introducen las nuevas tecnologías al pensar los procesos de subjetivación en el contexto social y tecnológico actual.

2. La máquina clásica de Canguilhem y la máquina monstruo de la computabilidad

La elucidación del concepto de cálculo mecánico efectuada por la teoría de la computabilidad junto a la sugerencia de la idea de programabilidad y a la realización de sistemas computacionales programables, plantean un escenario en donde se vuelve necesaria una revisión de las nociones clásicas de máquina y mecanismo tanto en dirección a conformar una noción de mecanismo que permita explicar una nueva idea de máquina, como una nueva idea de máquina que, acorde a los logros de la ciencia de la computación, permita evaluar los alcances e implicancias de la bastedad del dominio de lo computable.

En *Máquina y Organismo*, un texto clásico para la discusión sobre el mecanicismo en filosofía, Georges Canguilhem hace una propuesta conceptual sobre las nociones de máquina y mecanismo:

Se puede definir la máquina como una construcción artificial, obra del hombre, cuya función esencial depende de mecanismos. Un mecanismo es una configuración de sólidos en movimiento tal que el movimiento no deroga la configuración. El mecanismo es, entonces, un ensamblaje de partes deformables con restauración periódica de las mismas relaciones entre las partes.¹⁴

Desde el punto de vista propuesto por Canguilhem un mecanismo es un modelo abstracto -una configuración- que explica un fenómeno concreto, ya sea éste natural -Canguilhem considera organismos- o artificial -una máquina-. La particularidad conceptual de la máquina respecto del mecanismo parece radicar en que ella no se reduce a un mecanismo ciego, ya que su funcionamiento no se explica solo a partir de las relaciones de pura causalidad entre sus partes físicas -no se reduce a la mera configuración de sólidos en movimiento-, sino que hay en ella una dimensión teleológica aportada por lx humanx que la construye:

porque si el funcionamiento de una máquina se explica por las relaciones de pura causalidad, la construcción de una máquina no se comprende sin la finalidad, ni sin

¹⁴ Canguilhem, G. (1952) "Machine et organisme", en *La connaissance de la vie*. Librairie Hachette, Paris:, p.126.

el hombre. Una máquina es hecha por el hombre y para el hombre, en vista a algunos fines a obtener, bajo la forma de efectos a producir.¹⁵

Canguilhem reconoce, sin embargo, que esta finalidad práctica no es privativa de la máquina, pues se encuentra también en los mecanismos de los fenómenos naturales, “todo mecanismo debe tener un sentido, dado que un mecanismo no es una dependencia de movimiento fortuito y cualquiera”¹⁶. Afirmando esta dimensión teleológica y práctica de la noción general de mecanismo el autor afirma que la verdadera oposición no se da entre la explicación mecánica de fenómenos naturales y artificiales sino entre los mecanismos cuyo sentido o finalidad es latente y aquellos en el que el sentido o finalidad es patente -los organismos y las máquinas respectivamente-. Pero más allá de este sentido de patencia o latencia de los mecanismos Canguilhem pregunta “¿hay más o menos finalidad en la máquina que en el organismo?” y responde con base en una noción clásica de máquina:

Se diría gustosamente que hay más finalidad en la máquina que en el organismo, dado que la finalidad es rígida y unívoca, univalente (...) A medida que la finalidad está más limitada, el margen de tolerancia está más reducido, la finalidad parece estar más endurecida y pronunciada. En el organismo, por el contrario, se observa –y esto es demasiado conocido para que se insista- una vicariancia de las funciones, una polivalencia de los órganos.¹⁷

La noción de máquina de Canguilhem es clásica en tanto identifica la máquina con un propósito particular en función del cual se diseña y se utiliza su mecanismo, lo cual constituye a la máquina en un artefacto- la delimitación humana de su *para qué* la define- o un instrumento- es un medio para cumplir con un fin-. En base a esto el autor señala que en el mecanismo de la máquina hay más finalidad que en el del organismo, ya que el organismo puede adaptarse a una pluralidad de funciones: “un organismo tiene, pues, más latencia de acción que una máquina. Hay menos finalidad y más potencialidades”¹⁸.

Finalmente, en consonancia con lo anterior, Canguilhem deja ver que su concepción de máquina está en consonancia con su concepción de cálculo:

La máquina, producto de un cálculo, verifica las normas del cálculo, normas racionales de identidad, constancia y previsión, mientras que el organismo viviente actúa según el empirismo. La vida es experiencia, es decir, improvisación, utilización de ocurrencias; es tentativa en todo sentido. De ahí el hecho, a la vez masivo y a menudo subestimado, de que la vida tolera monstruosidades. No hay una máquina monstruo.¹⁹

La noción de máquina de Canguilhem es la de una máquina que verifica el ideal de cálculo decidible, esto es, un procedimiento deductivo con normas finitas, no ambiguas

¹⁵ Ídem, p. 141.

¹⁶ Ídem, p.143.

¹⁷ Ídem, p. 145.

¹⁸ Ídem, p.147.

¹⁹ Ibidem.

y generales que permitan obtener *siempre* el resultado correcto de la operación. Si la noción de máquina verifica este ideal de cálculo, entonces Canguilhem observa con razón que *no hay máquina monstruo*, en el sentido de que la máquina construida con un propósito particular realizará –siempre que no haya fallas– aquella acción que el mecanismo causal, determinista, invariante y predecible le circunscriba hacer.²⁰

Ahora bien, hubo resultados teóricos y evidencias prácticas que han cuestionado tanto las nociones clásicas de máquina y mecanismo, como la noción clásica de cálculo. Por un lado, Babbage en el siglo XIX con el diseño del Motor Analítico y Turing en el siglo XX con la noción de máquina universal mostraron la posibilidad de la existencia de máquinas reflexivas, capaces de transformar su propio curso de acción, programables –máquinas de propósito general– que, en términos de Canguilhem, poseen más latencia de acción que finalidad; por otro lado, los teoremas metamatemáticos en el siglo XX demostraron que la noción de cálculo que serviría para elucidar la noción de cálculo mecánico– y por ello de máquina y mecanismo– posee límites precisos y que el ideal de infalibilidad vinculado a la decidibilidad no caracteriza, como se esperaba, la noción de calculabilidad algorítmica.

En este sentido podemos sostener que la conformación de las nociones de máquina y mecanismo actuales se vincula, en gran medida, con la ruptura del ideal de un cálculo completo, consistente y decidible; la noción de cálculo que permite la elucidación de la noción de calculabilidad algorítmica es ciertamente un cálculo que *tolera monstruosidades*, lo cual ha sido demostrado, en lo fundamental, por los teoremas de incompletud de Gödel y el resultado de insolubilidad algorítmica del *Entscheidungsproblem*. Como además la tesis de Church-Turing admite una interpretación en términos de máquinas, ya sea en la versión fundacional de Turing o en el escenario de las ciencias de la computación actual, los teoremas formales que derivan en nuestra noción de cálculo efectivo permiten afirmar que sí *hay máquina monstruo* en tanto hay una noción de mecanismo universal e indecidible y, por ello mismo, impredecible.

Si se acepta la validez de la tesis de Church-Turing, se debe aceptar también el resultado de indecidibilidad que se sigue del *Entscheidungsproblem*: no existe una máquina de Turing que implemente un proceso finito general para determinar si otra máquina de Turing arrojará o no un resultado dado, de lo cual se sigue la indecidibilidad sobre cualquier proceso mecánico. Con el desarrollo de las ciencias de la computación, la Turing-computabilidad se constituye en el fundamento teórico del alcance y límite de lo que puede hacer una máquina computadora modelada bajo esta tradición. Lo que se aplica a una máquina de Turing se aplica así a programas de computadora. Lo que el resultado de indecidibilidad implica bajo esta lectura es, entonces, que no existe un

²⁰ Ésta es, de hecho, la noción de mecanismo que está a la base de las tesis antimecanicistas, atadas a temores humanistas, que expresan muchxs lógicxs y matemáticxs a fines del siglo XIX y principios del XX ante la posibilidad de la existencia de máquinas que realicen inferencias o cálculos. Charles Peirce, Henri Poincaré, John von Neumann y Godfrey Hardy son ejemplo de ello. Véase D’Andrea, A., “Matemática formal y mecanicismo: entre el sueño y la pesadilla” en *VI Coloquio Nacional de Filosofía, II Coloquio Internacional de Filosofía. Actualidad de la Filosofía: Instituciones, prácticas y resistencias*, UniRío, Río Cuarto, 2014. pp. 320-326.

programa de computadora que describa de antemano el comportamiento de otro programa de computadora para un *input* dado, a excepción que lo simule paso a paso. Podemos explicar deterministamente cada paso del comportamiento de la máquina por su mecanismo determinista sin que ello implique saber cómo se comportará globalmente; se trata, en efecto, de una noción de máquina determinista y, asimismo, impredecible.

Por otro lado, el teorema de Turing de existencia de la máquina universal señala que la noción de procedimiento algorítmico o mecánico que sostiene la tesis de Church-Turing tiene la posibilidad de concebir una máquina que no posee una finalidad rígida y unívoca, ya que una máquina universal simula el comportamiento de cualquier otra máquina de Turing o, de forma análoga, una máquina programable es una máquina de propósito general, un mecanismo que tiene ciertamente menos finalidad que potencialidades. La máquina universal de Turing es un mecanismo –un mecanismo de cálculo–, como tal tiene un sentido o finalidad que explica su funcionamiento, pero no es un sentido patente, no es una finalidad rígida y unívoca sino potencialmente infinita, el fin del mecanismo universal es precisamente simular la finalidad de cualquier otro mecanismo particular.

Notemos a tal respecto cómo esta caracterización de máquinas programables discute la clásica perspectiva humanista y antimecanicista que establece la oposición entre propósito particular/propósito general como fundante de la diferenciación entre máquina y humanx. Uno de los representantes más influyentes al respecto es, sin dudas, René Descartes, quien en *El Discurso del Método* explicita cómo la diferenciación entre el propósito particular de la máquina y el propósito general –o universal– de la razón humana puede ofrecer un método seguro para distinguir seres humanxs *verdaderxs* de máquinas que se les parecen y que pretenden pasar por humanxs:

aun cuando [las máquinas] hicieran varias cosas tan bien y acaso mejor que ninguno de nosotros, no dejarían de fallar en otras, por donde se descubriría que no obran por conocimiento, sino sólo por la disposición de sus órganos, pues mientras que la razón es un instrumento universal, que puede servir en todas las coyunturas, esos órganos, en cambio, necesitan una particular disposición para cada acción particular.²¹

La máquina monstruosa de la teoría de la computabilidad ya no se define a partir del mecanismo teleológico dependiente de la intención humana y ello hace que la disposición particular de sus partes no corresponda a la particularidad de un fin asignado; así mientras la razón humana es universal, la máquina también lo es. Claramente la disolución del criterio de demarcación cartesiano para distinguir humanxs *verdaderxs* de máquinas que lxs imitan, es una de las condiciones de posibilidad del desarrollo de sistemas computacionales inteligentes y así de la rediscusión sobre qué es unx humanx en un contexto que, en general, ya no se admite dualista ni esencialista.

En oposición a la concepción clásica de máquina, la cual concibe un mecanismo cuyas propiedades tienden a la realización física de un ideal de comportamiento regular, uniforme y en donde la consecución de un fin específico es asegurada, la máquina modelada por la tesis de Church-Turing y la máquina universal de Turing se presentan como máquinas que no tienen una única finalidad definida de antemano, además son máquinas que pueden fallar a nuestras expectativas humanas –pueden no parar, por ejemplo-, confrontando así tanto la tradición filosófico epistemológica mecanicista que ha buscado en la máquina una realización de lo infalible, como las intuiciones antimecanicistas y humanistas de quienes veían en la predecibilidad de la máquina el elemento que permitía establecer la diferencia entre el razonamiento humano y el mecánico.

3. La programación más allá de la intencionalidad humana

La relación entre la tecnología y lx humanx es pensada a partir de de la concepción clásica de máquina y de la perspectiva instrumentalista de la técnica en términos dualistas y jerárquicos, así lxs sujetxs humanxs diseñan mecanismos adecuados para la consecución de un fin determinado y la máquina –el objeto, el medio-, si no es una máquina monstruo, acatará dócilmente la intencionalidad humana. La máquina, bajo esta concepción, funciona como un asistente de lxs humanxs, a veces bajo la idea de prótesis, a veces bajo la de soporte materializado de la memoria o extensión de diversas capacidades humanas, siempre algo subalterno respecto de lo que lx humanx es y puede.

La noción de algoritmo especificada a partir del desarrollo de la teoría de la computabilidad y de la ciencia de la computación permite replantear la relación humanx máquina y las concepciones clásicas de máquina y técnica, ya que si bien un primer acercamiento nos haría pensar que programar una máquina es otra forma humana de imponer finalidades - es decir, de diseñar el mecanismo que explicará el funcionamiento de la máquina- una mirada más detenida nos muestra que programar una computadora no significa necesariamente asignarle una finalidad.

Ada Lovelace escribió en 1843 uno de los bálsamos filosóficos más empleados a la hora de aplacar los temores humanistas suscitados por una visión prometeica de la técnica: una máquina solo puede hacer aquello que le mandamos hacer, “no tiene pretensión alguna de originar nada. No puede hacer nada que nosotras no sepamos cómo ordenarle realizar”²². Si bien la idea de Lovelace puede ser cierta para la concepción clásica de máquina de Canguilhem, no necesariamente debe ser cierta para la concepción actual fundada en la noción de algoritmo y programabilidad. Programar una máquina para que realice alguna acción en particular -para que sea el editor de texto que me permite escribir estas líneas, por ejemplo- es ciertamente asignarle una finalidad; pero como nuestras máquinas computadoras corrientes son máquinas de

²² Lovelace, A. ([1843] 1961), “Notes on sketch of the analytical engine invented by Charles Babbage”, en Morrison, P. & Morrison, E. (eds.) *Charles Babbage and his Calculating Engines*. Dover, New York, p. 284.

programa almacenado, podemos programar la *misma* máquina para que cumpla con la finalidad de cualquier otra máquina— que sea un reproductor de videos, un reloj, un aire acondicionado o un automóvil, o todo ello junto-. Preguntar por la finalidad de cualquier máquina programable parece ser una pregunta trunca -¿para qué sirve una computadora?-. Como bien apunta Dijkstra, “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”²³, es decir, sin necesidad de cambiar la configuración de sólidos que hace que la máquina funcione en un determinado sentido y bajo una finalidad pre-asignada. No obstante, aunque la máquina programable no tenga una finalidad rígida asignada de antemano -pues una computadora de programa almacenado podría tener una infinidad numerable de finalidades²⁴-, parece claro y hasta deseable que siga cumpliendo con los mandatos de su programadorx humanx.

Ahora bien, ¿qué sucede si programamos a una máquina para que aprenda? Ya Turing en 1950 señaló la posibilidad conceptual de *máquinas niñas*, es decir máquinas que transformen sus mismas reglas de operación en el proceso de aprendizaje y conforme pasa el tiempo; el aprendizaje consistiría precisamente en la capacidad reflexiva que permite “la confección de su propio programa, o la predicción del efecto de alteraciones en su propia estructura”²⁵; dicho de otro modo, el aprendizaje maquínico consiste en la evolución algorítmica del mismo algoritmo. Qué vaya a hacer esa máquina es una pregunta que no se puede responder *a priori*, pero aún más, pensemos en cómo se complejiza la pregunta si consideremos además que nuestras máquinas computadoras no están aisladas, habitualmente están conectadas a Internet, a la cual podemos considerar como un conjunto realmente numeroso de algoritmos y humanxs que se interrelacionan: que hacen diferentes cosas, que aprenden, que entrenan, que enseñan y que generan una base verdaderamente compleja y numerosa de metadatos -la *Big Data*- que realimentan los mismos procesos de interacción. La máquina niña aprenderá en esa red, mutará con la mutación de las relaciones que se den en esa red; qué vaya a hacer la máquina niña, qué vaya a ser, es una pregunta que no estamos en condiciones de responder. Con el abordaje de Turing sobre el concepto de aprendizaje algorítmico, constatamos que el funcionamiento de la máquina niña es imprevisible o indecible en otro sentido además del señalado por la tesis (I) de la teoría de la computabilidad, en un sentido más cercano a lo que referimos cuando aludimos a la imprevisibilidad de lxs humanxs en su proceso de crecimiento y educación.

Con el concepto de máquina niña Turing está explorando en su célebre artículo de 1950 la posibilidad de máquinas que exhiban un comportamiento inteligente. Más allá de la disputa acerca de si este concepto de aprendizaje supone o no inteligencia, nos

²³ Dijkstra, E. , *On a cultural gap (Draft)*, 1985, p. 1. Disponible en: <https://www.cs.utexas.edu/~EWD/ewd09xx/EWD913.PDF>. (Último acceso 26/10/2019). La cursiva pertenece al original.

²⁴ Esta aseveración se sustenta en la demostración de existencia de un infinito numerable de máquinas de Turing de propósito particular.

²⁵ Turing, A., (1950) “Computing machinery and intelligence”, en *Mind*, vol. 59, 1950, n°236, p. 449.

interesa evidenciar el hecho de que programar una computadora no significa necesariamente darle órdenes para que realice una actividad o un grupo de actividades específicas. Turing vio la posibilidad del aprendizaje autónomo de los algoritmos, hoy sabemos que hay un nivel de programación autónoma en nuestros *mejores* algoritmos. En este sentido, Pablo Rodríguez señala:

Evolucionan, además de obedecer órdenes, de manera de generar correlaciones que no pueden estar dadas de antemano al procesamiento. Los algoritmos basan parte de su potencia en esta imprevisibilidad relativa de su funcionamiento; de lo contrario no les podríamos pedir nada que no sepamos antes. Y esa imprevisibilidad está ligada a su relación con los *Big data*, esto es, más datos de lo que cualquier ser humano en particular podría conocer por sí mismo.²⁶

Los llamados sistemas computacionales inteligentes han dejado de ser propiamente un instrumento o un artefacto, no pueden concebirse bajo la mirada clásica que quisiera convertirlos en un medio para un fin. Ya sea que acentuemos su universalidad (o programabilidad), su reflexividad o su inmensa capacidad de cálculo, las máquinas algorítmicas desafían los argumentos teleológicos o intencionalistas clásicos al mostrar las diferentes facetas en que una máquina se puede decir indecible, imprevisible o autónoma respecto de alguna voluntad en particular. *Esto*, bajo la perspectiva clásica es una máquina monstruo, pero no necesariamente tiene que ser la criatura doliente y acechante de Frankenstein o la súperinteligente y conspiradora HAL de Kubrick; en las monstruosidades también se hallan las posibilidades de un futuro no reproductivo.

4. Nuevas tecnologías, nuevxs humanxs

¿Hay novedad en lo que llamamos nuevas tecnologías? Más allá del debate sobre la existencia de la novedad radical o absoluta, nos interesa remarcar algunos puntos que consideramos importante no desestimar a la hora de pensar nuestro contexto tecnológico.

En primer lugar, la diferencia específica introducida por la incorporación de los algoritmos al mundo técnico no radica solamente en el incremento de la velocidad y cantidad de datos procesados por la máquina, sino en las características del cálculo que verifica la máquina: las nuevas máquinas son programables e indecibles, tienen la capacidad de cambiar reflexivamente su propio contenido y así aprender. En segundo lugar, la equiparación epistémica entre el cálculo humano y el cálculo mecánico (bajo una nueva concepción de cálculo y de mecanismo- y quizá de humanx-), habilita cada vez más a la delegación de procesos mentales, sociales, culturales, económicos y políticos en sistemas computacionales *inteligentes*, dando cada vez mayor grado de agencia a los dispositivos técnicos sobre los procesos sociales en general y de

²⁶ Rodríguez, P., “Gubernamentalidad algorítmica: sobre las formas de subjetivación en la sociedad de los metadatos” en *Revista Barda*, año 3, 2017, n°5, p. 29.

subjetivación en particular. En tercer lugar, la Internet es un entorno que favorece tanto la interacción y retroalimentación entre algoritmos como nuevos acoplamientos algorítmicos y no algorítmicos entre humanxs y no humanxs. En cuarto y último lugar, hay un mayor uso de tecnologías integradas y miniaturizadas e invisibilizadas, lo cual conduce a un rápido crecimiento de la cuantificación que se está produciendo de la sociedad: “El software está cuantificando y midiendo, cada vez más, nuestra vida cotidiana social y personal (...) Esto torna nuestras vidas en métricas cuantificables que son ahora visibles y susceptibles de computación y procesamiento”²⁷.

Queremos enfatizar este último punto, porque los algoritmos, a diferencia de las máquinas y artefactos tradicionales, son a menudo máquinas miniaturizadas, invisibilizadas, o incluso directamente inmateriales en su carácter de *software*, y esta parece ser una de las condiciones clave para que su presencia sea cada vez más ubicua y su eficacia cada vez mayor. En este sentido David Berry, en un llamado a la constitución de una filosofía del *software*, nos advierte que:

A medida que el software estructura cada vez más el mundo contemporáneo, curiosamente también se retira, y se vuelve cada vez más difícil para nosotros centrarnos en él, ya que está incrustado, oculto, descuidado o simplemente olvidado. El desafío es devolver la visibilidad al software para que podamos prestar atención tanto a lo que es (ontología), de dónde proviene (a través de la arqueología y genealogía de los medios) como a lo que está haciendo (a través de una forma de mecanología).²⁸

El carácter de no-material, digital o virtual asignado a lo que llamamos *software* oculta su existencia y, con ello, su performatividad. Berry nos recuerda que la condición para pensar cualquier fenómeno es advertirlo y comprenderlo. Precisamente en este punto se halla el desafío al que nos enfrenta la algoritmización creciente de la realidad, porque por un lado los algoritmos no se ven, por otro lado, no se terminan de comprender. Lo que no se ve ni se comprende adquiere fácilmente o bien el estatus de no existente (ya sea por un caso de falacia de apelación a la ignorancia o simple ceguera) o bien el estatus de divinidad. No creemos que el negacionismo ni la religión sean una buena respuesta para este –y ningún- problema.

Hay, ciertamente, nuevos fenómenos tecnológicos que deben ser visibilizados y pensados críticamente y aún así, es claro que podemos discutir el grado de novedad de las nuevas tecnologías. Asimismo creemos que debemos reconocer que las tecnologías computacionales han introducido cambios lo suficientemente relevantes como para señalar que entre antiguas y nuevas tecnologías no hay solo un proceso lineal de avance tecnológico, sino que hay también saltos y quiebres conceptuales, epistémicos y técnicos que podrían conducirnos a una revisión de los esquemas con los que la filosofía moderna ha pensado la política, la sociedad, los procesos de subjetivación y, en definitiva, qué significa ser humanxs. En efecto, podemos observar que en la actualidad

²⁷ Berry, D., *The philosophy of software: code and mediation in the digital age*, Palgrave Macmillan, London, 2011, p. 2.

²⁸ Idem, p. 4.

hay diversas alusiones a la *cultura algorítmica*²⁹, al *individuo algorítmicamente asistido*³⁰ y a la *gubernamentalidad algorítmica*³¹, entre otros importantes aspectos a partir de los cuales se señala que lo algorítmico es constitutivo de dimensiones tradicionalmente consideradas como privativas de los humanxs. Si la cultura, la subjetividad y la política no son, como otrora se esperaba, los terrenos en donde lo incomputable o no mecanizable encuentra su sede inamovible, si lo algorítmico se filtra y media la constitución de lo real y el pensamiento sobre lo real, entonces la búsqueda de un afuera del mundo algorítmico parece ser en el mejor de los casos una insistencia humanista ingenua, cuando no una insistencia humanista reproductivista de lo que histórica y políticamente ha sido nombrado como *lo natural*.

Lo que sea lo no computable, lo que exceda al algoritmo, lo que no se pueda decir mecánico, parece ser el último reducto de naturalistas prometeicos. En el mismo sentido, la fascinación fáustica por la infinitud del dominio de lo computable y las promesas de dominio y control con diversos fines, parecen responder a intereses que lejos de fomentar la exploración de posibilidades emancipatorias que ofrece el mundo algorítmico, lo presentan como la nueva religión. En este sentido, la discusión política sobre las nuevas tecnologías sigue siendo tan necesaria hoy como en épocas de antiguas tecnologías, el desafío es no caer en reduccionismos propios de un pensamiento binario y así obturar las posibilidades ofrecidas.

Creemos que tenemos la responsabilidad posicionarnos en una abierta oposición a aquellas dos perspectivas clásicas y opositivas que han delineado el pensamiento sobre lo técnico, pues creemos que a partir de la comprensión de las nuevas tecnologías computacionales -de sus alcances, límites y complejidades- y de una insistencia en la interacción entre humanxs y técnica, y no en la oposición o relación de dominio, es posible ir más allá del terror inmovilizante y de la fascinación ingenua que provoca lo monstruoso, y construir un proyecto tecno-político donde la tecnología pueda concebirse, desarrollarse y emplearse en un sentido estratégico, emancipatorio y no reproductivo -monstruoso, quizá-.

²⁹ Striphas, T., "Algorithmic culture", en *European Journal of Cultural Studies*, vol. 18, 2015, pp. 395-412. Disponible en: <https://journals.sagepub.com/doi/10.1177/1367549415577392> (Último acceso: 26/10/2019).

³⁰ Sadin, É., *La humanidad aumentada: la administración digital del mundo*, Caja negra, Bs. As., 2017.

³¹ Rouvroy, A. y Berns, T. (2016). "Gubernamentalidad algorítmica y perspectivas de emancipación. ¿La disparidad como condición de individuación a través de la relación?", en *Ecuador Debate*, n° 104, 2018, pp. 123-146. Y Rodríguez, P., *op. cit.*