

No hay hardware sin software: Crítica del dualismo digital

Javier Blanco ·
Agustín Berti **

Resumen

En los últimos treinta años se constata una creciente ubicuidad de computadoras en todas las esferas de la vida humana. Esta expansión obedece a las posibilidades programables de esos dispositivos y a su capacidad para suplir el funcionamiento de objetos técnicos previos. Desde el desarrollo de la noción de máquina universal, la noción de programa instaura una división entre hardware y software que se ha constituido como un neo-cartesianismo. En este trabajo reseñamos los conceptos de computación y programa, para luego discutir la ontología relacional de los objetos digitales y la necesidad de su medio asociado recuperando la lectura de la tecnicidad de Gilbert Simondon. En ese recorrido prestamos particular atención a la aparente paradoja de la materialidad de lo digital y al problema de la identidad y la copia de lo codificado. A partir de allí discutimos la noción de intérprete para repensar la imbricación e indistinción entre hardware y software a determinados niveles de abstracción para cuestionar las miradas dualistas.

Palabras clave

Hardware; Software; Objeto Digital; Dualismo; Intérprete.

Abstract

Over the last thirty years, there has been a growing ubiquity of computers in all spheres of human life. Such expansion is due to the programmability of such devices and their ability to replace the workings of previous technical objects. Since the developing of the

· FaMAF, UNC / Maestría en Tecnología, Políticas y Culturas, FCS-FA-FFyH, UNC.
javier.o.blanco@gmail.com

** CONICET / FA, UNC / Maestría en Tecnología, Políticas y Culturas, FCS-FA-FFyH, UNC.
agustin.beriti@gmail.com

notion of universal machine, the idea of program establishes a division between hardware and software that has become a form of neo Cartesianism. In this paper we present a brief account of the concepts of computing and program, in order to discuss the relational ontology of digital objects and their need of an associated milieu, reconsidering Gilbert Simondon's account of technicity. Following this conceptual path we focus on the apparent paradox of digital materiality and the associated problems of identity and copy of codified entities. From that standpoint we will discuss the concept of interpreter to rethink the imbrications and indistinctness between hardware and software at certain levels of abstraction to put into question dualist perspectives.

Key words

Hardware; Software; Digital Object; Dualism; Interpreter.

Puppet Master. It can also be argued that DNA is nothing more than a program designed to preserve itself. Life has become more complex in the overwhelming sea of information. And life, when organized into species, relies upon genes to be its memory system. So, man is an individual only because of his intangible memory... and memory cannot be defined, but it defines mankind. The advent of computers, and the subsequent accumulation of incalculable data has given rise to a new system of memory and thought parallel to your own. Humanity has underestimated the consequences of computerization.

Mamoru Oshii, *Ghost in the Shell*.

1. *Fantasmas*

Un fantasma recorre las redes: el software. Todas las fuerzas de la vieja Academia se han unido en Santa Alianza contra este fantasma: los fisicalistas y los semánticos, los humanistas y algunos mecanicistas, los libertarios franceses¹ y la teoría alemana de medios². ¿Qué hay de disruptivo en el software, qué novedad anida en esta nueva forma de existencia que lo vuelve a veces el indicio de una amenaza innombrable y otras una forma ilusoria que debe ser exorcizada del mundo material?

¹ TIQQUN. *La hipótesis cibernética*. Buenos Aires, Hekht, 2016.

² KITTLER, Friederich. There is no software. In: *CTheory*, Archives, 1995.

Parecería que la aparición del software hace retornar ciertas concepciones cartesianas, en particular una nueva versión de dualismo, con otras credenciales y justificaciones y renovadas posibilidades de ser sostenido: los programas serían entidades abstractas que admiten realizaciones múltiples pero cuyo modo de existencia estaría en el ámbito de lo mental. La distinción software/hardware replicaría la división cartesiana con la ventaja de que la conexión entre ambas sería explicable a partir de la adecuación a protocolos específicos.

La consideración del software como perteneciente al mundo mental se asocia al proyecto de la inteligencia artificial simbólica, a la posibilidad de imitar la mente humana por medio de un programa particular (o de una familia de programas). Esta concepción fue recibida con cierta resistencia por parte de los filósofos, siendo uno de los argumentos en contra la incompletitud de los sistemas formales y la consiguiente necesidad de que algunas capacidades mentales escapen a las formalizaciones. Más allá de cuán bien pueda sonar esto último, los argumentos a favor, supuestamente basados en las consecuencias del teorema de Gödel y los resultados de Turing, son débiles aunque continuamente vuelven a ser esgrimidos pese a haber sido refutados varias veces³.

Sin entrar en las variantes gödelianas, aparece también otra versión del dilema cartesiano, formulada por Alan Turing, pero que condensa una mirada común en la actualidad acerca del tipo de funcionamiento de una computadora.

En su artículo de 1950, Turing habla de la objeción de Lady Lovelace acerca de la posibilidad de que las máquinas puedan pensar, cuya formulación se volvió un lugar común: una computadora solo puede hacer aquello que sabemos cómo ordenarle que haga. El programa que corre en la computadora es nuestra manera de ordenarle qué hacer. En ese sentido, el programa aparece como una prescripción determinista del comportamiento a seguir. Considerada superficialmente, esta objeción implicaría que una computadora -o un robot- nunca podría tener comportamientos creativos, lo que eso signifique, que está inexorablemente atada a seguir una rutina preestablecida y precisa⁴. Esto se condice con pensar a una computadora como un artefacto técnico, definido para un propósito en las versiones intencionalistas. Sin embargo nuestras experiencias con las computadoras no es esa, la versatilidad enorme que poseen las excluye de esa versión

³ WEBB, Judson. *Mechanism, Mentalism, and Metamathematics: An Essay on Finitism*. Dordrecht: D. Reidel Publishing Company, 1980.

⁴ En la serie de HBO *Westworld* esto parece una característica distintiva de las inteligencias artificiales, sin embargo eso mismo parece ponerse en cuestión con el desarrollo de la historia.

teleológica⁵. Esta versatilidad es una consecuencia directa de que pueden pensarse a las computadoras modernas como una encarnación particular de la llamada máquina universal de Turing, es decir, una máquina que a partir de una codificación “externa” (un programa pero en la forma de un dato para dicha máquina) puede comportarse como cualquier otra máquina.

2. *Ontologías digitales*

La definición de qué son estas máquinas es una discusión sin resolver. Su ontología presenta algunos rasgos que los diferencian de otros modos de existencia. Definidos a partir de una similitud de funciones, determinados programas y archivos pueden ser identificados como una prolongación de objetos pre-digitales. Así el procesador de texto se percibe como la continuidad de la máquina de escribir, el editor de video como la de la isla de edición, un programa de diseño gráfico como la del tablero del diseñador, una planilla de cálculo como la del libro contable y así.

Como ya ha señalado Manovich⁶, la definición a partir del modo en que los programas se comportan por analogía con objetos de otros órdenes implica subestimarlos de manera grosera. Como discutiremos más adelante, los programas realizan muchas más funciones (y a escalas y velocidades inconmensurablemente superiores) que las de los objetos en los que se inspiran y de algún modo simulan. La tesis de Manovich es que el software es la realización de la máquina universal de Turing, en la medida en que puede copiar, usando programas, todas las máquinas preexistentes. Solo que al hacerlo en realidad las transforma en algo completamente nuevo, con más funciones (y en este sentido está haciendo una lectura metafórica más que lógica de la máquina universal). Si bien él no lo piensa en términos simondonianos, sostendremos que la ampliación radical a la que hace referencia se debe a su creciente indeterminación como máquinas.⁷ Con

⁵ It is possible, and even tempting, to view a program as an abstract mechanism, as a device of some sort. To do so, however, is highly dangerous: the analogy is too shallow because a program is, as a mechanism, totally different from all the familiar analogue devices we grew up with. Like all digitally encoded information, it has unavoidably the uncomfortable property that the smallest possible perturbations —i.e. changes of a single bit— can have the most drastic consequences.” Dijkstra, *On the cruelty of really teaching computing science*. Pag. 9.

⁶ MANOVICH, Lev. 2013. *Software takes Command*. Cambridge, Mass.: MIT Press.

⁷ Para una discusión sobre la indeterminación en el software véanse los trabajos de Mills en relación al software de Twitter y también de Hui en relación a los distintos lenguajes web de HTML a XML citados a continuación.

todo, la lectura de Manovich privilegia uno de los dos factores del dualismo antes mencionado: el software.

Por contraposición, Matthew Kirschenbaum⁸, recupera un aspecto fundamental de las máquinas de Turing realmente existentes, y que no es objeto de atención filosófica: las manifestaciones y funciones de los diferentes sistemas de almacenamiento. Se propone contemplar la especificidad de la existencia de lo codificado atendiendo a sus modos de existencia física. Resumiendo lo expuesto en *Mechanisms*, si bien en tanto código los programas son abstracciones, éstas siempre están almacenadas en algún soporte de memoria, es decir, tienen alguna forma material de existencia. Los dispositivos de almacenamiento son, entonces, un aspecto ineludible de la existencia de lo digital. Esta constatación no sólo tiene implicancias en las discusiones sobre uno de los pilares de la cultura digital, a saber el mito de la desmaterialización, sino que también abre una perspectiva para la ontología del software a partir de la identificación de una nueva dualidad que supera las interpretaciones neo-cartesianas de lo digital. Una misma codificación puede ser múltiplemente realizable, es decir, no se manifestará materialmente del mismo modo en un disco rígido (como incisiones láser) que en una memoria flash (como una secuencia de impulsos eléctricos), o en un ya obsoleto floppy disk (como polaridades magnéticas) y sin embargo en términos formales seguirá siendo la misma entidad.

Para describir este estado de cosas, Kirschenbaum propone al nivel de los programas una distinción entre materialidad formal y materialidad forense. Su distinción parte del modelo tripartito de Thibodeau, que considera primero, un objeto físico (la inscripción del código en un medio de almacenamiento dado); en segundo lugar, un objeto lógico (como datos reconocidos e interpretados como tales por un software particular, volveremos luego sobre este punto en particular); y en tercer lugar, un objeto conceptual (como aquellos con los que lidiamos “en el mundo real”, es decir el *output* del proceso como se manifiesta en pantalla, parlantes, impresión, etc.). Para Thibodeau entonces, un objeto digital será un objeto de información, es decir una codificación particular, cuya ontología requiera de estas tres condiciones.⁹

⁸ KIRSCHENBAUM, Matthew G. *Mechanisms: New Media and the Forensic Imagination*. Cambridge, Mass: MIT Press, 2012.

⁹ *Ibid.*, 3.

La tercera condición se corresponde con la noción que Manovich asume de modo metafórico para referirse al software de producción cultural como máquina universal. Desde su premisa, las relaciones que establecemos con esas entidades implica asumirlas como continuidades o extensiones de los objetos pre-digitales. Este puede ser un camino engañoso que puede derivar en un sublime tecnológico por ser incapaz de aprehender lo que suceda más allá de las manifestaciones perceptibles a los sentidos humanos. La tercera condición, es necesariamente antropocéntrica, tomando la percepción humana como la única relevante. Por ello Kirschenbaum, piensa la ontología de lo digital a partir de dos formas de materialidad que se corresponden con la primera y la segunda condición de Thibodeau, la materialidad forense y la formal, respectivamente. Los conceptos enfatizan otros aspectos que no aparecen caracterizados en el modelo tripartito.

La materialidad forense se basa en el principio de individualización que permite cada inscripción de un código en un dispositivo de almacenamiento dado, a partir de la idea de que no puede haber dos cosas exactamente idénticas en el mundo físico (en principio desde el momento en que no pueden ocupar el mismo espacio al mismo tiempo). En ese sentido, la materialidad forense es autográfica, del mismo modo que lo es una obra de arte. En los términos del célebre ensayo de Walter Benjamin sobre la reproductibilidad técnica, podríamos decir que este es un rasgo “aurático” en el sentido de lo único e irreplicable, la coincidencia espacial y temporal, el aquí y el ahora que individualizan a la entidad.¹⁰ Entonces, toda copia *traiciona* el principio autográfico de la materialidad forense.

Por el contrario, la materialidad formal es alográfica, su ontología se *consume* en la copia. La materialidad formal es abstracta, no aurática, donde todas las copias gozan de un mismo estatuto frente a la materialidad forense, concreta, individualizable. Esto es así porque en la reproducción alográfica no se requiere una fidelidad autográfica a un continuo de las variaciones posibles que singulariza a una entidad sino la ocurrencia de una serie discreta de variables dentro de un rango dado, que permitan identificar dos ocurrencias de una serie delimitada de signos como idénticas. Por este rasgo una misma materialidad formal puede tener múltiples materialidades forenses. Un ejemplo es la copia de un archivo de imagen .jpg de un diskette (un dispositivo magnético) a un disco rígido (uno mecánico) y luego de allí a una memoria flash (uno eléctrico). En términos

¹⁰ BENJAMIN, Walter. La obra de arte en la época de su reproductibilidad técnica. In: VERA BARROS, Tomás (Comp.) *Estética de la imagen*. Buenos Aires: La Marca, 2015, pp. 30-32.

alográficos, es un único objeto. En términos autográficos, son tres diferentes (una serie de polaridades magnéticas, una serie de inscripciones láser y una serie de variaciones eléctricas). Esta división, de algún modo, amplía el dualismo cuerpo-alma a la posibilidad de múltiples cuerpos (o encarnaciones) de una misma alma.

Sin embargo, como el propio Kirschenbaum constata en su estudio de casos¹¹, hay rasgos específicos de determinados dispositivos de inscripción, es decir de su materialidad forense, que persisten en la materialidad formal representada alográficamente en otros dispositivos diferente, aunque no comporten significado lógico formal. Los archivistas digitales son los primeros en registrar esta paradoja en los modos de existencia de los objetos digitales:

What is wild about digital objects is that there are extensive forensic, or artifactual, traces of the media they were stored on encoded on inside the formal digital object like a disk image. That is, the formal object of a disk image records some of the forensic, the artifactual, the thingyness of the original disk media that object was stored on. The forensic disk image is allographic but retains autographic traces of the artifact.¹²

En el nivel los objetos lógicos de Thibodeau, Yuk Hui propone una perspectiva novedosa para abordar las nuevas ontologías: considerar a los objetos digitales como extensiones de la idea clásica de objetos naturales, o bien como objetos técnicos particulares tomando como punto de partida la interpretación simondoniana propuesta en *El modo de existencia de los objetos técnicos*.¹³ Para esto es indispensable comprender la naturaleza de los medios asociados en los cuales existen tales objetos existen y función de los que se organizan.

La necesidad de adaptación al medio [*milieu*] es la causa de lo que Simondon denomina proceso de concretización, la tendencia a una progresiva incorporación en su esquema de funcionamiento de los obstáculos que le presenta el medio y que lo hacen evolucionar.¹⁴ A determinado nivel de complejidad, como en el caso de los objetos

¹¹ KIRSCHENBAUM, *Op. Cit.* pp. 111-157.

¹² OWENS, Trevor. The is of the Digital Object and the is of the Artifact. In: *The Signal. Digital Preservation*. October 25, 2012 <<http://blogs.loc.gov/digitalpreservation/2012/10/the-is-of-the-digital-object-and-the-is-of-the-artifact/>>

¹³ HUI, Yuk. What is a Digital Object? In: *Metaphilosophy*. vol. 43, n. 4, 2012, pp. 380-395. New Haven: Southern Connecticut State University/Wiley & Sons.

¹⁴ SIMONDON, Gilbert. 2008. *El modo de existencia de los objetos técnicos*. Buenos Aires: Prometeo.

técnicos industriales, el medio comienza a ser tan artificial como el propio objeto. Y asumiendo la continuidad propuesta por Hui, la relación indisociable de objeto técnico y medio asociado, al que denomina medio digital, es constitutiva de esta nueva fase.

En términos de la materialidad forense podemos ver a un objeto digital dado como una secuencia de bits, o incluso, más concretamente, como su realización física en términos de voltajes, de una distribución de orientaciones magnéticas en una superficie adecuada o de agujeros en un papel o un disco. Pero no parece posible pensar la identidad de un objeto digital sólo en términos de su realización física. Como señalamos antes, cuando copiamos un archivo de un CD-ROM a un disco de estado sólido [SSD], las respectivas realizaciones físicas pierden cualquier semejanza posible. Sin embargo a nivel de la percepción humana, como objetos conceptuales siguiendo a Thibodeau, las estamos identificando como idénticas (el mismo uso de la palabra “copiar” da cuenta de esto).

En términos de materialidad formal, considerar al objeto digital sólo como una secuencia de bits es también relativamente problemático, ya que la misma secuencia puede ser interpretada de muy diferentes maneras dependiendo del contexto digital en el cual se inserte. Y por ello la propuesta de Kirschenbaum y el modelo tripartito de Thibodeau resultan insuficientes. Puede ser un programa en código fuente, en código de máquina, una aplicación como un procesador de texto, un archivo de texto. Ahora, dado un contexto específico, no cualquier secuencia de bits es allí un objeto, y es el mismo contexto el que determina las condiciones de posibilidad de que los objetos digitales existan como tales. Un error de codificación convierte una canción codificada en formato .flac en mero ruido. O en un nivel mayor de abstracción, un error puede convertir un programa que interpreta las codificaciones y reproduce canciones en un programa que nos devuelve repetidamente el mismo mensaje de error.

Hui intenta caracterizar la variedad de objetos digitales existentes en el medio asociado de la red. Este medio se ha ido agrandando de manera exponencial, abriendo nuevas posibilidades (más y diferentes objetos digitales) y desafíos (mantener la interoperatividad y la coherencia de los datos). El problema aquí es la imposibilidad de establecer una diferencia esencial entre medio y objeto, por ejemplo entre un archivo de texto y un procesador de textos. Esta in-diferencia es constitutiva de la computación como disciplina y está demostrada en el teorema de Turing de existencia de una máquina universal que comentamos al comienzo. Aquí hay que ser cuidadosos al considerar el modo de existencia de los objetos digitales, ya que las distinciones están solo dadas por

el marco. Dada la velocidad del cambio tecnológico, los marcos pueden cambiar, y de hecho lo hacen todo el tiempo, y en un instante un programa puede dejar de serlo un programa para convertirse en una indescifrable secuencia de bits. De ser un programa pasa a ser un problema: ¿cómo darle sentido a ese objeto o cómo hacer que una materialidad forense dada “devenga” objeto, es decir, materialidad formal con la cual podamos relacionarnos. Un ejemplo sintomático de esto son los archivos encriptados, tan necesarios para evitar el panóptico tecnológico. Si se dispone de la clave de acceso, este objeto puede ser cualquier objeto; si no se dispone de la clave (y el protocolo criptográfico es lo suficientemente seguro) solo tenemos ruido.

Si bien es interesante establecer diferencias entre objetos digitales de acuerdo a su posibilidad de manipulación, o en otros términos a cuan indeterminado sea, no hay que perder de vista que tal posibilidad no es una propiedad intrínseca del objeto digital, sino una propiedad relacional establecida a partir de un marco de referencia (un compilador o intérprete, un sistema operativo, una codificación de imagen o sonido).

El aspecto que diferencia radicalmente a los objetos digitales de otros objetos técnicos, es que los elementos constitutivos del medio digital suelen ser ellos mismos objetos digitales. Si bien, siempre es necesaria una inscripción física, el anclaje de lo digital en él puede diferirse tanto como se quiera, es decir puede haber un medio digital que simule ser un hardware particular. Esto tiene algunas consecuencias que pueden verse como propiedades distintivas de los objetos digitales. Por un lado, los objetos digitales admiten múltiples realizaciones, compartiendo esto con los objetos técnicos, incluso de manera más clara. Pero la identidad misma de los objetos digitales sólo puede considerarse de manera relacional, por lo que el medio asociado resulta indisoluble de su modo de existencia. Una cadena de bits puede realizarse físicamente de las maneras más diversas, pero su significado como objeto digital depende de la interpretación que hagan de sí los programas de su entorno digital. Dichos programas también son, a su vez, objetos digitales, y por lo tanto también admiten múltiples realizaciones, y también son identificados en su propio entorno o medio asociado (que puede incluir el entorno que ellos conforman, pero no necesariamente se identificará con él). Esta cadena de medios asociados, de meta-medios, meta-meta-medios, meta-meta-meta-medios y así cuantas veces se requiera puede extenderse, no habiendo *a priori* límites conceptuales para dichas extensiones. Pero la interpretación del código es un aspecto fundamental para su existencia. Por su materialidad formal: cuando un objeto digital deja de ser interpretable,

deja de existir en tanto objeto. Y para que pueda ser interpretable, es necesario un medio asociado digital, como advierte Hui.

En el primer volumen de *La técnica y el tiempo. El pecado de Epimeteo*, Stiegler advierte que además de la obviedad de que no hay técnica sin hombre tampoco hay hombre sin técnica. La técnica es el rasgo elemental de lo humano que suple aquello de lo que carece y que hace del ser humano un ser protésico que se co-constituye con su medio (el que es, en gran medida, también artificial). La técnica, a su vez, sólo puede ser tal en la posibilidad de anticipación que separa al hombre (que es consciente de su mortalidad y por ello procura diferirla lo más posible) de los demás seres orgánicos (que son sólo perennes), pero también de lo inerte y de la materia inorgánica organizada que él crea (las herramientas e inscripciones). Es este último modo de existencia, el de la materia inorgánica organizada, el que permite postular la existencia de exteriorizaciones, que son las que a su vez habilitan a pensar la existencia de un interior humano. Sólo con la aparición de memoria fuera del cuerpo, nace la conciencia del cuerpo como algo separado. Pero, además, este depósito fuera del cuerpo es acumulable, fundando así el tiempo al poder tomar conciencia no sólo de la diferencia espacial del sujeto con el medio, si no también del diferimiento temporal. Stiegler presenta un giro novedoso, la técnica, aspecto co-constitutivo de lo humano, es tanto el lenguaje como el utillaje [*ouillage*], y ninguno de los dos puede existir sin los hombres, que son el medio asociado básico de la técnica. Simon Mills retomará de otro modo esta intuición para pensar la evolución de las redes sociales *qua* objetos técnicos, como discutiremos más adelante.

Stiegler localiza el problema fundamental de la técnica en la co-constitución originada en carencias biológicas de la especie (de garras, de escamas, de pelo, de alas, de branquias, por ejemplo) que instalan al hombre a la indeterminación y evitan su ultradaptación a un medio particular. La procura protética para suplir tal carencia es, por ello, el rasgo distintivo de lo humano que lo diferencia de las demás especies. Al recurrir a la prótesis necesariamente se descubre el exterior del cuerpo: la flecha y la palabra se lanzan fuera del propio cuerpo y, al hacerlo, permiten al hombre tomar conciencia de su interior. Pero también proyectar fuera de sí (el grito, el proyectil) requiere anticipar el efecto y de este modo, como señalamos antes, las prótesis fundan el tiempo. Y lo que es más importante aún, tanto el lenguaje como el utillaje obedecen a una gramática que los dota de intelegibilidad.

La idea de una gramática de los objetos no es nueva. Hay una serie de ordenamientos y procedimientos que si no se siguen hacen que los objetos devengan inútiles. Sin funcionamiento, dejan de ser objetos técnicos y no pueden distinguirse de otras formas de materia inerte. Partiendo de este resumen brevísimo de la filosofía stigleriana, podemos decir que los objetos digitales pueden ubicarse *entre* del lenguaje y el utillaje. Los objetos digitales poseen una gramática que habilita su interpretación, como el lenguaje y están condicionados por un medio asociado que habilita su funcionamiento, como el utillaje. A diferencia de un destornillador eléctrico con cabeza Phillips (en cruz) que no sirve para nada si queremos atornillar un tornillo de cabeza plana, un objeto digital como el viejo *Pong* que se jugaba en una consola Atari a fines de los 70 hoy sí puede jugarse en cualquier teléfono celular, emulación mediante. La emulación lo acerca así por ejemplo, más a un cuento de Bukoswski (mal) traducido al español ibérico (a juicio de lectores latinoamericanos), que a una máquina que sólo funciona en determinadas condiciones. La vieja consola, es decir, el medio asociado del cartucho donde estaba el programa del juego, hoy sólo existe como un software que simula el hardware del Atari para poder ejecutar el programa.

El problema de la distinción entre lenguaje y utillaje (así como el de la distinción entre software y hardware) quizá ofrezca pistas para pensar el modo en que el cambio epistemológico ocurrido en 1993 con la aparición de los primeros navegadores con interfaz gráfica que permitieron el acceso masivo a la World Wide Web y que redefinió la brecha entre técnica y cultura. Este cambio hizo del software un fantasma ubicuo, una máquina cultural universal, lo que dio pie a la ilusión de la desmaterialización y reactualizó una forma de dualismo cartesiano.

Como remarca Hui, si los objetos digitales son considerados como objetos técnicos, hay diferentes grados de concretización en diferentes formatos de objetos digitales, y dicho grado de concretización no necesariamente se incrementa con nuevas tecnologías. Hui toma el paso de GML a HTML como formato de documentos en la Web como una pérdida de concretización, ya que HTML es un lenguaje más simple y más liviano. La aparición de, por ejemplo, XML como lenguaje de representación de objetos digitales daría cuenta de este proceso de concretización, “if by concrete here we mean that the concepts of the objects are more well defined and the relations between

parts of the objects and between objects are more explicit—that is, no longer limited by hyperlinks but by parsing and comparing well-structured data.”¹⁵

Sin embargo la propuesta de Hui parece quedar presa de la dinámica intra técnica del propio Simondon y referirse exclusivamente a fenómenos que suceden en la materialidad formal. Simon Mills, extiende la idea de la implicación humana como medio asociado necesario para la concretización de los objetos digitales de internet a partir del análisis de la evolución del sistema global del mercado de intercambio de valores (Forex) y cómo éste involucra no sólo una interacción con un medio físico (natural dado) y el funcionamiento técnico interno sino también una dimensión social, afectiva en términos de Mills, que es parte constitutiva del medio asociado del mercado como objeto digital:

It’s clear that an extraordinary amount of technical infrastructure needs to be in place for the market-system to exist (e.g. global networks, server farms, computing hardware and software). However, this form of technology also requires engagement from the psycho-social in order to generate an associated milieu and become truly technological. This does not just mean economic, political and institutional structures but also the affective engagement of the traders themselves.¹⁶

De hecho, Mills llama la atención sobre el hecho de que como entidades en constante devenir, no clausurados, los objetos digitales, y en especial los que tienen a la Web 2.0 como su medio asociado, se definen a partir de su carácter relacional incluso más que los objetos técnicos previos, y en ese sentido involucran de manera activa en su operación a las tres condiciones señaladas por Thibodeau (la física, la lógica y la conceptual), que deben abordarse en conjunto.

3. Intérpretes y programas

Alan Turing se propuso en 1936 resolver el problema de la decisión planteado por David Hilbert unos años antes, es decir la pregunta por la existencia de un método efectivo de determinar si una sentencia de la aritmética es verdadera o no. Presenta

¹⁵ HUI. *Op. Cit.*, p. 389.

¹⁶ MILLS, Simon. Concrete Software: Simondon’s mechanology and the techno-social. In: *Fibreculture*, 18, 2011, p. 8.

entonces una manera de entender qué es un procedimiento efectivo -o un algoritmo- a través de los procedimientos y estructura de una máquina. El problema formal de la decisión se resuelve a partir de construir una sentencia aritmética que codifique un predicado que será verdadero si una máquina codificada en el argumento del predicado se detiene o no. Turing mostrará además que no hay ningún mecanismo, identificado ahora con esas máquinas, que pueda determinar la terminación de cualquier otra máquina. Hay, asimismo, una de esas máquinas que es capaz de imitar a cualquier otra máquina. En otros términos, dicha máquina particular – la llamada máquina universal- puede tomar como entrada la codificación de una máquina de Turing cualquiera y generar el comportamiento que la caracteriza. Veamos estas ideas de manera más precisa.

Una máquina de Turing puede ser definida a partir de un conjunto finito de estados, un alfabeto finito de símbolos de entrada y una función de transición, la que suele ser presentada como una tabla de transición. Los datos de entrada vienen en una cinta infinita (pero con una cantidad finita de símbolos no nulos), sobre la que habrá una posición diferenciada que indicará el símbolo corriente. Cada una de las filas de la tabla indica para cada estado y símbolo de entrada dados como se cambia el símbolo corriente de la cinta, si se mueve para la izquierda o la derecha la posición corriente, y cuál será el nuevo estado.

En el artículo de 1936, Turing pretende modelar, de alguna manera, el comportamiento de un sujeto que realiza acciones mecánicas para resolver un problema matemático elemental. Así, el “computador” presentado por Turing para realizar las computaciones prescritas por sus máquinas abstractas es una persona equipada con lápiz y papel que a partir de una tabla de transición -una codificación de un comportamiento cuyos datos de entrada están la cinta- va aplicando mecánicamente los pasos prescritos en esa tabla. Cada paso indica posibles modificaciones a una posición de la cinta, un posible cambio de posición y cuál es el siguiente estado. Si se llega a un estado y un símbolo de entrada para el cual no hay ninguna regla en la tabla, el programa termina.

Como decíamos más arriba, la máquina universal es una máquina particular cuyo comportamiento consiste en recorrer la cinta leyendo el código de una máquina dada, y luego comportarse como dicha máquina tomando como entrada el resto de la cinta. Es en este sentido que decíamos que la máquina universal puede imitar el comportamiento de cualquier otra máquina, al menos en lo que respecta al comportamiento de

entrada/salida. Una propiedad importante de esta máquina es que no podemos predecir su comportamiento. Esto se sigue de la solución -negativa- al problema de la detención. Esto es: poder predecir si una máquina de Turing cualquiera dará un resultado -positivo o negativo- o no se detendrá nunca. Esta es una manera muy fuerte de establecer que no podemos saber *a priori* lo que hará una máquina programable. Dado que la máquina universal puede imitar a cualquier máquina y que no puede predecirse el comportamiento de algunas de dichas máquinas no puede predecirse en particular el resultado de evaluar una máquina universal para un conjunto de datos iniciales. Disponemos, entonces, de una máquina particular, la máquina universal, para la cual podemos demostrar que es impredecible, que no hay ningún método efectivo para determinar *a priori* su comportamiento. Por supuesto que una manera de intentar saber cómo se va a comportar es llevar adelante el proceso mecánico que ella prescribe, pero no hay ninguna garantía de que ese proceso arroje algún resultado observable.

Ahora bien, la arquitectura y funcionamiento de una máquina universal destaca un aspecto que se ha transformado en ubicuo en las tecnologías digitales contemporáneas. Dicho aspecto lo constituye la noción de programa. Su centralidad y su carácter funcional se manifiestan más claramente en la máquina universal. Una máquina de Turing tiene un programa que permite, a partir de los datos de entrada, generar un comportamiento. El vínculo entre el programa y dicho comportamiento es prescriptivo en un sentido fuerte. Lo cual, como vimos en el párrafo anterior, no implica que sea posible – en toda circunstancia – determinar qué comportamiento se va a generar. Este vínculo prescriptivo depende, como dijimos, de los datos de entrada, pero constituye un rasgo estructural y funcional que va más allá de dichos datos. Es más, es posible presentar una concepción relacional de computación anclada en la programabilidad. Si un programa es el vínculo prescriptivo que permite generar un comportamiento a partir de un conjunto de datos, entonces la programabilidad de un sistema se puede caracterizar por el tipo de comportamientos generados.

La noción de intérprete ha sido y es central para el desarrollo de la ciencia computacional, pero también tiene un rol explicativo importante y es constitutivo de la relación de programabilidad. Un programa es tal en tanto que existe un intérprete (sea este otro programa corriendo en algún sistema, otro tipo de mecanismo o un humano siguiendo reglas) que puede generar un comportamiento a partir de la prescripción codificada en el programa; un intérprete es tal en tanto existan formas específicas de

codificar comportamientos que puede generar mecánicamente a partir de dichas codificaciones. Referido al código de un programa, el intérprete juega un rol análogo al *digital milieu* que propone Hui.

La separación entre hardware y software se vuelve irrelevante bajo esta reconstrucción relacional de la noción de computación. Las propiedades estructurales son las esenciales y pueden variar, incluso de maneras externamente indiscernibles, entre ser realizadas en el hardware mismo o como programas específicos para un intérprete, también éste constituido por una combinación particular de hardware y software.

A Turing machine is initially envisioned as a machine with mechanical parts, *hardware*. But its code on the tape of the universal machine functions as a *program*, detailing the instructions to the universal machine needed for the appropriate computation to be carried out. Finally, the universal machine in its step-by-step actions sees the digits of a machine code as just more data to be worked on. This fluidity among these three concepts is fundamental to contemporary computer practice. A program written in a modern programming language is data to the interpreter or compiler that manipulates it so that its instructions can actually be executed. In fact Turing's universal machine can itself be regarded as an interpreter, since it functions by interpreting successive quintuples to perform the tasks they specify.¹⁷

La distinción ontológica entre hardware y software, a veces su identificación con cierta forma de dualismo cartesiano y con la existencia de entidades abstractas, pierde de vista que dicha distinción es difusa, de difícil o imposible delimitación y que no se condice con las prácticas de la programación. Ambos, hardware y software, adquieren su modo específico de existencia en relación uno con otro, teniendo de alguna manera primacía el software, por su especificidad, por su versatilidad y porque los problemas más complejos e interesantes aparecen ahí. No es extraño que las tecnologías actuales de desarrollo de hardware imiten al desarrollo de software y hayan incorporado en su diseño técnicas y condiciones de verificación similares a las usadas para el desarrollo de software.

En palabras del pionero de las ciencias de la computación, Edsger Dijkstra:

¹⁷ DAVIS, Martin. *The Universal Computer. The road from Leibniz to Turing*. New York: W. W. Norton and Company, 2000, p. 165.

But before a computer is ready to perform a class of meaningful manipulations - or calculations, if you prefer - we must write a program. What is a program? Several answers are possible. We can view the program as what turns the general-purpose computer into a special-purpose symbol manipulator, and does so without the need to change a single wire (This was an enormous improvement over machines with problem-dependent wiring panels.) I prefer to describe it the other way round: the program is an abstract symbol manipulator, which can be turned into a concrete one by supplying a computer to it. After all, it is no longer the purpose of programs to instruct our machines; these days, it is the purpose of machines to execute our programs.¹⁸

4. Conclusiones indeterminables

El recorrido que seguimos en este trabajo está lejos de haber llegado a conclusiones adecuadas. Encontramos vínculos sugerentes entre campos disciplinares disímiles como son las humanidades digitales, la filosofía de la técnica y las ciencias de la computación. El modo de existencia del software y el hardware fue puesto en consideración en busca de una caracterización que supere formas dualistas que oscurecen la comprensión de estos nuevos objetos, y que no se condicen con las prácticas, tanto científicas como cotidianas.

Una noción relacional de computación, extendida a los objetos digitales en general, permite mostrar que hardware y software tienen más características en común que diferencias ontológicas, y da cuenta de nuevas maneras de estructuración del mundo que tienen consecuencias profundas en nuestras formas de vida. Comprender el origen y las condiciones de posibilidad de los objetos digitales permite conceptualizar sus formas de manifestación y constituir un marco interpretativo adecuado. En particular, varias de las afirmaciones de Manovich aquí consideradas y que a nuestro entender van en la dirección correcta, pueden justificarse mejor en dicho marco, delimitando también sus alcances y límites. Lo mismo sucede con el concepto de medio asociado social de Mills que permite enriquecer el de medio digital de Hui pero que en la que se debe conceptualizar con mayor precisión el modo en el que las interacciones humanas son

¹⁸ DIJKSTRA, Edsger. *On the cruelty of really teaching computing science*. EWD-1036. Universidad de Texas, 1988, p. 14.

incorporadas como datos procesables por los programas. Este trabajo presenta apenas un esbozo preliminar de dicho programa.

Bibliografía

BENJAMIN, Walter. La obra de arte en la época de su reproductibilidad técnica. In: VERA BARROS, Tomás (Comp.) *Estética de la imagen*. Buenos Aires: La Marca, 2015.

DAVIS, Martin. *The Universal Computer. The road from Leibniz to Turing*. New York: W. W. Norton and Company, 2000.

DIJKSTRA, Edsger. *On the cruelty of really teaching computing science*. EWD-1036. Universidad de Texas, 1988.

HUI, Yuk. What is a Digital Object?. In: *Metaphilosophy*. vol. 43, n. 4, 2012, pp. 380-395. New Haven: Southern Connecticut State University/Wiley & Sons.

JACKENDOFF, Ray. *Consciousness and the Computational Mind*. Cambridge, Mass.: MIT Press, 1994.

KITTLER, Friederich. There is no software. In: *CTheory*, Archives, 1995
<<http://www.ctheory.net/articles.aspx?id=74>>

KIRSCHENBAUM, Matthew G. *Mechanisms: New Media and the Forensic Imagination*. Cambridge, Mass: MIT Press, 2012.

MANOVICH, Lev. *Software takes Command*. Cambridge, Mass.: MIT Press, 2013

MILLS, Simon. Concrete Software: Simondon's mechanology and the techno-social. In: *Fibreculture*, 18, 2011,
<<http://eighteen.fibreculturejournal.org/2011/10/09/fcj-127-concrete-software-simondon%e2%80%99s-mechanology-and-the-techno-social/>>

OWENS, Trevor. The is of the Digital Object and the is of the Artifact. In: *The Signal. Digital Preservation*. October 25, 2012
<<http://blogs.loc.gov/digitalpreservation/2012/10/the-is-of-the-digital-object-and-the-is-of-the-artifact/>>

STIEGLER, Bernard. *La técnica y el tiempo I. El pecado de Epimeteo*. Hondarribia: Hiru, 2002.

SIMONDON, Gilbert. *El modo de existencia de los objetos técnicos*. Buenos Aires: Prometeo, 2008

TIQQUN. *La hipótesis cibernética*. Buenos Aires: Hekht, 2016.

TURING, Alan. On Computable Numbers, with an Application to the Entscheidungsproblem. Proceedings of the London Mathematical Society. vol. 2, n. 42, pp. 230–265, 1936.

TURING, Alan. Computing machinery and intelligence. *Mind*, vol. 59, pp. 433-460, 1950.

WEBB, Judson. C. *Mechanism, Mentalism, and Metamathematics: An Essay on Finitism*. Dordrecht: D. Reidel Publishing Company, 1980.