Filosofía e Historia de la Ciencia en el Cono Sur

Selección de trabajos del IX Encuentro y las XXV Jornadas de Epistemología e Historia de la Ciencia

José V. Ahumada
A. Nicolás Venturelli
Silvio Seno Chibeni

Editores

Asociación de Filosofía e Historia de la Ciencia del Cono Sur,
Área Lógico–Epistemológica de la Escuela de Filosofía y
Centro de Investigaciones de la Facultad de Filosofía y Humanidades
Universidad Nacional de Córdoba

Ahumada, José – Venturelli, Nicolás – Chibeni, Silvio Seno (Editores)
Filosofía e Historia de la Ciencia en el Cono Sur. Selección de trabajos del IX Congreso y las XXV Jornadas de Epistemología e Historia de la Ciencia.

Edición técnica: Natalia Rojo
Diseño de tapa: Manuela Eguía

Ahumada, José
Filosofía e historia de la ciencia en el Cono Sur : selección Selección de trabajos del IX Encuentro y las XXV Jornadas de Epistemología e Historia de la Ciencia / José Ahumada; Nicolás Venturelli; Silvio Seno Chibeni; compilado por José Ahumada; Nicolás Venturelli; Silvio Seno Chibeni; editado por José Ahumada; Nicolás Venturelli; Silvio Seno Chibeni.- 1a ed . - Córdoba : Editorial de la UNC, 2015.
716 p. ; 21 x 17 cm.

ISBN 978-987-707-026-2

1. Filosofía de la Ciencia. 2. Historia. 3. Epistemología. I. Ahumada, José, comp. II. Venturelli, Nicolás, comp. III. Seno Chibeni, Silvio, comp. IV. Ahumada, José, ed. V. Venturelli, Nicolás, ed.Seno Chibeni, Silvio, ed. VI.
CDD 501

Correo electrónico:
ejorn@ffyh.unc.edu.ar

Internet: http://www.ffyh.unc.edu.ar/ejorn/
http://www.afhic.com/

La naturaleza de la computación y la computación de la naturaleza

Andrés A. Ilcic *

Well, Mr. Frankel, who started this program, began to suffer from the computer disease that anybody who works with computers now knows about. It's a very serious disease and it interferes completely with the work. The trouble with computers is you play with them. They are so wonderful.

Richard Feynman

Introducción

Sin duda una de las características más definitorias de nuestro actual mundo tecnológico es desde hace algunas décadas el desarrollo de computadoras, las ciencias de las mismas y las ciencias que han cambiado radicalmente desde que las computadoras se volvieron un método irremplazable para emprender nuestro estudio del mundo. Desde el punto de vista filosófico, estudiar la naturaleza de la computación presenta la posibilidad de enfrentar la conexión entre dos campos que han sido trabajadas por los filósofos y los científicos pero que sigue presentando novedades. Esto se debe a que la computación suele presentarse atada tanto a una noción abstracta y matemática, como a una noción que se desprende de las ambiciones científico-técnicas en el diseño mismo de las máquinas de computar. Es esta una de las razones que hacen que muchas de las preguntas que se presentan en el campo de la filosofía de la matemática y de la filosofía de la tecnología tengan un "análogo computacional". Ahora bien, uno de los ítems más importantes, desde mi punto de vista, es la conexión entre la física y la computación. El presente trabajo es el primero de una serie que busca comenzar una exploración filosófica de esta conexión, intentando dar una definición de "computación naturalizada", esto es entender a la computación como un fenómeno natural que ocurre en el mundo y las consecuencias más fuertes que se siguen de sostener esta tesis. Por ser un primer acercamiento a estas cuestiones, las conclusiones que se ven reflejadas son pocas pero sirven para plantear el campo de discusión y abrir nuevas posibilidades de cómo entender al "mundo computacional".

Se empieza por una breve exposición de las nociones clásicas de computación: máquinas de Turing, el cálculo lambda de Church y las funciones recursivas de Gödel-Kleene. Estás son entendidas bajo el concepto de modelo, que ha sido trabajado en filosofía de las ciencias, con la clara diferencia en que bajo la concepción usual los modelos en las teorías científicas, la construcción de estos suele ser guiada por la teoría mientras que en la ciencia de la computación modelo y teoría se vuelven coexstensionales. Expongo brevemente estos modelos que muchas veces son nombrados en la literatura filosófica pero pocas veces se expresa de manera resumida

* Universidad Nacional de Cordoba. ailcic@ffyh.unc.edu.ar

su funcionamiento, con excepción del primer modelo. La segunda parte del trabajo explora las nociones de computación concreta, es decir computación en tanto realizada por un sistema físico.

En esta segunda parte además se hacen explícitos los argumentos que hacen que sea necesario explorar la relación entre la física y la computación, siguiendo a Toffoli (1982) y a Smith (2002). La discusión nos lleva a explorar la naturaleza del universo en términos computacionales, dando lugar a la última sección del trabajo que toma prestado el título de Wheeler (1990) e intenta responder a la pregunta sobre qué tan lejos podríamos llegar con una descripción computacional del universo.

1. ¿De qué habla la teoría de la computación?

La computación se suele definir como una operación que comienza con ciertas condiciones iniciales y un tiempo, y sobre ellas se aplican una serie de reglas de manera tal de producir, en un tiempo distinto, un *output* o resultado que se sigue de esa serie de reglas. Ese conjunto de reglas suele ser llamado algoritmo:

> *Algoritmo. Def.: una descripción finita de pasos en un procedimiento que se vuelve infalible (foolproof):* siempre produce un resultado de un input dado; puede ser implementado en cualquier material ya que su fuerza proviene de una estructura lógica; y los pasos son máximamente tontos, no requieren nada extra para ser llevados a acabo, es decir, nada de tipo intencional.

De la misma manera en que la ciencia hace modelos de los fenómenos en la naturaleza para poder predecir su comportamiento, lo mismo ocurre con la ciencia de la computación. Soy de la creencia de que este fenómeno también es *natural* en el mismo sentido en que lo es un proceso físico, aunque lo es en un sentido muy amplio. Esa amplitud es lo que hace que además de permear muchos aspectos de la naturaleza, sea posible que tantas definiciones bastante diferentes del fenómeno que han sido dadas por científicos y filósofos puedan capturar lo mismo (ser equivalentes) pero nunca de manera completamente abarcadora: siempre dejan algo fuera.[1]

Intentemos ahora una definición de computación, lo más abarcadora posible.[2]

> *Computación. Def.: La computación es un método para convertir elementos de un conjunto en elementos de otro conjunto.3*

Usualmente los elementos del conjunto son números, aunque uno bien podría arreglárselas con otra clase de elementos (pagando un precio, normalmente sobre la velocidad del cálculo). Un modelo de computación es una descripción de un método para producir el conjunto de instrucciones necesario para realizar una computación. Durante el siglo XX surgieron muchos de estos modelos. A manera de refrescar, veamos ahora los modelos abstractos (en tanto no consideran los detalles materiales de la implementación) más estudiados. Estos son (1) las máquinas de Turing, (2) el cálculo Lambda de Church y (3) las funciones recursivas de Gödel-Kleene. Todos surgen, al menos parcialmente, de dar una respuesta posible al *Entscheidungsproblem* o problema de la decisión que fue formulado en la manera en que lo conocemos hoy por David Hilbert en su (1928, 72-3), quien ya lo había anticipado en otro tono como el 2do problema en la famosa conferencia de Paris de 1900, que no es otro que el problema de la consistencia de la matemática.

Las máquinas de Turing son dispositivos hipotéticos propuestos por Alan Turing en 1936, que proveen una formalización de la noción de algoritmo. Consisten en una cabeza de lectura y escritura montada sobre un pedacito infinito de papel que está dividido en pequeñas celdas. El cabezal puede leer, borrar y escribir símbolos sobre la celda, una acción a la vez por unidad de tiempo de acuerdo a un programa que le dice qué hacer en el siguiente instante dado el estado en el que se encuentra ahora. Existe un estado inicial (los símbolos que estaban en el papel antes de empezar) y si hay solución al menos un estado "de parada", en el que la máquina termina sus operaciones.

(2) Fue Alonzo Church el que notó que los algoritmos pueden ser entendidos como una función matemática que mapea ciertos estados (*inputs*) a otros (*outputs*). Este modelo consiste en un lenguaje de λ-términos y un conjunto de reglas que permiten las transformaciones de estos términos, definidos inductivamente de tal manera de poder dar cuenta de que si x es una variable es un término; si t es un término y x una variable, la λ-abstracción $\lambda x.t$ es un término; y si t y s son términos, la aplicación ts es un término. Los casos interesantes son más complejos como puede ser $(\lambda x(\lambda t_1)).\lambda t_2$ en donde los t son términos y x una variable ligada y toda la fórmula representa una manera de reescribir la expresión de la izquierda reemplazando cada instancia de la variable ligada por λt_2. (Por ejemplo, $(\lambda x(fx)).b$ es fb). Un programa en este modelo corresponde a la expresión $(\lambda x(\lambda t_1))$ y las condiciones iniciales $.\lambda t_2$. El resultado es un λ-término que resulta de la operación continua del programa aplicado sobre el *input*, un proceso llamado β-*reducción* y que además de ser el corazón del modelo, nos hace acordar al lenguaje de programación LISP.

(3) La idea detrás de este modelo se le debe a Gödel en 1934, y fue extendido por Kleene durante la década del 1930 pero especialmente en Kleene 1952. Las funciones recursivas generales (FRG) se componen desde un pequeño conjunto dado de reglas que definen funciones sobre el conjunto de los números naturales y con ellas se pueden construir más funciones. Un programa en este modelo es una FRG construida con las reglas más simples, el *input* sobre el que opera el programa es el número natural que se toma por argumento y el resultado de su ejecución es nuevamente un número natural. Por lo general se toman como primitivas las funciones que resultan de estos axiomas:

_ Una función constante o cero: $\forall x, f(x)=0$

_ La función sucesor es recursiva primitiva: $s(x)=x+1$

_ La función proyección: una FRG puede dar como resultado alguno de sus argumentos. Esto es que para alguna $f(x_1,\ldots,x_n)=x_i$

_ Composición: una nueva función puede ser definida por la composición de dos o más funciones. Si *f(x)* y *g(x)* son recursivas, también lo es *f(g(x))*

_ Recursión: las funciones recursivas pueden tener definiciones recursivas.[4]

La equivalencia entre estos modelos es lo que suele ser llamado como *Tesis de Church-Turing*, aunque ninguno de ellos se refirió nunca a sus propuestas como "tesis" sino como "definiciones". Kleene (1952) se refiere a ellos como "la tesis de Turing" y "la tesis de Church", y curiosamente con el tiempo el segundo término empezó a referir al primero, quizás del hecho de que la comunidad empezó a igualar "recursivo" con "computable".

2. ¿Y si la "computación" está ahí afuera?

Ahora ya sabemos qué quiere decir computar abstractamente. Sin embargo, la pregunta que guía este trabajo es si la computación puede entenderse como un fenómeno que está ahí afuera, un fenómeno de la naturaleza. Sin descartar algunas otras posibilidades en el espectro, creo que hay tres formas de encarar la pregunta, que acomodadas en orden de radicalidad de la propuesta se pueden ver como (1) computación concreta; (2) los aspectos físicos de la computación; y (3) la física digital.[5]

En Piccinini (2010) encontramos cinco visiones sobre la computación concreta, es decir de la computación tal y como es realizada por un sistema físico. La primera versión es la de Putnam (1960) que básicamente sostiene aquello que puede ser descrito en términos de una descripción computación está realmente implementando esa computación. Mientras que la primera versión era en términos de máquinas de Turing, Putnam lo extendió a una descripción general de un sistema S que implementa una computación si (i) hay una función que mapea algunos estados de S a estados de una descripción computacional C, y (ii) que para cada transición de estado en C hay una transición análoga en S. La obvia consecuencia de que el conjunto de los elementos que terminan realizando computaciones es elevado puede ser moderada si restringen el tipo de mapeos aceptables, por ejemplo, requiriendo que los mapeos dependan de una relación causal entre los estados, o una que soporte contrafácticos, o una con una descripción disposicional.

Otra de las visiones que podemos encontrar es la de Fodor, a quien le debemos el eslogan "no hay computación sin representación" (Fodor, 1981, 180). Esta postura implica una restricción aún mayor sobre los mapeos, esta vez semántica, en tanto sólo se van a aceptar como descripciones computacionales los mapeos que vayan de estados físicos que cuentan como una representación a los estados computacionales, y es una de las más usadas por los filósofos de la mente. Claro que uno puede no comprometerse con la semántica y hasta dejarla completamente de lado, sosteniendo una postura sintáctica. En ésta, la computación queda ligada a clases de estructuras lingüísticas manipuladas sólo por sus propiedades sintácticas, por lo que sólo de los estados físicos que efectivamente sean considerados como sintácticos se dirán que son computables. Algunos autores han ido tan lejos como para sostener que una sintaxis bien curada puede reemplazar por completo a la semántica. Haugeland, por caso, es uno de los que sostiene que

> de manera tal de establecer la verdad semántica de un símbolo [*token*] en tal sistema [formal], alcanza con meramente probarlo formalmente (jugar el juego). Así es cómo las "dos vidas" de los símbolos se juntan; y es la idea básica detrás de la formalización de la lógica moderna y la matemática. De hecho, dado un sistema formal interpretado con axiomas verdaderos y reglas que preservan la verdad, si te encargas de la sintaxis, *la semántica encargará de ella misma* [if you take care of the syntax, *the semantics will take care of itself*]. (Haugeland, 1981, 44. Énfasis en el original)[6]

La otra postura que se encuentra en la literatura es la del mismo Piccinini (2007, 2008), quien da cuenta de la computación concreta de una manera netamente mecanicista, lo que le permite descartar tanto a la semántica como a la sintaxis puesto que serán considerados como sistemas de computación concretos aquellos sistemas cuyas función mecánica y organización le permita hacer computaciones. Por explicación mecanicista de un sistema X, el considera "una descripción de [un sistema] X en término de los componentes espaciotemporales de X, sus funciones y su organización, al efecto de que X posee sus capacidades a causa de cómo los componentes de X y sus funciones están organizadas" (Piccinini 2007, 506). Sostiene que, en oposición a una simple explicación causal, las explicaciones mecanicistas pueden distinguir entre las operaciones correctas y los errores de dicho sistema, aunque éstas no pueden darnos la razón de por qué algunos sistemas computan y otros no (2007, 508). Acota la computación a aquellas propiedades funcionales que son relevantes para la individuación computacional, es decir a

> la presencia en un mecanismo de ciertos componentes (tales como memorias, procesadores, etc.), relaciones relevantes entre los componentes (tales como la transmisión de señales), el estado de estos componentes (tales como letras de un alfabeto discreto y estados digitales monádicos) y las funciones de los componentes (tales como realizar operaciones sobre letras). (Piccinini, 2008, 209).

Bajo esta concepción, entonces, una descripción "abstracta" de un sistema físico es en realidad una descripción de un sistema concreto que omite ciertos detalles, por lo que la computación concreta también puede ser descrita independientemente del medio que la implementa o de la que es una instancia. Como los algoritmos, son independientes del medio.

Si bien la postura de Piccinini me parece de las más interesantes, no creo que sea suficiente para dar cuenta de qué es lo que la computación permite hacer, y es la parte que se deja entrever de los modelos de computación abstractos que presentamos antes y que ahora podemos señalar. Si bien es necesario un mecanismo, no es condición suficiente y lo que hay que agregar es precisamente lo que Piccinini y otros querían dejar afuera: la semántica. Esto se debe a que al querer hacer *algo* con una computación, esa transformación tiene que ser considerada como *significando algo* más. Esa semántica, ese *sentido*, no tiene por qué se representacional, no es necesario pensar que la computación representa algo intrínsecamente. Así, aunque la computación ocurre naturalmente en algunos sistemas tanto naturales como artificiales, de manera tal de poder obtener *outputs* que sean útiles, algo *externo* al sistema computacional propio tiene que ser el "agente" que realiza el mapeo. Los sistemas no *computan* por computar, lo que hacen es producir un resultado desde ciertas condiciones iniciales. Y esta forma de colocar las cosas nos trae de vuelta a la física, la segunda forma de responder a la pregunta que señalé antes. La historia de la física y la computación ya es bastante larga pero aprovechemos la ocasión para recordar la conferencia que se organizó entre el 6 y el 8 de mayo de 1981 en el MIT, que buscaba reunir a expertos en las dos disciplinas que creían en la fuerte conexión entre las mismas.[7] En ella, Tomaso Toffoli, uno de los organizadores, comentaba que cualquier forma de computación, ya sea por un humano o por una máquina, es una actividad física y que de manera tal de computar mejor, debemos descubrir más acerca de cómo funciona la naturaleza, extendiendo a la física misma por medio de la computación. Acerca de la demanda por parte de una nueva física hacia la teoría de la computación, comentaba:

> ¿Cuáles son las circunstancias que nos llevan a hacer estas nuevas demandas a la teoría de la computación? Mientras que la teoría evolucionó como una rama abstracta de la matemática, ahora nos estamos dando cuenta de que mucha de la computación que queremos hacer está caracterizada por una restricción importante: debe ser llevada a cabo ya en asociación con la naturaleza, ya en contra de ella. Las teorías convencionales de la computación modelan a la naturaleza lo suficientemente bien como para decirnos *qué* puede ser computado, pero no lo suficiente como para decirnos *cómo* computar mejor. (Toffoli, 1982, 167. Énfasis en el original.)

La primera razón por la que una ciencia que puede ser "abstracta" debe considerar a la física, de nuevo siguiendo a Toffoli, es para evitar el platonismo: si bien es posible obtener una descripción abstracta de la información, esa información sólo puede existir si es sostenida por un soporte físico, aunque éste no sea necesariamente *material*, en tanto puede ser bajo la forma de la interacción entre partículas. Por otro lado, también está la conexión entre la computación y la termodinámica, en tanto desde Landauer (1961) sabemos que la computación necesariamente disipa calor al borrar un bit de información. Hace poco esto ha sido verificado experimentalmente por Bérut *et al.* (2012). Por último, Toffoli también nos recuerda que "los mismos axiomas de la computabilidad y de la teoría de la complejidad computacional son una estilización de ciertas restricciones físicas, como Turing y von Neumann ya sabían muy bien" (Toffoli, 1982, 169).

Sin embargo, todavía nos queda la duda de cuál es precisamente el fenómeno *natural* que la computación está modelando. Y éste no es otro que la causalidad misma: *cómo* cambian las cosas, en el sentido más elemental posible de "cambiar".

> Aunque todavía no se la reconoce como tal, la teoría matemática basada en la teoría de la recursión, las máquinas de Turing, los análisis de complejidad y demás –ampliamente conocida como la "teoría de la computación"– no es ni más ni menos que *una teoría matemática del flujo de la causalidad*. (Smith, 2002, 43. El énfasis es nuestro.)

Tenemos aquí nuestro fenómeno. Preguntémonos, antes de sacar las conclusiones de esta primera exploración, qué tan lejos se puede llegar con una descripción computacional de las cosas.

3. La computadora y el universo[8]

Básicamente, al describir las cosas como una computadora se puede llegar tan lejos como uno quiera. Hay varias versiones de esta posición "pancomputacionalista". La primera es la versión *ilimitada*, debida básicamente a Putnam y a Searle, que sostiene que cualquier sistema físico lleva a cabo *toda* computación. Otra versión es la llamada interpretativista que sostiene que si se puede ver como una computadora se trata de una computadora. Por otro lado, encontramos una versión limitada que sostiene que todo sistema físico realiza alguna computación o, en una versión más débil aún, unas pocas computaciones. Existe también una versión causal que se sigue de la descripción de la causal de la computación descrita anteriormente en la que si todo tiene una estructura causal, entonces todo sistema causal hace efectiva la computación definida por esa estructura causal. La descripción de la computación como procesamiento de información también puede verse como dando origen a una tesis pancomputacionalista dado que todo estado físico conlleva información. Sin embargo, la tesis más interesante es la versión óntica del pancomputacionalismo que sostiene que en la descripción más profunda posible de todo, el mundo mismo *es* computacional. En esta afirmación es necesario notar que se están haciendo dos afirmaciones, por un lado un enunciado empírico y uno metafísico por otro.

El empírico es sostener la posibilidad de que las magnitudes y transiciones de estado que se dan en una descripción computacional sean tal y como son dadas por esa descripción, esto es, sin aproximaciones. Esta descripción suele venir en dos sabores, uno clásico y uno cuántico. El primero cuenta como modelo por excelencia a los *automata* celulares y fuerza una descripción fundamental del universo en términos discretos en los que cualquier aleatoriedad se vuelve tan solo aparente en el mejor de los casos y en la que no hay magnitudes medidas en números reales. La versión cuántica del pancomputacionalismo no va tan lejos como para requerir que el mundo sea fundamentalmente discreto sino que se contenta con postular una descripción del universo que no hace uso de los dígitos binarios sino de *qudits* (de los que el sospechoso usual es el famoso *qbit*) y que permite un número no denumerable de estados gracias a la superposición cuántica. Si bien es cierto que actualmente nuestras mejores descripciones físicas de los fenómenos básicos respetan las leyes de la mecánica cuántica, puede ser que no sea el caso de *ser* necesario puesto que en principio no hay evidencia en contrario de que el universo pueda ser fundamentalmente discreto. Sigo a 't Hooft (2005) en mantener esta tesis, que hace a la condición de posibilidad de la llamada física digital.

El enunciado metafísico que se desprende de esta tesis pancomputacionalista suele ser mantenido también por quienes afirman el empírico pero son lógicamente independientes. Éste sostiene que el universo mismo está hecho de computación, por lo que la computación es ontológicamente anterior a todo lo demás. Esto claramente va en contra de la noción tradicional de que la computación requiere de un substrato físico sobre el que ser implementado; ahora tendríamos sistemas físicos que computan ("hardware") construido por una descripción "soft" del universo. Si suponemos que el universo es, por ejemplo, un autómata celular, la descripción tradicional sostiene que existe *ahí afuera* una entidad física que corresponde a una celda en la grilla. Pero bajo la nueva afirmación ontológica, no hay tal cosa *fundamentalmente*, todo lo que "hay" en el fondo son entidades computacionales.

4. Conclusiones y perspectivas

Hemos visto algunas de las razones por las que creemos que es necesario pensar a la computación en términos físicos y por las que los "procesos computacionales" son en realidad abstracciones de procesos físicos, y también hemos señalado la posibilidad que habría de ir en la otra dirección, esto es de considerar a los procesos computacionales como primitivos y derivar de ellos los físicos. Bajo esta mirada, la computación sería una descripción de qué es posible, en la concepción más amplia de "posibilidad", y al considerar los aspectos físicos estamos restringiendo ese universo de posibilidades a lo físicamente posible. La noción de mecanismo es incompleta para dar cuenta de qué ocurre en una computación, dado que algo así como una semántica externa va a tener que ser agregada a esa configuración. Otro factor que empieza a aparecer al considerar estos factores es la de información, que ahora hasta surge como un candidato para la base ontológica del mundo.

Si bien hay mucha especulación en todo esto, parece bastante seguro afirmar con Smith que:

> Lo que ha sido llamado (y para casi todos todavía es) una "Teoría de la Computación" es de hecho una teoría general del mundo físico —específicamente, una teoría de qué tan difícil es, y qué se requiere, para que partes del mundo en una configuración física cambien a otra configuración física. Se aplica a todas las entidades físicas, no sólo a las computadoras. No es más matemática que el resto de la física, en tanto usa estructuras matemáticas (abstractas) para modelar fenómenos físicos (concretos). A fin de cuentas, entonces, debe ser unida con la física —porque en un sentido es física. (Smith 2002, 42)

Esto nos hace enfrentar la recomendación de rehacer nuestro "mapa intelectual" desde un punto de vista bastante distinto del tradicional. Reconocer que la teoría de la computación es en realidad una teoría general del mundo físico es el eje central y nos lleva a requerir una seria consideración de las tesis de la posición que ha dado a llamarse "física digital" cuyas consecuencias epistemológicas, matemáticas y físicas recién empiezan a ser exploradas y aquí sólo han sido vislumbradas. Un claro ejemplo de esta propuesta, junto con un programa de investigación, es *hacer* un nuevo tipo de ciencia, al estilo de Wolfram (2002), que es un redescubrimiento de una tesis que aparece por primera vez en Zuze (1967). Está claro que no parece ser el lugar hacia donde la ciencia (ni la física ni la computación) se dirige actualmente, pero en principio nada impide que pueda serlo.

Notas

1. Presumiblemente, esta serie de trabajos también dejen bastante afuera, pero espero que menos. Un hecho curioso parece ser que en las ciencias naturales suele existir una teoría guiando la construcción del modelo (y el modelo haciendo de mensajero entre el fenómeno y la teoría) mientras que en la ciencia de la computación modelo y teoría parecen ser coexstensionales.

2. Naturalmente, recurrimos a la teoría de conjuntos.

3. Uno quisiera a veces ser lo suficientemente *quineano* para decir que dada una ontología basada en conjuntos no hace falta más nada para decir que la computación es "natural".

4. Quizás la tendencia a sólo estudiar el modelo Turing de computación venga por la sugerencia del mismo Gödel, quien reconoció que la formulación de Turing era la mejor: "El progreso más significativo fue posible gracias a la definición precia del concepto de procedimiento finito [...] Este concepto [...] es equivalente al concepto de una "función computable de enteros" [...]. La forma más satisfactoria, en mi opinión, es la de reducir el concepto de procedimiento finito al de una máquina con un número finito de partes, tal como ha sido realizado por el matemático británico Turing. (Gödel, 1995, 304-5).

5. También creemos que la tendencia de pasar de (1) a (3) es más bien "natural" pero no lo defendemos, aquí, aunque quizás se desliza "naturalmente" en el texto.

6. Todas las traducciones son del autor.

7. Fue en esta conferencia que alguien, Richard Feynman, sugirió por primera vez la necesidad de contar con computadoras cuánticas para poder simular los fenómenos descritos por la teoría cuántica.

8. Como dijimos antes, el título de esta sección se le debe a Wheeler (1982), que es su trabajo presentado en la conferencia que fue mencionada en la sección anterior. Dicho sea de paso, él fue quien acuñó el ahora famoso lemma "It from bit" o de dónde salen las cosas: "*It from bit.* O dicho de otra manera, todo *'it'* —toda partícula, todo campo de fuerza, incluso el continuo espacio tiempo mismo— deriva su función, su sentido [*meaning*], su misma existencia enteramente—incluso si en algunos contextos es sólo indirectamente— de las respuestas inducidas por un aparato a preguntas del tipo sí-o-no, opciones binarias, bits." (Wheeler, 1990, 5)

Bibliografía

BÉRUT, A., ARAKELYAN, A., PETROSYAN, A., CILIBERTO, S., DILLENSCHNEIDER, R., & LUTZ, E. (2012). Experimental verification of Landauer's principle linking information and thermodynamics. *Nature*, *483*(7388), 187-189. http://doi.org/10.1038/nature10872

FODOR, J. A. (1981). The Mind-Body Problem. *Scientific American*, *244*, 114–25.

GÖDEL, K. (1995). Collected works, vol. III. *Unpublished Essays and Lectures. Oxford University Press, New York*.

HAUGELAND, J. (1981). Semantic Engines: An Introduction to Mind Design. En *Mind Design*. MIT Press.

HILBERT, D., & ACKERMANN, W. (1928). Grundzüge der theoretischen Logik. *Berlin, Heidelberg*.

HOOFT, G. ’T. (2001). *How Does God Play Dice? (Pre-)Determinism at the Planck Scale* (arXiv e-print No. hep-th/0104219). Recuperado a partir de http://arxiv.org/abs/hep-th/0104219

KLEENE, S. C. (1952). *Introduction to metamathematics.* WoIters-Noordhoff and North-Holland.

LANDAUER, R. (1961). Irreversibility and heat generation in the computing process. *IBM journal of research and development*, *5*(3), 183–191.

PICCININI, G. (2007). Computing Mechanisms. *Philosophy of Science*, *74*(4), 501–526.

PICCININI, G. (2008). Computation Without Representation. *Philosophical Studies*, *137*(2), 205–241.

PICCININI, G. (2010). Computation in Physical Systems. En *Stanford Encyclopedia of Philosophy*.

PUTNAM, H. (1960). Minds and Machines. En *Dimensions of Mind* (pp. 57–80). New York University Press.

SMITH, B. C. (2002). The foundations of computing. En *Computationalism: new directions*, 23–58.

TOFFOLI, T. (1982). Physics and computation. *International Journal of Theoretical Physics*, *21*(3-4), 165-175.

TURING, A. (1936). On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, *42*, 230–265.

WHEELER, J. A. (1982). The computer and the universe. *International Journal of Theoretical Physics*, *21*(6-7), 557–572.

WHEELER, J. A. (1990). Information, physics, quantum: The search for links. En (W. Zurek, ed.) *Complexity, Entropy, and the Physics of Information.* Redwood City. *CA: Addison-Wesley.*

WOLFRAM, S. (2002). *A new kind of science*. Wolfram media Champaign.

ZUSE, K. (1967). Rechnender raum. Elektronische Datenverarbeitung 8 (1967) 336–344.