



## Datos y semántica en computación Análisis de la discusión de Fresco y Wolf con Floridi

Xavier Huvelle\*

### Introducción

En los últimos años la discusión sobre la naturaleza de la información ha resurgido en gran parte gracias a Floridi. Pero esta discusión no es nueva y ya fue ampliamente abordada por la Cibernética a principios del Siglo XX, en particular por Wiener (1948), Ashby (1960), Shannon (1948) y Simon (1947). Pero este abordaje e identificación contemplan nociones muy dispares acerca de la información, o por lo menos plantean funciones distintas. Van Benthem y Adriaans (2008) enumeran por lo menos tres grandes tradiciones que se fueron conformando con el tiempo: una noción de información algorítmica y compresión (tipo Kolmogorov), una noción probabilística y cuantitativa de la información (tipo Shannon) y una noción basada en conocimiento y lógica, en la que un agente usa y representa información. Resultaría complicado poder encajar a los Cibernéticos en alguna de estas nociones dado que comparten varios elementos de estas perspectivas. Por ejemplo, plantear una noción de información sin agente y dejar de lado la teoría de la información de Shannon resultaría imposible para Simon. En cambio, para Shannon el estatuto del agente no importa, sino que importa lo que se transmite, no por quién. Hoy en día, estas cuestiones son relevantes y nos permiten observar conflictos en algunos de los supuestos que proponían los Cibernéticos, en particular acerca de la naturaleza de la información. Aunque los Cibernéticos hablan en términos “físicos” lo que más interesa para estos autores es responder la pregunta por cómo se procesan los datos y la información, pero no realmente sobre lo propiamente físico de la información. ¿Es la información una señal eléctrica, un signo lógico u otro elemento? Fresco y Wolf (2014), para responder a esta pregunta, propusieron una noción de computación física no simbólica basada en

\* Universidad Nacional de Córdoba (UNC), Facultad de Filosofía y Humanidades, Córdoba, Argentina.

xavier.huvelle@gmail.com

la *información instruccional* de la teoría de la información propuesta por Floridi (2010, 2011). En un artículo de 2018, Fresco y Wolf vuelven sobre esta noción de información instruccional y proponen cambiarla por una *información funcional* de tipo *teleológica* basada en Corning (2007). En este trabajo presentamos esta discusión entre Fresco, Wolf y Floridi respecto de la información instruccional, además presentamos tres críticas basadas en la lógica programada. Estas críticas cuestionan la propuesta de Fresco y Wolf para plantear si la información instruccional (funcional) que proponen está bien caracterizada, si su propuesta puede evitar el *pancomputacionalismo* y si pueden prescindir de un contenido semántico externalista como ellos afirman.

## Discusión

El origen de la concepción actual de información se encuentra atado con la teoría de la información de Shannon y de la Cibernética (Ashby 1960; Wiener 1948). La perspectiva tomada por Shannon y Wiener se basa en el aspecto ingenieril de transmisión de la información y no tanto con los elementos semánticos que la componen. Floridi (2010, 2011) ofrece, por su lado, una noción muy peculiar de información semántica que distingue con una caracterización de los *datos*. Los datos en esta perspectiva son la falta de uniformidad entre por lo menos dos variables no interpretadas que son distintas una de otra y que han sido dejadas para futuras interpretaciones (Floridi, 2011, p. 85). Mientras que la información semántica se trata de una definición formal en la que un objeto  $O$  es una instancia de información semántica, entendida como contenido semántico, si y solo si: (a) consiste en  $n$  data  $D$ ; (b)  $D$  está bien formado; (c)  $D$  es significativamente independiente de su recipiente (Floridi, 2010, p. 21). El punto (a) es lo que se entiende por un dato positivo y (b) significa que el dato es estructurado de acuerdo a algunas reglas (reglas sintácticas, por ejemplo). Cuando (a) y (b) son combinados, forman la forma más simple de información: *Datos estructurados* (Fresco & Wolf, 2014, p. 1472). Mientras que (c) es el *contenido semántico* y requiere que el dato estructurado se adecue con la semántica del sistema o del código. Fresco y Wolf (2014) recuperan las caracterizaciones ofrecidas por Floridi para elaborar una noción de información para la computación digital. La información instruccional es entonces prescriptiva y no descriptiva. Fresco y Wolf dirán que es más bien imperativa y la definen de esta manera: “el contenido semántico es

una instancia de la información instruccional si y solo si su satisfacción resulta en una acción en un contexto dado” (2014, p. 1473).

No hizo falta esperar mucho para que en un artículo de Fresco y Wolf (2018) se pueda leer: “La conclusión principal de este artículo es que, si adoptamos la posición conceptual de Floridi sobre información para formular una postura de procesamiento de la información de la computación, [esta] nos lleva a un callejón sin salida conceptual” (Fresco & Wolf, 2018, p. 2). En otras palabras, hay que abandonar la idea de información instruccional de Floridi. Según Fresco y Wolf (2018, p. 5), podemos interpretar los datos en Floridi en relación con lo que se diferencian: (a) diferencias *de re* que son una falta de uniformidad en el mundo real considerado como “datos puros” o “datos proto-epistémicos”; (b) diferencias *de signo* que son una falta de uniformidad entre por lo menos dos señales que por ejemplo son menos o más capacidad de carga de un teléfono o una señal eléctrica variable en una conversación telefónica; (c) diferencias *de dicto*, que son una falta de uniformidad entre dos símbolos, por ejemplo la diferencia en el alfabeto griego entre  $\alpha$  y  $\beta$ . La forma analizada por ambos autores es la (b), *de signo*, pero para ellos no queda claro si la definición de Floridi para las señales solo se aplica en un mundo físico. Si fuera así, Fresco y Wolf afirman que esta definición no sería válida para toda función matemática o máquinas de Turing. Las máquinas de Turing, por ejemplo, caerían en el dominio de diferencias *de dicto* en el que los datos mostrarían una falta de uniformidad entre símbolos (cuadrados vacíos y una marca en el cuadrado). Pero en el caso de las funciones matemáticas no podemos describir como símbolos a los miembros de un dominio o co-dominio siendo un subconjunto de los reales. Por lo tanto, se podría interpretar a estos datos de diferencias *de signo* para tratar sobre entidades abstractas y físicas. Dos nociones de diferencias son ofrecidas por Fresco y Wolf (2018, p. 6), una *actual* en la que dos señales son actualmente distintas y una *potencial* que existe, pero podría no haber existido. Estos conceptos buscan explicar por ejemplo cómo caracterizaríamos un dato (singular) en la computación de una puerta AND de dos entradas y una salida (Fig. 1). En donde el *input A* procesa una señal (dato singular) de un voltaje entre 0.1 V y 0.5 V (bajo), mientras que el *input B* procesa una señal de un voltaje entre 1.1 V y 1.5 V (alto). En caso que obtenga un voltaje entre 0.6 V y 1 V será considerada como indefinida.



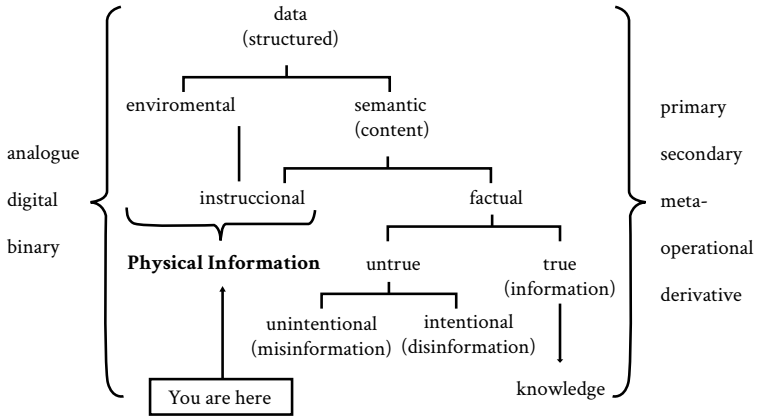
**Fig. 1:** Ejemplo de puerta AND de dos entradas y una salida, donde A es un voltaje bajo entre 0.1 V y 0.5 V, y B es un voltaje alto entre 1.1 V y 1.5 V.

A partir de estas nociones de *actual* y *potencial*, Fresco y Wolf ofrecen cuatro posibles interpretaciones:

- i) El input A y el input B tienen una señal *actual* y el datum es la diferencia entre estas dos señales.
- ii) El input A y el input B tienen cada uno una señal *actual* y cada señal es un datum de la distinción espacial de ambos.
- iii) El input A y el input B son distintos en tanto que uno es *actual* y el otro *potencial*.
- iv) El input A y el input B son ambas señales (alta o baja) que pueden *potencialmente* existir o no. (2018, p. 8)

La interpretación i) es problemática, dado que no permite diferenciar entre el *datum* del *input* A y aquel del *input* B. Por ejemplo, tendríamos tres posibles resultados: uno en el que no hay diferencias, otro en el que el *input* A tiene un voltaje alto y otro en el que el *input* B tiene un voltaje alto. En este sentido nunca podríamos determinar los valores del *output* ya que no podríamos distinguir el voltaje alto o bajo de A y B. ii) es también problemática por dos razones: 1) Si A y B reciben 0,5 V y que la diferencia entre ambas es solo espacial (A y B). ¿Qué diferencia A con 0,5 V de B con 1,1 V? Ninguna. Pero a pesar de esto serán procesados distintamente por la puerta ya que el voltaje implica un cambio en su valor de verdad. Y 2) Una señal ocupando distintas regiones en el espacio no se aplica en una puerta NOT ya que solo tiene una única entrada. iii) es directamente inconsistente ya que no cumple con el diseño que posee la puerta dado que, si uno de los *inputs* posee una señal *actual* y el otro no,<sup>1</sup> entonces no habrá *output* posible. Solo si la señal fluye en A y B tendremos a Q.

<sup>1</sup> Al no ser *actual* es *potencial*, si es *potencial* significa que es posible que tenga un valor X. Las puertas lógicas no permiten posibilidades si una de las entradas es *actual*.



**Fig. 2:** Cuadro de Floridi jerarquizando los tipos de información y ubicando la información instruccional.

*Nota:* Imagen extraída de *Information: A very short introduction* (fig. 11) de L. Floridi, 2010, Oxford University Press.

En la interpretación iv), la puerta AND procesa datos debido que el input A y el input B son señales (*datum*) que se distinguen por la presencia o ausencia de una señal de alto voltaje o bajo voltaje (diferencia potencial). Un “enabler”<sup>2</sup> asegura la presencia o la ausencia de la corriente por lo que el input A y el input B siempre tendrán un estado. En caso de que el “enabler” tenga una memoria, dicen Fresco y Wolf (p. 9), la puerta booleana hará uso de la memoria para procesar los datos y por lo tanto convirtiendo todos los circuitos booleanos en poseedores de estados.

El punto iv) motiva a Fresco y Wolf a preguntarse si la definición de dato ofrecida por Floridi es realmente no interpretada, además de que la propia lógica booleana describe el funcionamiento físico de las puertas en 0 y 1 que, ya de por sí, son una interpretación. Concordamos con esta idea de que existen interpretaciones en estos dos niveles, complicando la perspectiva de “dato puro”<sup>3</sup> avanzada por Floridi. El ejemplo propuesto por Fresco y Wolf nos muestran la presencia de algún contenido semántico

<sup>2</sup> Un *enabler* es para, Fresco y Wolf (2018, p. 7), una característica que permite hacer un puente entre la separación conceptual o temporal que existen entre una señal potencial y una actual, en donde indica la posibilidad contrafactual de que una señal potencial podría haberse manifestada o de que la señal actual no se habría manifestada.

<sup>3</sup> Por dato “puro” queremos decir “dato no interpretado”.

que posee la señal y que puede interpretarse como dato. La cuestión ahora es cómo establecer el contenido semántico y qué sería “significar”. Para Floridi este proceso se da a partir de la información instruccional.

Información instruccional aparece en Floridi como un subtipo de “contenido semántico” o de “información ambiental” (Fig. 2). Ambos (contenido semántico e información ambiental) se distinguen respecto de si se requiere de un “significado” de información o no. En el caso de información ambiental se busca ofrecer un tipo de información que posee un significado de forma independiente a un productor inteligente. Por ejemplo, las puertas booleanas no involucran semánticas según Floridi (2010, p. 35). Mientras que el contenido semántico depende de que los datos se ajusten al significado (semántico) del sistema, código o lenguaje elegido. De este modo la información es significativa independientemente del receptor. El problema que ven tanto Fresco y Wolf (2018) como Corning (2007) es: ¿cómo podemos caracterizar lo que es significativo o no? En Floridi el proceso no es claro, y si se sale de un entorno de computación simbólica es aún más problemático debido que existen, como se mostró en el ejemplo de los circuitos booleanos, algunas formas de interpretación.

Fresco y Wolf (2018) afirman entonces que hay cuatro mayores problemas con las nociones de Floridi: (a) que la noción de “dato” no se encuentra bien definida para los sistemas físicos; (b) el requisito de forma adecuada<sup>4</sup> [*well-formedness*] se aplica para los sistemas simbólicos, pero no es claro cómo se aplica en sistemas físicos no simbólicos; (c) el requisito de significado implica una teoría de la significación que no es dada por Floridi; y (d) la información instruccional se encuentra definida desde la teoría *General de la Información* propuesta por Floridi.

Una salida interesante es la propuesta por Fresco y Wolf (2018) que se basa en una teleología funcional del receptor/usuario y en donde diferencian de forma jerárquica los conceptos *datum* (dato solo), *signo*, *señal* y *símbolo* siendo el más abstracto. Fresco y Wolf los definen de la siguiente manera:

- a) Datum es un acto sistemático, evento, proceso o estructura (o su ausencia) desde el cuál un receptor puede, pero no todavía, funcionalmente responder dado que es sensible a las variaciones de sus formas.

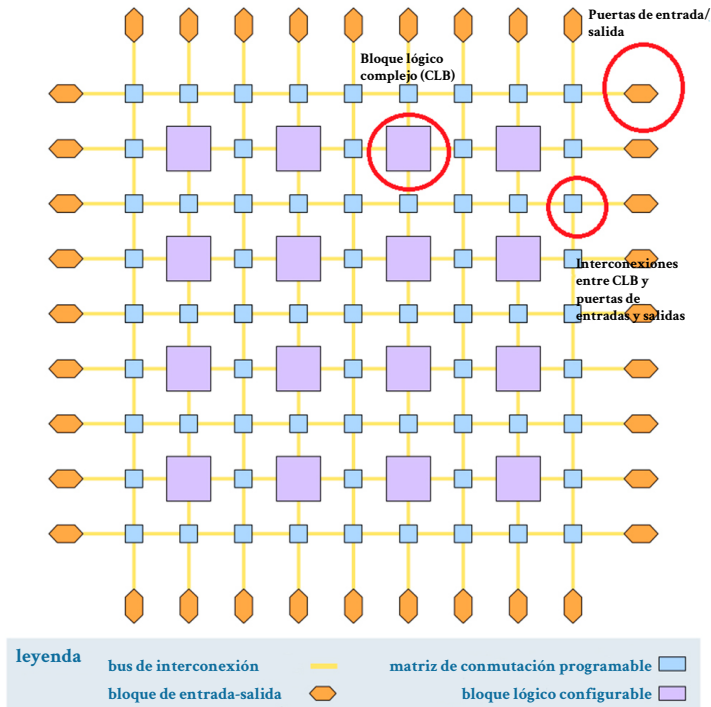
---

<sup>4</sup> Preferimos dejar este punto de lado en este trabajo por falta de extensión. Pero la idea principal de “*well-formedness*” para Floridi es que los datos corresponden a las reglas sintácticas que son manejadas por el sistema, código o lenguaje elegido. Se vincula con el contenido semántico que hemos descrito.

- b) Signo es un datum en el cual el receptor es diseñado en responder sistemáticamente.
- c) Una Señal es un signo que ha sido enviado por un emisor diseñado en emitirlo como un signo para un tipo de receptor particular.
- d) Un símbolo es una señal que es parte del sistema auto-referencial y gobernado por reglas. (2018, p. 16)

Retomando el ejemplo de la puerta AND, un *datum* es una corriente eléctrica que solo puede ser calificada como tal para esta puerta que es este receptor en particular. Si en vez de corriente tenemos agua o rayos ultra violetas, estos, no serán recibidos, el receptor en este sentido debe ser capaz de hacer la diferencia (de forma negativa). En caso del *signo*, la puerta debe responder sistemáticamente a ciertos valores (entre 0.1 V y 0.5 V –bajo–, entre 1.1 V y 1.5 V –alto–, y entre 0.6 V y 1 V –indeterminado–) que son *signos*. La *señal* es el ordenamiento espacio-temporal del output de la puerta AND con otras señales proviniendo de las otras puertas. Este ordenamiento es crucial ya que puede implicar comportamientos o computaciones distintas y es en esta etapa superior que se ejerce la información instruccional (desarrollada a partir de las características citadas aquí). Existe en esta etapa una distinción funcional, dicen Fresco y Wolf, entre los *inputs* operacionales (*outputs* de la puerta) y el *input* de control (información instruccional). Finalmente, los *símbolos*, enfatizan la referencia inter-simbólica dentro de un sistema particular. Los símbolos pueden, en caso de la lógica formal, referirse a otros símbolos sin hablar de cosas que existen “en el mundo” y por lo tanto pueden ser considerados como sistemas “cerrados”. De esta manera, Fresco y Wolf plantean la posibilidad de construir máquinas abstractas de procesamiento de la información simbólica. Por ejemplo, dicen Fresco y Wolf (2018, p. 20), un programa es un conjunto de símbolos y el lenguaje de programación es el sistema auto-referencial que definen los *inputs* y operaciones permitidas en el programa. El código que no ha sido compilado puede, según los autores, manifestarse físicamente como una señal.

La estrategia de Fresco y Wolf es, entonces, usar la descomposición jerárquica de Floridi y el aporte teleológico del usuario/receptor de Corning para el manejo de los datos/signos/señales y símbolos. Para Corning (2007, p. 302) el ambiente contiene información de control latente o potencial pero que este potencial no difiere de sus propiedades físicas. En este sentido, este potencial se actualiza solamente cuando un sistema



**Fig. 3:** Representación de un chip FPGA de lógica programada (Ventura, 2014). Los círculos (nuestros) representan los niveles donde actúa la programación en un FPGA.

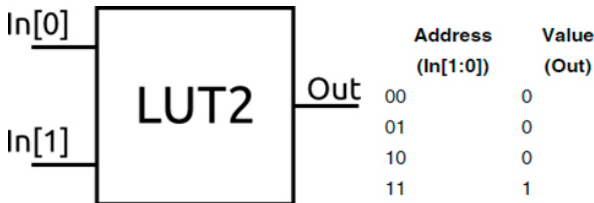
*Nota:* Imagen adaptada de “¿Qué es la lógica programada?” de V. Ventura, <https://polaridad.es/logica-programada-programable-que-es-pld-fpga-hdl-cpld/>

propositivo (cibernético) lo utiliza, por lo que es contexto-dependiente y específico al usuario. La idea de fondo de esta estrategia es interesante para tratar el problema de si hay que diferenciar los datos de una noción semántica o no, pero esta no se encuentra libre de problemas. La lógica programada puede representar un contraejemplo del ejemplo de puerta lógica usado por Fresco y Wolf. En efecto, para algunos aparatos más pequeños se hace uso de un tipo de diseño implementado en chips que reconfigura los circuitos a partir de modificaciones en un programa o a partir de los componentes.

Estos se diferencian de los circuitos clásicos permitiendo cambiar y modificar desde los componentes (*hardware*) o desde un programa (en una memoria de estado sólido) la configuración lógica de los circuitos.



Si tomamos un FPGA (un tipo de chip de lógica programada, ver figura 3) podemos encontrar por lo menos tres problemas. Primero, para Fresco y Wolf el argumento para evitar el *pancomputacionalismo* era que la puerta lógica AND podía solamente procesar un único tipo de *input* por ejemplo el voltaje *bajo* y *alto*. En la concepción de la lógica programada esto no es así, la misma puerta AND puede ser reconfigurada y procesar otros valores que no necesariamente sean voltajes sino cualquier elemento expresado mediante una memoria RAM a nivel simbólico. Segundo, la jerarquización de *datum*, *signo*, *señal* y *símbolo* puede verse modificada por el rol que puede tener la programación del circuito a partir de un programa. En la figura 4 un LUT2 de un FPGA se compone de un *input* de memoria RAM y de un *output*. El *output* posee los valores binarios 0 y 1 que forman una tabla lógica, el LUT2, al emplear dos variables, permite computar todas las tablas posibles dado que se puede programar el *output* deseado. En este sentido, la información instruccional no actúa solo en el ordenamiento de las señales, sino que configura y determina qué señales computar. Por lo tanto, lo que determina la computación es cómo hemos programado el LUT y lo que se considera válido como signo. Además, existen por lo menos dos instancias de ordenamientos posteriores ejercidas por la programación: a) se programa las interconexiones o el ruteo que permite implementar funciones (Fig. 3); y b) los bloques de *inputs* y *outputs* del FPGA son también programables. Por lo tanto, tenemos un ordenamiento estructural de las señales en a) y b). Una vez realizado este ordenamiento, no es necesario el uso de un programa una vez obtenido el *output* del FPGA ya que la señal se encuentra ordenada y cumple la función deseada.

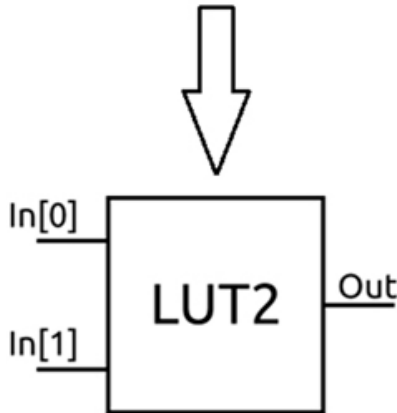


**Fig. 4:** A la izquierda un Look-Up Table (LUT) de dos entradas. A la derecha una tabla de verdades de un LUT2 en donde la columna de izquierda representa los valores ofrecidos por la RAM y a la derecha los valores de verdades binarios 0 y 1.

*Nota:* Imagen tomada de "How does an FPGA work?" de J. Rajewski, 2018, <https://alchitry.com/blogs/tutorials/how-does-an-fpga-work>

Una tercera dificultad, para Fresco y Wolf, es el tratamiento del “significado”, en este caso el significado se encuentra asociado con la computación física y la teleología de la información para la computación. Un *datum* o signo es significativo en tanto que la puerta lo reconoce como signo y permite producir una señal. A esto lo consideran como una semántica interna y no como una semántica externa. Pero ahora que la lógica programada puede determinar lo que consideramos como signo: ¿cuán interna es esta semántica? O mejor expresado: ¿cuán externa es? Si se define el *output* deseado a partir de la programación nos encontramos con dificultades para poder diferenciar ambas semánticas. El *output* del LUT2 es definido a partir de una semántica deseada por el programador para una función determinada. Esta semántica puede ser considerada como “interna” ya que es propia a la computación deseada en el *Bloque Lógico Complejo* (CLB) pero es externa por ser establecida desde el nivel simbólico (Fig. 5). Para Fresco y Wolf es este nivel simbólico que determina la externalidad de la semántica del sistema.

Ahora bien, la segunda crítica es la más problemática y responderla desde Fresco y Wolf es relativamente complejo. Una de las propuestas es que, en realidad, en un circuito no disponemos siempre de *data* y signos, sino que ambos son reemplazados por señales ya que una puerta procesa señales que pueden haber sido producidas por puertas anteriores (2018,



**Fig. 5:** La instrucción establece cuál puerta el LUT2 debe ser.

*Nota:* Imagen adaptada de “How does an FPGA work?” de J. Rajewski, 2018, <https://alchitry.com/blogs/tutorials/how-does-an-fpga-work>

p. 18). Pero si fuera este el caso, ya que poseemos señales en los *inputs*, significa que tenemos información instruccional que ya actuó en este nivel y la noción de dato se entremezcla con la de señal. La terminología es confusa y tenemos entonces dos posibles interpretaciones: (a) una señal como *datum* y signo en donde no actuaría la información instruccional hasta convertir el *output* en señal que luego es ordenado por la información instruccional (que no resuelve nuestra crítica); y (b) en la que la señal es entendida como *datum* y signo pero que esta vez sí actúa la información instruccional. Si (b) es cierta entonces tenemos dos problemas: i) no tiene sentido diferenciar *datum*, signos y señales ya que con señales sería suficiente; y ii) el nivel simbólico no sería independiente de la computación física como Fresco y Wolf desean ya que la información instruccional podría actuar en el nivel de la señal y por lo tanto tener una incidencia física.

La primera crítica parece solucionarse si aceptamos el aspecto propositivo y funcional de la computación, en este sentido solo sería aceptado como datos y signos lo que pueda convertirse en señales a través de la puerta lógica. Pero, como vimos con la crítica anterior, si el signo y el *datum* son considerados como señales donde la información instruccional puede actuar y que solo necesitaríamos señales para computar, entonces perdemos la fortaleza que permitía la definición del signo. La señal podría ser cualquier elemento que pueda convertirse en señales y daría lugar a un pancomputacionalismo. Pero por otro lado podemos si se mantiene la diferencia entre signo y señal disolver la primera crítica ya que distinguimos señal de signo.

## Conclusión

En este trabajo hemos analizado principalmente las propuestas de Floridi y de Fresco y Wolf respecto de la cuestión de la conformación del concepto de dato e implícitamente de información en computación. Hemos descriptos las críticas realizadas por Fresco y Wolf acerca de Floridi, y también hemos planteado el problema conceptual que reviste hablar y usar una noción de dato “puro”. Luego hemos presentado la alternativa ofrecida por Fresco y Wolf basada en la postura funcional y teleológica de Corning. Hemos presentados las críticas realizadas por Fresco y Wolf y su adaptación de la teoría de Corning y Floridi. Finalmente planteamos tres críticas a partir de la lógica programada hacia la postura de Fresco y Wolf en el que tenemos algunas dificultades para poder mantener la

jerarquía propuesta. (a) Respecto de dónde y cómo se ejecuta la información instruccional en el nivel de la señal, (b) Que no consideramos que el argumento en contra del *pancomputacionalismo* puede aplicarse en la lógica programada, (c) Que su postura no puede evitar realmente un tipo de semántica externalista como ellos creen.

## Referencias

- Adriaans, P., & Benthem, J. F. A. K. van (Eds.). (2008). *Philosophy of information*. Amsterdam: Elsevier.
- Ashby, W. R. (1960). *Design for a brain: The origin of adaptive behavior* (2<sup>nd</sup> ed.). Chapman & Hall.
- Corning, P. (2007). Control information theory: The “missing link” in the science of cybernetics. *Systems Research and Behavioral Science*, 24, 297-311.
- Floridi, L. (2010). *Information: A very short introduction*. Oxford: Oxford University Press.
- Floridi, L. (2011). *The philosophy of information*. Oxford: Oxford University Press.
- Fresco, N; Wolf, M. (2014). The instructional information processing account of digital computation. *Synthese*, 191, 1469-1492. doi: 10.1007/s11229-013-0338-5
- Fresco, N; Wolf, M. (2018). Data, signals and information in digital computation. *Logique & Analyse*, 241, 1-23.
- Rajewski, J. (2018). Ilustración del Lut2 (Figura 4). Recuperado de <https://alchitry.com/blogs/tutorials/how-does-an-fpga-work>
- Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27, 379-423.
- Simon, H. A., (1947). *Administrative behavior: A study of decision-making processes in administrative organization*. Macmillan Co.
- Ventura, V. (2014). Ilustración del circuito FPGA (Figura 3). Recuperado de <https://polaridad.es/logica-programada-programable-ques-pld-fpga-hdl-cpld/>

Wiener, N. (1948). *Cybernetics: Or control and communication in the animal and the machine*. Cambridge, MA: MIT Press.