

# No hay software

*The Eastern World is exploding*, cantaba Barry McGuire. Lo hizo por primera vez en los salvajes 60 para disuadir a sus amigos, entre vinilo y grabadores de cassettes, que viviáramos en vísperas de la destrucción. La segunda, después de una brillante reedición electrónica que llevó su viejo canto al vinilo a la punta de la lista de AFN Dharan, para convencer a los guerreros occidentales de Tormenta del Desierto a través de onda corta que ellos o nosotros vivimos en vísperas de la destrucción!...

McGuire (o más bien la procesadora digital de señales que podía cancelar sin dejar rastros una negación perpetuada fonográficamente) tuvo razón, pero solamente porque las explosiones no cuentan. El hecho que exploten torres de petróleo o cohetes Scud, nietos lejanos de la V2 del Reich, no tiene importancia. El este puede explotar tranquilamente, porque lo único que cuenta es lo que está ocurriendo en el mundo occidental: una implosión, en primer lugar, de las tecnologías de avanzada y como consecuencia de ello, también de un conjunto de significantes que de otro modo quizá aún podría llamarse espíritu del siglo. Sin técnica informática no hay deconstrucción, decía Derrida en Siegen. Los escritos y los textos (incluyendo el texto que acabo de leer), ya no existen en tiempos y espacios perceptibles, sino en las celdas de los transistores de las computadoras. Y dado que en las tres últimas décadas las hazañas de Silicon Valley han logrado reducir el tamaño de los transistores a dimensiones submicrónicas, es decir, de menos de un micrón, nuestra escena literaria actual se puede describir ahora en términos de geometría fractal: como autosemejanza de las letras a través de unas seis décadas, desde la propaganda alta como una casa hasta el mismo mapa de bits del tamaño de un transistor. En el comienzo alfabético de la historia mediaban, entre un camello y un gamel, su letra hebrea, justamente dos décadas y media; en cambio, con la miniaturización de todos los signos a medidas moleculares, ha desaparecido el propio acto de escribir.

Como sabemos todos, pero no lo decimos, ya nadie escribe. Al parecer, la literatura, ese tipo particular de software, quizá esté tratando de salir de su confusión insanable entre uso y alusión. Hasta en los tiempos de los himnos de Hölderlin la sola alusión de por ejemplo un relámpago, parece haber sido aún evidencia suficiente para un posible uso poético.<sup>2</sup> En cambio, hoy en día, después de la transformación de este relámpago en electricidad, la escritura humana consiste en inscripciones que no solamente se graban en silicio por medio de litografía electrónica; sino que a diferencia de todas las herramientas de escritura de la historia también están en condiciones de leer y escribir por sí mismas.

Por lo tanto, el último acto de escribir de la historia puede haber sido, cuando a fines de la década del 70, un equipo de ingenieros de Intel bajo la dirección del Dr. Marcian E. Hoff extendió algunas docenas de metros

<sup>1</sup> Agradezco a Wolfgang Hagen/Radio Bremen, quien ofreció a sus oyentes la comparación entre las dos versiones de *Eve of Destruction*

<sup>2</sup> Véase, a la brevedad, Thomas Hafki, *Elektrizität und Literatur. 1750-1816*.

cuadrados<sup>3</sup> de papel de dibujo en el piso de un garaje desocupado de Santa Claras para dibujar la arquitectura del hardware de su primera microprocesadora integrada. Este dispositivo manual de 2000 transistores y sus canales de comunicación fue reducido en un segundo paso, ahora mecánico, al tamaño de la uña de un pulgar del chip real y en un tercer paso, fue escrito en silicio por medio de aparatos electro-ópticos. Después que el producto final, el.4004, el prototipo de todas las microprocesadoras existentes hasta ese momento, había ocupado su lugar entre las nuevas computadoras personales del comitente japonés de Intel, en un cuarto paso, empieza nuestra literatura posmoderna.

Con la actual complejidad del hardware de las microprocesadoras, las técnicas de diseño manuales ya no tienen ninguna posibilidad de competir. Ya no hay papel de dibujo que sirva a los ingenieros para desarrollar la siguiente generación de computadoras, solamente pueden usar el diseño con apoyo de computadoras (Computer Aided Design). Las capacidades geométricas de la respectiva última generación de computadoras alcanzan apenas para diseñar la topología de la generación siguiente. Así es que los pies de quienes te llevarán fuera, nuevamente quedan delante de la puerta.

Sin embargo, con sus primitivas copias al carbónico, Marcian E. Hoff ya había dado el ejemplo casi perfecto de una máquina de Turing. Desde la tesis de Turing, de 1937, cualquier operación aritmética, ya sea realizada por humanos o máquinas, puede ser formalizada como una cantidad contabilizable de órdenes que operan sobre una banda de papel infinitamente larga y sus signos discretos. El bosquejo de Turing, de una tal máquina de papel,<sup>4</sup> cuyas operaciones sólo incluían escribir y leer, adelantar y retroceder, resultó el equivalente matemático de todas las funciones calculables y se encargó de que la inocente denominación profesional de computadora fuera desplazada completamente por el sentido mecánico de la palabra.<sup>5</sup> Las Máquinas Universales de Turing sólo necesitan ser alimentadas con la descripción (el programa) de cualquier otra máquina para imitar efectivamente a esta máquina. Como es la primera vez, desde Turing, que se puede hacer abstracción de las diferencias de hardware entre ambas máquinas, la forma más estricta, específicamente física, de la denominada Hipótesis de Church-Turing apunta a declarar que la propia naturaleza es una Máquina Universal de Turing.



Esta afirmación, como tal, hizo que la implosión del hardware duplicara la explosión del software. Desde que las computadoras pueden ser implementadas, desde 1943 sobre la base de válvulas y a partir de 1949 sobre la base de transistores, existe también el problema de describir y leer de alguna manera las máquinas de escribir-leer

<sup>3</sup> 1978, dicen que cuando se diseñó la procesadora Intel 8086, estas copias al carbónico habrían cubierto 64 m<sup>2</sup> de papel milimetrado. Véase, Klaus Schrödl, 1990, *Quantensprung. DOS, Fascículo 12, pag. 102 f.*

<sup>4</sup> Véase, Alan M. Turing. 1937, *Über berechenbare Zahlen. Mit einer Anwendung auf das Entscheidungsproblem. en: Turing, .1987, pag. 40 s.*

<sup>5</sup> El progreso de este desplazamiento es demostrado por los autores y linotipistas de los *Programmer's Reference Manuals* de Intel. Después de su traducción al inglés usual, a partir de la orden de la coma flotante *f2xml*, es decir, la 2. potencia de una magnitud de entrada menos 1, no se obtiene por ejemplo "Computadora 2<sup>x</sup>-I«, sino »Computadora 2<sup>x</sup>-II«. Véase, Intel Corporation, 1989, *387DX. User's Manual. Programmer's Reference*. Santa Clara/CA., pag. 4-9. Asimismo, Intel Corporation, 1990, *1486 Microprocessor. Programmer's Reference Manual*. Santa Clara/CA., pag. 26 – 72.

universales que en sí mismas son ilegibles. Como se sabe, la solución se llama software, es decir, el desarrollo de lenguajes de programación superiores. El antiquísimo monopolio de los lenguajes cotidianos, de ser su propio metalenguaje y por lo tanto, no tener ya no tener el monopolio del otro, se ha derrumbado y ha dado lugar a una nueva jerarquía de lenguajes de programación. Entre tanto, esta Torre de Babel posmoderna<sup>6</sup> va desde simples códigos de ordenes, cuya extensión lingüística todavía es una configuración de hardware, pasando por *Assembler*, cuya extensión está constituida justamente por aquellos códigos de órdenes, hasta los denominados lenguajes superiores, cuya extensión también se llama –después de toda clase de rodeos– pasando por *Interpreter*, *Compiler* y *Linker*, también *Assembler*. Por lo tanto, en la actualidad, escribir es –también como desarrollo de software– una cadena casi interminable de autosemejanzas. Tal como la descubrió la geometría fractal. Sólo que, a diferencia del modelo matemático, sigue siendo un imposible físico-fisiológico seguir alcanzando todas estas capas. Las modernas tecnologías mediáticas ya apuntan básicamente, desde el cine y el gramófono, a obviar las percepciones sensoriales. Ya no podemos saber lo que hace nuestro escribir y al programar, menos aún.

Para ilustrar esta situación bastan casos de la vida diaria, como por ejemplo el programa de procesamiento de textos del que proceden mis palabras. El *genius loci* de Palo Alto que produjo los primeros programas operativos y también los más elegantes, tendrá que disculpar que un súbdito de la Microsoft Corporation limite sus ejemplos a los más tontos de todos.

Para procesar textos, es decir, para convertirlos con Microsoft DOS en máquina de papel en un IBM AT se requiere la compra previa de un paquete de software comercial. En segundo lugar, algunos archivos del paquete deben tener las extensiones EXE o COM, ya que de lo contrario nunca podría iniciarse un procesamiento de textos con DOS. Los archivos ejecutables y sólo ellos mantienen una relación peculiar con sus propios nombres. Por un lado, tienen nombres autoreferenciales como WordPerfect, por otro, un nombre más o menos críptico, tal como el acrónimo sin vocales WP. Claro está, que el nombre completo sólo

<sup>6</sup> Véase, Wolfgang Hagen, 1989, *Die verlorene Schrift. Skizzen zu einer Theorie der Computer*. En: *Arsenale der Seele. Literatur- und Medienanalyse seit 1870*. Ed. Friedrich A. Kittler y Georg Christoph Tholen, Munich, pag. 221: “En la estructura del lenguaje de la lógica de las máquinas de Neumann ya se advierte la fundamentación de la escisión del principio de software y documentación de software y es así que desde 1945 se forma una torre babilónica de programas de computación, cuya utilización ya no tiene nada que ver con la sensata instauración de un lenguaje mecánico, una torre de software con errores no documentados, dialectos insanablemente confusos y una acumulación de actos de lenguaje que ya nadie puede reproducir”. [„In der Sprachstruktur der Neumannschen Maschinenlogik liegt also schon der prinzipielle Auseinanderfall von Software und Software-Dokumentation begründet, und so türmt sich seit 1945 ein babylonischer Programm-Turm von Computer-performances auf, deren Benutzung mit der sinnvollen Veranstaltung einer Maschinensprache nichts mehr zu tun hat. Ein Software-Turm mit undokumentierten Fehlern, heillos verworrenen Dialekten und einer Anhäufung von sprachlichen Akten, die niemand mehr nachvollziehen kann.”]

En una descripción menos precisa, pero desesperada, un experto de UNIX expresa: „Después de algún tiempo de uso, casi todos los grandes sistemas operativos se caracterizan por un alto grado de „contaminación“. Proliferan en todas las direcciones y dan la impresión que son ruinas que apenas si se pueden mantener en pie. [»Fast alle großen Betriebssysteme zeichnen sich nach einem gewissen Alter durch einen hohen „Verschmutzungsgrad“ aus. Sie wuchern in alle Richtungen und machen den Eindruck, als seien sie Trümmer; die nur noch mühsam zusammengehalten werden.»] (Horst Drees, 1988, “UNIX. Ein umfassendes Kompendium für Anwender und Systemspezialisten. Haar, pag. 19)

El experto de UNIX omitió delicadamente el nombre de una empresa como Microsoft cuando se refería a las proliferaciones.

sigue sirviendo necesariamente a las estrategias publicitarias de los productores de software que aún usan el lenguaje cotidiano., lo que se explica porque el Disk Operating System alias DOS no podría leer nombres de archivos con más de ocho letras. De allí que las abreviaturas o acrónimos impronunciables, sin vocales, esta anulación de una innovación griega elemental, no solamente es necesaria para la escritura posmoderna, sino también completamente suficiente. Más aún, parecen haber conferido nuevamente poderes mágicos al alfabeto, por primera vez después de su invención. La abreviatura WP efectivamente hace lo que dice. A diferencia, no solamente de la palabra WordPerfect, sino también de antiguas palabras europeas vacuas como espíritu (*Geist*) o palabra (*Wort*), los archivos de computación ejecutables comprenden todas las rutinas y datos que son necesarios para su realización. Si bien el acto de escribir, de apretar las teclas W, P y enter no completa la palabra, pone en marcha una operación efectiva de WordPerfect. Tales son los triunfos que otorga el software.

A continuación, para no quedar a la zaga de la barra de herramientas, el *paperware* anexo –más o menos inflacionario– todavía duplica la magia. Como tienen que tender un puente por encima del abismo entre lenguajes formales y cotidianos, electrónica y literatura, los manuales de software comunes presentan su paquete programático como si fueran agentes lingüísticos, que imponen su poder omnimodo sobre recursos del sistema, espacios de direcciones y parámetros de hardware de la respectiva computadora: WP, reclamado desde la celda de mando con el argumento X, conectaría el monitor del modo A al B, empezaría en C y finalmente volvería a D, etc.<sup>7</sup>

Pero sucede que todas las operaciones que lleva a cabo el agente WP mediante el Paperware son completamente virtuales, pues cada operación individual debe llevarse a cabo “de acuerdo con” DOS. De hecho, solo trabaja el sistema operativo y le sigue su shell: [COMMAND.COM](#) revisa el amortiguador de teclado buscando un nombre de archivo de 8 bytes, traduce las direcciones relativas de un archivo que haya encontrado a direcciones absolutas, descarga esta versión modificada de la memoria masiva externa a la RAM de silicio y entrega finalmente la ejecución temporaria del programa a las primeras líneas de código de un esclavo llamado WordPerfect.

Sin embargo, el mismo argumento también se puede aducir en contra de DOS, pues en definitiva, el sistema operativo opera simplemente como ampliación de un sistema básico input/output llamado BIOS. Ningún programa de aplicación, ni siquiera el sistema microprocesador básico podría llegar a arrancar, si algunas funciones elementales –que por razones de seguridad están grabadas en silicio y por lo tanto forman parte del hardware imborrable– no dispusieran de la facultad del barón Münchhausen de rescatarse a sí mismas del pantano tirando de su propio cabello.<sup>8</sup>

<sup>7</sup> El único ejemplo contrario que conozco procede –no por casualidad– de la Free Software Foundation de Richard Stallman, la que pronosticó al software Copyright una lucha tan heroica como desesperada. El ejemplo contrario sería: »When we say that >C-n moves down vertically one line we are glossing over a distinction that is irrelevant in ordinary use but is vital in understanding how to customize Emacs. It is the function *next-line* that is programmed to move down vertically. C-n has this effect because it is bound to that function. If you rebound C-n to the function *forward-word* then C -n will move forward by words instead.« (Richard Stallman, 1988, GNU Emacs Manual. 6. Ed. Cambridge/MA., pag. 19.)

<sup>8</sup> Esto puede pasar por una traducción libre de Booting.

Cualquier transformación material de entropía en información, de un millón de transistores dormidos en diferencias de tensión eléctricas supone necesariamente un hecho material que se llama *reset*.

En principio, esta degradación de software a hardware, de planos de observación más elevados a otros más bajos, puede producirse durante cualquier cantidad de décadas. A pesar de sus capacidades metafóricas como por ejemplo *call* o *return*, hasta las *Code-Operations* elementales se reducen a manipulaciones de signos absolutamente locales y con ello, según Lacan, a significantes de potenciales eléctricos. Toda formalización en el sentido que da Hilbert a la palabra, tiene el efecto de suprimir la teoría, simplemente porque „la teoría es un sistema de afirmaciones que ya no son significativas, sino un sistema de oraciones que son una sucesión de palabras, cuya sucesión de palabras es, a su vez, una sucesión de letras. Por eso, solamente se puede distinguir por la forma, qué combinaciones de palabras son oraciones, qué oraciones son axiomas y qué oraciones resultan de otras como consecuencia directa de aquellas.” [*die Theorie ein System nicht mehr bedeutsamer Aussagen ist, sondern ein System von Sätzen als Wortfolgen, welche Wortfolgen ihrerseits Buchstabenfolgen sind. Deshalb kann man allein aufgrund der Form unterscheiden, welche Wörterkombinationen Sätze sind, welche Sätze Axiome und welche Sätze als unmittelbare Folgen aus anderen hervorgehen.*]<sup>9</sup>

Cuando el significado se reduce a oraciones, las oraciones a palabras, las palabras a letras, tampoco hay software. Más bien, no lo habría si los sistemas de computación no hubieran tenido que coexistir hasta ahora en un ambiente de lenguajes cotidianos. Sin embargo, este ambiente está formado desde una famosa invención, doblemente griega,<sup>10</sup> por letras y monedas, *letters and litters*. Entre tanto, estas buenas razones económicas han erradicado prolijamente la humildad de Alan Turing, quien en los albores de la era de la computación prefería leer impresiones de máquinas binarias y no decimales.<sup>11</sup> Contrariamente a ello, la denominada filosofía de la denominada comunidad computarizada hace todo lo posible para esconder hardware detrás de software, significantes electrónicos detrás de interfaces hombre-máquina. Con altruismo, los manuales de programación para lenguajes superiores advierten acerca del desorden espiritual que se produciría en los *assembler* al escribir funciones trigonométricas.<sup>12</sup> Amablemente, los procesos de BIOS (y sus dispositivos automáticos especializados) se hacen cargo de la función de “ocultar de su programa, los detalles del hardware en que se basa el control” [*die Einzelheiten der Steuerung zugrunde liegender Hardware vor Ihrem Programm zu verstecken*]<sup>13</sup>. Por lo tanto, no muy diferente que en el gradualismo de las jerarquías de los ángeles medievales, las funciones operativas como COMMAND.COM esconderían el BIOS, los programas de aplicación como WordPerfect, al sistema operativo, etc., hasta que finalmente, en los últimos años, dos cambios fundamentales en el diseño de las computadoras (o en el esquema científico del Pentágono) condujeron al cierre exitoso de este sistema secreto.

Primeramente se desarrollaron interfaces gráficas útiles en un plano deliberadamente superficial que, como ocultaban los actos de escribir todavía ineludibles para la programación,

<sup>9</sup> Stephen C. Kleene, citado en: Robert Rosen, 1988, *Effective Processes and Natural Law*. En: Herken, 1988, pag. 527.

<sup>10</sup> Véase, Johannes Lohmann, 1980, *Die Geburt der Tragödie aus dem Geiste der Musik*. Archiv für Musikwissenschaft, 37, pag. 174.

<sup>11</sup> Véase, Andrew Hodges, 1983, *Alan Turing: the enigma*. New York, pag. 399. -

<sup>12</sup> Véase, TOOL Praxis, 1989, *Assembler Programmierung auf dem PC*, Ed. 1. Würzburg pag. 9.

<sup>13</sup> Nabajyoti Barkalati. 1989, *The Waite Group's Macroassembler Bible*. Indiana/IL., pag. 528.

privan a sus usuarios de toda una máquina. Ni siquiera el compendio de gráfica computada autorizado por IBM asegura que las interfaces de usuario gráficas computarizadas acelerarían o harían más eficientes la programación de sistemas que las sencillas barras de herramientas.<sup>14</sup> Luego apareció, en relación directa con ADA,<sup>15</sup> el lenguaje de programación Pentagon, pero en el plano microscópico del propio hardware, un nuevo modo de funcionamiento de la procesadora que se llamó *Protected Mode*, que según el *Microprocessor Programming Manual* de Intel tenía el único objeto de disuadir a «untrusted program» y «untrusted users» de cualquier acceso a recursos de sistema como canales de entrada/salida o al núcleo del sistema operativa. Sin embargo, en este sentido técnico son dignos de confianza los usuarios que en *Protected Mode* (como el que reina por ejemplo en UNIX) ya no tienen permiso para manejar sus máquinas.

Esta cruzada victoriosa incontenible del software es una singular inversión de lo que demostrara Turing: que no pueden presentarse problemas que se pueden calcular matemáticamente que no puedan ser resueltos por una simple máquina. En el lugar exacto de esta máquina, la hipótesis física de Church-Turing – que equiparó el hardware físico con los algoritmos para su cálculo – creó un hueco, que pudo ocupar exitosamente el software, no sin aprovechar su oscuridad.

En definitiva, los lenguajes superiores de programación, cuanto más alta y en lenguaje más

<sup>14</sup> Véase, James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, 1990, *Computer Graphics. Principles and Practice. 2. Ed. Reading/MA.* pag. 397 f.: «Direct [graphic] manipulation is sometimes presented as being the best user-interface style. It is certainly quite powerful and is especially easy to learn. But the Macintosh interface can be slow for experimented users in that they are forced to use direct manipulation when another style would generally be faster. Pointing the file )Chapter 9< with direct manipulation requires the visual representation of the file to be found and selected, then the Print command is involved. Finding the file icon might evolve scrolling through a large collection of items. If the user knows the name of the File, typing >Print Chapter 9< is faster. Similarly, deleting all files of type >txt< requires finding and selecting each such File and dragging it to a trash can. Much faster is the UNIX-style command >rm 'txt<, which uses the wildcard \* to find all Files whose names end in >.txt<.» De todo esto surge finalmente la siguiente exageración desconsoladora referida al propio fin de las computadoras: «Some applications, such as programming, to not lend themselves to direct manipulation.»

<sup>15</sup> Sobre la relación entre Pentagon, ADA e Intel iAOX 432, la primera microprocesadora con Protected Mode, de cuyo fracaso económico nació luego el avance industrial de la 80286 a la 80486 véase Glenford Myren jr., 1982, *Overview of the Intel iAPX 432 Microprocessor*. En: *Advances in Computer Architecture*. New York, S. 335-344. (Agradezco el dato a Ingo Ruhmann/Bonn.). Quien quiera saber más sobre este fracaso debe meditar largamente sobre la frase: «The 432 can be characterized as a three-address-storage-to-storage architecture, there are no registers visible to programs.» (pag. 342). [*Über den Zusammenhang zwischen Pentagon, ADA und Intels iAOX 432, dem ersten Mikroprozessor im Protected Mode, dessen ökonomischem Scheitern dann der Industriestandard vom 80286 bis zum 80486 entsprang, vgl. Glenford Myren jr., 1982, Overview of the Intel iAPX 432 Microprocessor. In: Advances in Computer Architecture. New York, S. 335-344. (Mit Dank an Ingo Ruhmann/Bonn.) Wer dem Scheitern näherkommen will, meditiere eine Assemblersitzung fang über den Satz: »The 432 can be characterized as a three-address-storage-to-storage architecture, there are no registers visible to programs.« (pag. 342)]*

cotidiano crezca su torre babilónica, trabajan –en su forma estándar- igual que las denominadas funciones de un solo uso de la primera criptografía matemática.<sup>16</sup> En su forma estándar, tales funciones se pueden calcular en un tiempo razonable, por ejemplo, cuando el tiempo que necesita la máquina sólo se incrementa en expresiones polinómicas de la complejidad de la función. En cambio, el tiempo necesario para la forma inversa, es decir, para retrotraer el resultado de la función a sus parámetros iniciales se incrementaría en relación exponencial y por lo tanto, inadmisiblemente, con respecto a la complejidad de la función. En otras palabras, las funciones de un solo uso protegen los algoritmos de su propio resultado.

Al software, esta propiedad criptográfica, le viene como anillo al dedo. Brinda un camino cómodo para eludir el contenido, es decir que, según lo demostrara Turing, el concepto de propiedad intelectual se ha convertido en imposible y con algoritmos, en más imposible aún. Esta propiedad criptográfica no hace más que insistir en el hecho que el software, como cualidad independiente de la máquina, no exista. Todas las licencias, *dongles* y patentes presentadas tanto para WP como para WordPerfect, demuestran la funcionalidad de las funciones de un solo uso. Hace poco, tribunales americanos han confirmado – a despecho de cualquier honor matemático- derechos de Copyright para algoritmos.

De allí que no llame la atención que recientemente la más alta jerarquía, específicamente IBM, haya iniciado la caza de fórmulas matemáticas que podrían evaluar algoritmos por su originalidad más allá de la medida de la información de Shannon y el contenido de la información de Kolmogorow. La IBM echa en cara a Shannon justamente el hecho que en los buenos tiempos de la teoría de la información coincidiera más o menos, la máxima información con el máximo ruido, es decir que las series de lanzamiento de dados eran premiadas matemáticamente. Pero tampoco la medida de Kolmogorow, que premia el algoritmo más corto entre todos los algoritmos posibles para obtener un output constante, los satisface. Según Kolmogorow todo el esfuerzo para lograr un sistema de tablas trigonométricas o astronómicas se perdería en las simples ecuaciones en que se basan dichas tablas. Por lo tanto, la nueva medida-IBM tiene profundidad lógica al confundir asentaderas con ingenio, como se define a continuación:

“El valor de una noticia [...] no parece residir ni en su información (sus partes absolutamente imprevisibles), ni en su evidente redundancia (repetición textual, frecuencias de bits desiguales), sino más bien en algo que se podría llamar redundancia soterrada, en partes que solamente son difícilmente previsibles y hechos que el receptor –en principio- podría haber deducido por sí mismo sin que se las digan, aunque sólo con considerable gasto de dinero, tiempo o trabajo de cálculo. En otras palabras, el valor de una noticia es la cantidad de trabajo matemático u otro que ha realizado plausiblemente su emisor y que su receptor no debe realizar nuevamente.” [“Der Wert einer Nachricht [...] scheint weder in ihrer Information (ihren absolut unvorhersagbaren Teilen) noch in ihrer offensichtlichen Redundanz (wörtlichen Wiederholungen, ungleichen Bitfrequenzen) zu liegen, sondern vielmehr in etwas, was begrabene Redundanz heißen könnte - in Teilen, die nur mit Schwierigkeiten vorhersagbar sind, und Sachverhalten, die der Empfänger im Prinzip hätte herausfinden können, ohne sie gesagt zu bekommen, aber nur mit beträchtlichem Aufwand an Geld, Zeit oder Rechenleistung. Mit anderen Worten, der Wert einer Nachricht ist der Betrag an mathematischer oder andersartiger Arbeit, den ihr Sender plausiblerweise

<sup>16</sup> Para lo siguiente, véase Patrick Horster, 1982/1985, *Kryptologie: eine Anwendung der Zahlentheorie und Komplexitätstheorie*. Mannheim-Viena -Zürich, pag.23-27.

*aufgebracht und den ihr Empfänger nicht noch einmal zu vollbringen hat.*]<sup>17</sup>

El rigor matemático, la medida IBM de profundidad lógica, podría sustituir así todos los antiguos conceptos en lenguaje cotidiano, necesariamente inexactos de originalidad, autoría y Copyright, es decir, también ponerlos en duda en cuanto a su procedimiento. Sólo que justamente el algoritmo destinado al cálculo de la originalidad de algoritmos es, en sí mismo, incalculable según Turing.<sup>18</sup>

En esta trágica situación, el derecho penal –al menos en Alemania- ha abandonado el concepto de propiedad intelectual de un software que es igualmente inmaterial, y en su lugar definió software como “cosa”. La comprobación del Tribunal Federal Supremo, de que ningún programa de computación podría funcionar jamás sin la correspondiente carga eléctrica en circuitos de silicio,<sup>19</sup> demuestra una vez más, que la imposibilidad virtual de distinguir entre software y hardware de ninguna manera se debe solamente -como les gustaría creer a los teóricos de sistemas- a un cambio del ángulo de observación.<sup>20</sup> Hay buenos motivos para apoyar la absoluta necesidad y por lo tanto la primacía del hardware.

Sin embargo, sólo una vez hubo máquinas con recursos de tiempo y espacio ilimitados, con infinita provisión de papel y con ilimitada velocidad de cálculo y esto fue en la comunicación de Turing titulada “Sobre los números calculables con una aplicación al problema decisivo” [*Über berechenbare Zahlen mit einer Anwendung auf das Entscheidungsproblem.*] Sin embargo, estos parámetros imponen límites estrictos en el propio código a todas las máquinas que se puedan hacer físicamente. La incapacidad de Microsoft DOS para reconocer archivos de más de ocho letras como por ejemplo, WordPerfect, pone en evidencia de manera trivial y obsoleta, no solamente un problema que ha producido incompatibilidades cada vez mayores entre las diferentes generaciones de microprocesadoras de 8, 16 y 32 bits. También indica, en principio, una imposibilidad de la digitalización, es decir

<sup>17</sup> Charles H. Bennett, 1988, *Logical Depth and Physical Complexity*. En: Herken, 1988, pag. 230.

<sup>18</sup> Nuestro agradecimiento a Oswald Wiener/Dawson City.

<sup>19</sup> Véase, M. Michael König, 1991, *Sachlich sehen. Probleme bei der Überlassung von Software*. c't, fascículo 3, pag. 73.

<sup>20</sup> Más bien se podría „suponer“ como dice en una carta de Dirk Baecker del 15-4-1991, “que la diferenciación entre hardware y software es una diferenciación que debe cuidar la vuelta de la diferenciación entre la posibilidad y la imposibilidad de programación en el área de lo programable. Avala igualmente la calculabilidad de la técnica y, en ese sentido, la propia técnica. Solamente puede avalarla porque la „unidad“ del programa solamente puede realizarse cuando la ecuación y el cálculo se reparten en respectivos dos lados de manera tal que siempre haya un solo lado operativo a disposición y el otro lado se pueda mantener constante. [“Eher könnte man, wie in einem Brief Dirk Baeckers vom 15.4.1991, vermuten, daß die Unterscheidung - zwischen Hardware und Software eine Unterscheidung ist, die den Wiedereintritt der Unterscheidung zwischen Programmierbarkeit und Nichtprogrammierbarkeit in den Bereich des Programmierbaren zu betreuen hat. Sie steht gleichsam für die Berechenbarkeit der Technik, und in dem Sinne für die Technik selbst. Sie kann dafür nur stehen, weil die Einheit, des Programms nur realisiert werden kann, wenn Gleichung und Berechnung jeweils so auf zwei Seiten verteilt werden, daß immer nur eine Seite operativ zur Disposition steht und die andere Seite konstant gehalten werden kann.”]

de calcular el cuerpo de los números ciertos, por lo tanto de lo que antes solía llamarse naturaleza.<sup>21</sup>

Esto quiere decir, como lo expresa el Los Alamos National Laboratory, que: „Utilizamos computadoras digitales, cuya arquitectura nos es dada en forma de máquina física, con todas sus limitaciones artificiales. Primeramente debemos reducir descripciones algorítmicas continuas a descripciones que puedan ser codificadas con un aparato, cuyas operaciones fundamentales son contabilizables. Logramos esto por vía de fraccionamientos múltiples, que habitualmente se llama discretización. Finalmente, el *compiler* reduce este modelo a una forma binaria que es determinada en gran medida por imposiciones de la máquina.” [»Wir benutzen digitale Computer, deren Architektur uns in Form einer physikalischen Maschine mit all ihren künstlichen Beschränkungen gegeben ist. Wir müssen kontinuierliche algorithmische Beschreibungen erst auf Beschreibungen reduzieren, die auf einem Gerät, dessen fundamentale Operationen abzählbar sind, codiert werden können. Wir erreichen das auf dem Weg vielfältiger Zerstückelungen, die üblicherweise Diskretisierung heißen. Der Compiler schließlich reduziert dieses Modell auf eine binäre Form, die weitgehend von Maschinenzwängen bestimmt wird.]

Para el problema original, el resultado es una reproducción discreta y sintética de un micromundo, cuya estructura se puede fijar arbitrariamente por medio de un esquema de diferenciación y cualquier arquitectura de computadora que se elija. El único resto de la antigua continuidad es el uso de una aritmética basada en números, cuya particularidad es la valoración desigual de los bits y cuya consecuencia son singularidades engañosas para los sistemas no lineales.

Eso es exactamente lo que hacemos cuando creamos un modelo del mundo físico con aparatos físicos. No es el proceso idealizado y sereno que imaginamos habitualmente, cuando argumentamos sobre estructuras matemáticas fundamentales y es muy diferente de una máquina de Turing.<sup>22</sup> [Genau das tun wir, wenn wir ein Modell der physischen Welt mit physischen Geräten erstellen. Es ist nicht jener idealisierte und serene Prozeß, den wir üblicherweise beim Argumentieren über fundamentale Rechenstrukturen ausmalen, und von Turingmaschinen weit entfernt.]

Por lo tanto, ya no correspondería seguir la teoría física de Church-Turing e „inyectar así una conducta algorítmica de la que no existe evidencia, en el comportamiento del mundo físico“ [»in s Verhalten der physikalischen Welt ein algorithmisches Verhalten zu injizieren, für das keinerlei Evidenz gibt«].<sup>23</sup> Si el mundo no ha sido creado por dados arrojados por Dios, la conducta algorítmica de las nubes de lluvia o de las olas del mar no incluye, sino que excluye el hecho que sus moléculas trabajan como computadoras de su propia actividad. Si fuera a la inversa, lo importante sería, calcular por sí mismo “el precio de la posibilidad de programación” [»Preis der Programmierbarkeit«]. Esta capacidad decisiva de las computadoras, evidentemente no

<sup>21</sup> No comprendo entonces, cómo en la famosa comunicación de Turing, después de describir en la primera oración „los números calculables como aquellos números ciertos“ [»berechenbare Zahlen in Kürze als diejenigen reellen Zahlen«, “cuyas expresiones decimales se pueden calcular con medios finitos” [»deren Dezimalausdrücke mit endlichen Mitteln errechnet werden können«] (Turing 1937, pag. 19), luego definió a los números calculables como contabilizables y finalmente pudo denominar a  $\pi$ , en su carácter de “valor límite de una sucesión convergente calculable” [»Grenzwert einer berechenbar konvergenten Folge«] como número calculable (pag. 49 s.)

<sup>22</sup> Brosil Hasslacher, 1988, *Algorithms in the World of Bounded Resources*. En: Herken, 1988, pag. 421 s.

<sup>23</sup> Hasslacher; 1988, pag. 420.

tiene nada que ver con software, única y exclusivamente depende del grado en que pueda ser albergada –como sistema de escritura- por un respectivo hardware. Cuando Claude Shannon demostró en 1937 „en su trabajo magistral, quizá el de mayores consecuencias que se haya escrito nunca“ [»in der wohl folgenreichsten Magisterarbeit, die je geschrieben wurde«<sup>24</sup>], que simples relés de telégrafo pueden implementar todo el álgebra booleana, se había instalado dicho sistema de notación. Además, cuando a principios de la década del 70 el circuito integrado derivado del transistor de Shockley, combinó en un mismo chip de silicio -una resistencia controlable- con su propio óxido, -el aislador casi ideal-, la programabilidad de la materia “pudo asumir el control“ [»die Kontrolle übernehmen«],<sup>25</sup> tal como lo había pronosticado Turing. Si existiera el software, uno de los elementos más baratos de la Tierra, sería un negocio de miles de millones de dólares, pues la combinación de silicio y óxido de silicio en el chip constituye un hardware casi perfecto. Por un lado, trabajan millones de elementos de conexión en las mismas condiciones físicas, lo cual es decisivo, sobre todo, para un parámetro crítico que es la temperatura del chip e impide desviaciones de crecimiento exponencial en la tensión del transistor. Por otro, estos millones de elementos de conexión permanecen eléctricamente aislados entre si. Sólo esta relación paradójica entre dos parámetros físicos, la continuidad térmica y la discretización eléctrica, permite que sean circuitos integrados digitales y no simplemente autómatas finitos como tantas otras cosas en la Tierra y se aproximen a aquella Máquina Discreta Universal en la que se ha sumergido el nombre de su inventor –Turing- hace mucho tiempo..

Esta diferencia de estructura se puede ilustrar muy fácilmente. Por ejemplo, “una cerradura de combinación es un autómata finito, pero no se puede descomponer en una cantidad básica de componentes elementales que también pueden ser configurados nuevamente para simular un sistema físico cualquiera. En consecuencia, la cerradura de combinación no es programable estructuralmente y en este caso solamente se puede llamar efectivamente programable en un sentido restringido, ya que su estado puede establecerse para provocar una clase limitada de conductas” [»ist ein Kombinationsschloß ein endlicher Automat, kann aber nicht in eine Basismenge von elementaren Komponenten zerlegt werden, die zur Simulation eines beliebigen physikalischen Systems auch wieder neu konfiguriert werden könnten. Folglich ist das Kombinationsschloß nicht strukturell programmierbar, und in diesem Fall kann es effektiv programmierbar nur in dem eingeschränkten Sinn heißen, daß sein Zustand gesetzt werden kann, um eine eingeschränkte Klasse von Verhaltensweisen zu bewirken.«] En cambio, una computadora digital que se usa para simular una cerradura de combinación, es programable estructuralmente, pues esta conducta se logra por síntesis de una cantidad enorme de componentes de conexión elementales“ [»ist ein Digitalcomputer, der zur Simulation eines Kombinationsschlusses eingesetzt wird, strukturell programmierbar, weil dieses Verhalten durch Synthese aus einer kanonischen Menge elementarer Schaltkomponenten erreicht wird«.<sup>26</sup>

<sup>24</sup> Friedrich-Wilhelm Hagemeyer, 1979, *Die Entstehung von Informationskonzepten in der Nachrichtentechnik. Eine Fallstudie zur Theoriebildung in der Technik in Industrie- und Kriegsforschung*. Tesis. FU Berlin, pag. 432.

<sup>25</sup> Turing, 1959, *Intelligente Maschinen. Eine häretische Theorie*. En: Turing, 1987, pag. 15.

En cambio, los componentes de conexión, ya sean relés de telégrafo, válvulas electrónicas o finalmente transistores de silicio, pagan un precio por su capacidad de ser desarmados en partes discretas. Aparte del caso trivial por ser discreto, del procesamiento de textos, (que sin embargo casi desaparece detrás de todas las demás aplicaciones científicas, militares e industriales de la computadora), las computadoras digitales se siguen enfrentando a un ambiente de nubes, guerras y olas por ser los únicos “órganos SI/NO en sentido estricto de la palabra” [»Ja-Nein-Organen im strengen Sinne des Wortes«]<sup>27</sup>. Este aluvión de números grandes y ciertos, como diría Ian Hacking, solamente lo domina por agregado aditivo de nuevos elementos de conexión, ya que los 2000 transistores de la Intel 4004 se convirtieron en 1,2 millones en la actual nave insignia, la 80486. Sin embargo, es posible demostrar matemáticamente, que la cuota de aumento de las conexiones posibles entre estos elementos y por lo tanto el trabajo de cálculo como tal, es una función de la raíz cuadrada del límite superior. Dicho de otra manera, el sistema „no puede acompañar con índices de crecimiento polinómicos y menos exponenciales a todo el problema” [kann »nicht mit polynomischen Zuwachsraten des Problemumfangs mithalten«]<sup>28</sup>, um von exponentiellen Raten ganz zu schweigen«]. Justamente el aislamiento entre elementos digitales o discretos que asegura su capacidad de funcionamiento, por lo menos cuando las condiciones no son ni tropicales ni árticas, también limita la magnitud de las posibles conexiones con los vecinos locales de cada elemento. En cambio, cuando se trata de interacciones globales, como las que los chips digitales sólo conocen cuando se calientan, las posibilidades de conexiones podría aumentar hasta una barrera superior que podría ser del cuadrado de todos los elementos involucrados, de acuerdo con “las leyes de las fuerzas vigentes” [»den geltenden Kräftegesetzen«]<sup>29</sup> y la lógica combinatoria.

Sin embargo, justamente esta interrelación óptima es la que por otro lado, que se podría denominar físico, caracteriza a los sistemas no-programables. Gracias a sus efectos globales recíprocos, tales sistemas, ya sean ondas o seres, pueden presentar tasas de crecimiento polinómico de la complejidad, pero sólo son calculadas por máquinas que no deben pagar el precio de la posibilidad de ser programadas. Es evidente que este tipo hipotético, pero muy necesario, de máquinas sería puro hardware: serían máquinas físicas que trabajarían en un entorno formado por máquinas físicas que solamente estarían sometidas a las mismas limitaciones que sus fuentes, lo mismo que ellas mismas. Ya no existiría software en el sentido corriente de una abstracción siempre posible. Los procesos de una máquina de ese tipo, si bien seguirían abiertas a prescripciones algorítmicas, tendrían que trabajar sustancialmente en un sustrato material, cuya posibilidad de interrelación autorizaría reconfiguraciones cambiantes de sus celdas. Aún cuando „también este sustrato podría describirse en expresiones algorítmicas con ayuda de simulaciones, su caracterización es de tan incommensurable significación para su efectividad [...] y tan íntimamente ligado con la elección del hardware”<sup>30</sup> [»auch dieses Substrat, mithilfe von Simulationen, in algorithmischen Ausdrücken beschrieben werden könnte, ist seine Charakterisierung doch von so unermesslicher Bedeutung für die Effektivität (...) und mit der Hardware-Auswahl so eng verknüpft«], que su programación ya no tendrá nada en común con las máquinas parecidas a la de Turing.

<sup>26</sup> Michael Conrad, 1988, *The Prize of Programmability*. En: Herken, 1988, pag. 289.

<sup>27</sup> Véase, John von Neumann, 1951/1967, *Allgemeine und logische Theorie der Automaten*. Guía Nr. 8, pag. 150.

<sup>28</sup> Conrad, 1988, pag. 293.

<sup>29</sup> Conrad, 1988, pag. 290.

<sup>30</sup> Conrad, 1988, pag. 304.

Tales máquinas urgentemente necesarias y seguramente ya no tan lejanas, como son las que estudia la informática actual y ya son rondadas por la industria del chip,<sup>31</sup> podrían tentar a algunos observadores de Dubrovnik a reencontrar en ellas el rostro habitual del hombre, evolucionado o no. Puede ser. Al mismo tiempo, nuestro no menos habitual hardware de silicio ya busca satisfacer actualmente muchos de los requerimientos que se imponen a los sistemas no programables de relaciones recíprocas. Entre su millón de transistores ya tiene lugar un millón al cuadrado de relaciones recíprocas: la difusión de electrones y los efectos de túnel cuántico-mecánicos pasan por todo el chip. Sólo que la técnica de fabricación actual trata tales interacciones como barreras del sistema, efectos físicos secundarios, fuentes de perturbaciones, etc. Minimizar al menos, todos los ruidos que son imposibles de impedir, exactamente ese es el precio que debe pagar la industria de las computadoras por máquinas programables estructuralmente. La estrategia inversa, es decir, maximizar los ruidos, no solamente retomaría el camino de IBM a Shannon; si bien sería el único camino hacia aquél cuerpo de números ciertos que antes se llamaba caos.

»Cant't you understand what I'm tryin' to say«, dice - sin remake - en *Eve of Destruction*

<sup>31</sup> Así es que la primera red neuronal, procedente justamente del discreto imperio del chip de INTE, recurre, según puedo ver, por segunda vez en la historia de la empresa, después de la procesadora de señales i2920, a un simple aumento de operaciones analógicas.