*SOFTWARE**

Eloína Peláez

Al observar las dimensiones sociales del *software*, es importante recordar que el término *software* está construido en sí mismo socialmente e implica importantes cuestiones sociales. Nos hemos familiarizado tanto con el término que tendemos a olvidar que es de origen reciente y a pasar por alto que su significado literal no es el de una "cosa blanda", sino un "bien blando" o "mercancía blanda". Es significativo que el término se utilizó al mismo tiempo que se desarrollaba la idea de que el resultado de la programación podría venderse en el mercado. La mercantilización contradictoria de la programación como *software* ha sido crucial para su desarrollo.

El uso del término *software* para referirse a la programación de computadoras se remonta al inicio de los años sesentas, cuando fue inventado por analogía con el término *hardware*, incorporado a su vez a la jerga de la ingeniería en los años treintas.[1] *Software* no se usaba directamente en su sentido literal de bien blando o mercancía blanda; pero el desarrollo del término reflejaba la creciente separación de la programación con respecto de la máquina que se programaba y del surgimiento del punto de vista de que el resultado del trabajo del programador podía verse como un producto similar a la máquina misma.[2]

---

* Deseo expresar mi agradecimiento a John Holloway por las numerosas discusiones de estas ideas que encontraron expresión en este ensayo. Dicho reconocimiento no implica la aceptación de cualquier responsabilidad legal o financiera.

1 I. H. Gould. "In Pursuit of Terminology", en *The Computer Bulletin*, febrero, 1972, pp. 84-90

2 E. Peláez. *A Gift from Pandora's Box: the Software Crisis*, PhD Thesis, University of Edinburgh, 1988.

En los primeros años de la computación, los programas no trataban como un producto particular que podía comprarse o venderse. Se asumían como algo que debería compartirse libremente por los usuarios de las computadoras. En la medida en que no los desarrollaban los mismos usuarios, los programas venían "en paquete", esto es, los proporcionaba el fabricante de computadoras en la misma forma, por ejemplo, que las instrucciones de una lavadora se proporcionan con la máquina misma. Como destaca Franklin Fisher, un economista experto en "el caso de la IBM contra los Estados Unidos":

> El aspecto negativo detrás de esa venta en paquete era que resultaba muy difícil imaginar cómo vender el *software* como una propiedad. Este concepto era extraño... En esos primeros años, el *software* era algo que tu escribías y compartías con la gente.[3]

Este sobrentendido se expresaba claramente en una carta del Profesor Galler de la Universidad de Michigan al editor de *Communications of the ACM* en 1960:

> Ha llamado mi atención el hecho de que la Universidad Estatal de Arizona ha producido un paquete estadístico 704 que aparentemente aporta mucho al análisis factorial y algunas otras cosas. Desafortunadamente, la nota que ví indica un cargo de 3 dls. por el manual, 4 dls. por una caja de tarjetas binarias y 20 dls. por cuatro cajas y media de tarjetas SAP (más los derechos de envío).
> Creo que esto tiene implicaciones muy desafortunadas para la profesión de la computación. Cuando uno tiene que cubrir los costos de impresión y de las tarjetas, esto es una cosa. Pero en este caso, sin embargo, está claro que lo que se está cobrando es el desarrollo del programa, y aunque me disgusta particularmente el hecho de que esto provenga de una universidad, creo que es perjudicial para la profesión en su conjunto. No existe un paquete 704 que no se haya beneficiado directamente del libre intercambio de programas que hizo posible el COMPARTIR.Si comenzamos a vender nuestros programas, esto establecerá precedentes indeseables.[4]

Contra las objeciones de gente como Galler, la idea de que los programas podían producirse como mercancías comenzó a echar raíces al final de los

---

[3] F. Fisher. Entrevista realizada por E. Peláez.

4 W.L. Frank. "The Second Half of the Computer Age", en *Datamation*. Mayo 1976, p.92.

sesentas. La rápida expansión de la computación comercial, asociada con la llamada "tercera generación", y particularmente las series 360 de IBM, tuvieron importantes consecuencias para la mercantilización del *software*. La creciente escala de producción y utilización de computadoras significó la apertura de una nueva brecha entre el productor y el usuario: los productores estaban ocupados ahora en la producción en masa y muchos de los nuevos usuarios tenían poca experiencia en computación. La complejidad de los nuevos sistemas operativos causó enormes problemas, tanto para los productores como para los usuarios.

Esta brecha se llenó en alguna medida con el desarrollo de paquetes de *software* de una industria productora independiente. Estos paquetes se desarrollaron originalmente con fines no comerciales por grupos de usuarios de computadoras. La experiencia compartida en estos grupos condujo al "concepto de desarrollar programas de aplicación general y de esta forma al ahorro de tiempo y dinero que de otra manera se desperdiciaría *'reinventando la rueda'*".[5] Los paquetes de *software* tenían ventajas para los usuarios, tanto en términos de costos como en términos de estandarización.

Tan pronto como se comprendieron las ventajas de los paquetes de *software*, las presiones de la competencia mercantil condujeron a los fabricantes de computadoras a proporcionarlos con sus máquinas. En una época de rápida expansión, sin embargo, los fabricantes encontraron dificultades para satisfacer adecuadamente la creciente demanda, y hacia finales de los sesentas muchos paquetes de *software* ya los proporcionaban las empresas independientes. Las primeras empresas de *software* se fundaron al final de los años cincuentas y el inicio de los sesentas, funcionando como consultorías para las agencias militares y espaciales; pero no fue sino al final de los sesentas cuando creció rápidamente su número (de aproximadamente 50 en Estados Unidos, en 1965, a cerca de 500, en 1968) y cuando el concepto de compañía de *software* en tanto productora y vendedora independiente, se desarrolló primeramente. Este desarrollo lo estimuló considerablemente la decisión de IBM, en junio de 1969, de "despaquetarse", esto es, de vender su *hardware* separado de su *software*.

---

[5] R.V. Head y E.F. Linick. "Software Package Acquisition", en *Datamation*. Octubre, 1968, p.22.

El establecimiento de productores independientes de *software* no significó, sin embargo, la total mercantilización del *software*. Para que algo pueda comprarse y venderse como mercancía en el mercado, primero tiene que ser reconocido como propiedad. La naturaleza del *software* como propiedad, más allá del *cassette* o del disco sobre el cual está escrito, era problemática. En los sesentas no estaba claro aun si el *software* podría considerarse como propiedad, o lo que significaría que así lo fuera.

Un problema inherente a cualquier noción de propiedad intelectual es el que se refiere a la originalidad. Como Dansinger lo señaló, en un artículo en 1968, al enfatizar las dificultades de cualquier aplicación del concepto de propiedad al *software*, "se han hecho muchos reclamos de que un programa es meramente una simple reformulación de otro programa previamente existente".[6] Él establece una analogía con la cuestión de la originalidad en las disputas por los derechos musicales: "el negocio de la música ha sufrido durante muchos años este problema. No se ha desarrollado ningún método objetivo para resolver estas disputas".[7]

Incluso si el problema de la originalidad se pudiera resolver, no quedaría claro qué es lo que quedaría cubierto bajo el derecho de propiedad del *software*: ¿sería el código detallado de un programa o el diagrama lógico, o los resultados, o el algoritmo subyacente al programa, o el 'ver y sentir' del *software*?

Un tercer problema a resolver antes de tratar el *software* como propiedad fue la forma de protección. Las dos principales formas de derechos de propiedad intelectual son los derechos de reproducción (*copyright*) y la patente. Bajo la Ley de Derechos de Reproducción de los Estados Unidos, el autor o el propietario de los derechos de reproducción tiene control exclusivo durante 28 años sobre los derechos a reproducir cualquier forma de expresión. Bajo la Ley de Patentes los inventores tienen el derecho exclusivo a controlar el uso de sus inventos y los métodos que ellos implican durante un período

---

[6] S.J. Dansiger. "Proprietary Protection of Computer Programs", en *Computers and Automation*. Febrero, 1968, p.32.

[7] *Idem.*

de 17 años. Los derechos de reproducción se registran y cualquier disputa sólo tiene lugar después del registro; en el caso de las patentes, éstas se solicitan y la Oficina de Patentes examina la solicitud para saber si el invento en cuestión satisface los criterios de:

a) no haber sido inventado con anterioridad, y
b) no ser obvio.

La Oficina de Derechos de Reproducción de Estados Unidos decidió, en mayo de 1964, que los programas de computación podían registrarse si se satisfacían ciertos requisitos. Sin embargo, la aceptación de programas para su registro no necesariamente dio protección legal a los programas porque la Oficina de Derechos de Reproducción todavía no asumía ninguna posición sobre si un programa podía considerarse como algo "escrito por un autor". La oposición a la idea de que los programas podían ser sujetos de un derecho de reproducción provino, principalmente, de las universidades que argumentaban que dicho derecho obstaculizaría el desarrollo de la programación computacional. Si los derechos de reproducción hubieran existido desde el comienzo — argumentaban ellos— la preparación de programas habría "sido emprendida bajo la amenaza constante de acciones de infracción sujetas a cargos de plagio con relación a los programas protegidos por derechos ya existentes" y "es dudoso que el incremento de los programas y de las técnicas de programación en años recientes hubiera sido posible".[8]

La mayor parte de la discusión sobre la protección de la propiedad de *software*, durante los sesentas, se concentró sobre la cuestión de las patentes. La aproximación inicial de la Oficina de Patentes fue que los programas de computación no eran patentables porque no eran "métodos o aparatos, sino más bien procesos o fórmulas matemáticas".[9] Sin embargo, bajo la presión de las compañías de *software*, y después de haber sido revocadas en diversos casos por las Cortes, la Oficina de Patentes finalmente revisó sus lineamientos, al final de 1969, y anunció que en el futuro la solicitud de patentes para *software* se consideraría.[10]

[8] R.P. Bigelow. "Legal Aspects of Proprietary Software", en *Datamation*. Octubre, 1968, p.33.
[9] *Idem.*
[10] "As We Go to Press", en *Computers and Automation*. Noviembre, 1969, p.11.

En realidad, las implicaciones prácticas de patentar se han limitado en los hechos. El desarrollo del *software* ha sido demasiado rápido y la maquinaria de protección de patentes demasiado lenta como para que las patentes tengan una gran significación en la protección del *software*. En la práctica, la forma más común para tratar de proteger el *software* ha sido mediante el secreto en el comercio de programas apoyado en previsiones contractuales, sea entre proveedores y usuarios o entre patrones y programadores. La forma más común de esto es la llamada —*shrink-wrap licence*—: un contrato detallado sobre la cubierta o dentro de la envoltura de la mayoría de los discos de *software*, el cual acepta el comprador al abrir el paquete. La mayoría de semejantes contratos estipula que el comprador no puede usar o copiar el *software* excepto bajo los términos del contrato.[11]

La inefectividad de la protección del *software* como propiedad ha devenido más y más obvia a la luz del desarrollo de la computación desde los años sesentas, especialmente la proliferación de computadoras personales y el establecimiento de redes de comunicación internacional entre computadoras. El hecho de que los costos de reproducción del *software* sean minúsculos en comparación con los precios del mercado hace extremadamente difícil imponer relaciones mercantiles. Se asume dentro de la industria del *software*, que existe al menos una copia no autorizada por cada venta autorizada de un programa de *software* y la *Software Publishing Association* estima que los productores de *software* perdieron aproximadamente 1 billón de dls. en ventas, durante 1986, como resultado del copiado no autorizado (tanto para fines de ganancia como para uso personal).[12]

Los remedios legales no han sido capaces de mantenerse en línea con el extendido rechazo a aceptar el *software* como propiedad. Por lo menos una Corte en Estados Unidos sostuvo que las *shrink-wrap licences* no son válidas legalmente y que son, en cualquier caso, difíciles de imponer. Las compañías de *software* volvieron su atención hacia las leyes de patentes y derechos de reproducción. En Estados Unidos las Cortes extendieron la aplicación de la

---

[11] A.W. Branscomb. "Who Owns Creativity?", en Forester, T.(ed), *Computers in Human Context*. Blackwell, Great Britain, 1989, p. 412.

[12] *Ibid.* pag. 410.

ley de derechos de reproducción al aceptar que puede protegerse no sólo el código base, sino también el código objeto, e incluso los diagramas de flujo que contienen la lógica y la secuencia del programa. Más recientemente, las Cortes reconocieron que las características "no literales" de un programa, como su distintivo "ver y sentir" pueden protegerse por derechos de reproducción.[13] El ejemplo más reciente de este tipo de casos es la demanda emprendida por la Xerox en contra de Apple, reclamando más de 150 millones de dls. en daños, y señalando que el desplegado de la pantalla distintivo de las computadoras Macintosh viola los derechos de propiedad de la Xerox. En su demanda, Xerox indica que el anterior presidente de Apple, Mr. Steve Jobs, visitó el Centro de Investigación de la Xerox en Palo Alto, durante 1979, y quedó muy impresionado con el "exclusivo y revolucionario diseño *user-friendly* de uno de los productos de la Xerox, y de esta forma obtuvo la idea para el desplegado de la Macintosh. Mr. Jobs no ocultó el hecho de sus visitas a la Xerox ni la influencia que tuvieron sobre el desarrollo de la Macintosh; pero niega haber infringido algún derecho de propiedad, en tanto Apple no ha protegido legalmente la "idea" de un utensilio gráfico interfásico, sino la "expresión" de esa idea. Mientras tanto, Apple está demandando a Microsoft y a Hewlett-Packard por infringir sus propios derechos de reproducción.[14]

El caso de Apple pone de relieve las dificultades para imponer los conceptos de propiedad sobre las "ideas" y su "expresión". Tanto los elementos fantásticos de estos casos como las dificultades generales para imponer la protección del *software* como propiedad, sugieren que en un nivel el *software* jamás ha sido aceptado plenamente como propiedad. La tradición de que compartir el *software* entre los usuarios sin ningún cargo queda fuera de discusión, como Fisher y Galler señalaron claramente durante los años cincuentas y sesentas, sigue vigente en gran medida. La reproducción no autorizada de *software*, al menos para usos privados, no es tanto una falta de respeto por la propiedad sino un rechazo a la aceptación del establecimiento de derechos de propiedad. Entre numerosos programadores y académicos se da por sobrentendido el compartir los programas y las ideas de una forma no

[13] *Ibid.* p. 412.

[14] "Xerox Files Sofware Suit Against Apple Mac", en *Financial Times,* 16 de diciembre, 1989,

mercantilizada. Es notable, por ejemplo, en el desarrollo de las computadoras paralelas, donde las universidades han jugado por primera vez un papel extremadamente importante para el desarrollo tecnológico, que figuras prominentes hablen orgullosamente de haber contribuido con importantes ideas para el trabajo de amigos, en otros proyectos. Así, Charles Seitz, quien encabeza el desarrollo del hipercubo en Caltech, se deleita contando como él y su equipo contribuyeron con la idea del cubo-n binario para la Máquina de Conexión de Danny Hillis desarrollada en el MIT; si él siguiera la lógica del caso Xerox-Apple, probablemente emprendería una acción legal en contra de Hillis para recuperar algunas de las ganancias logradas con la Máquina de Conexión.[15]

El rechazo a captar el *software* como propiedad tiene una dimensión internacional, que se vuelve cada vez más importante con la expansión de la comunicación de datos a nivel internacional. Un informe reciente en el *Financial Times* señala que la reproducción no autorizada de *software* en Italia está causando una preocupación particular en los productores de *software*: no hay una ley específica que prohiba el copiado de *software* y los italianos, según sugiere el artículo, muestran poco respeto por los derechos de propiedad de las compañías de *software*.[16] Branscomb se refiere al extendido uso no autorizado de *software* en países como Corea y Taiwan para poder desarrollar su capacidad industrial.[17] Aun si el *software* se protegiera efectivamente por leyes de patentes y de derechos de reproducción locales, esto sería extremadamente difícil a nivel internacional. El ejemplo cubano del tratamiento del derecho de reproducción, basado en el rechazo a aceptar que el conocimiento pueda poseerse privadamente, sugiere que no se trata de un asunto de honestidad, como se alega algunas veces, sino de comprensión de que las relaciones de propiedad son relaciones de poder.

Muchas cuestiones similares surgen en relación con otra de las dimensiones de la protección del *software* como propiedad, la que ha recibido mucha publicidad últimamente. Los virus de computadoras, el fraude y el sabotaje

---

[15] C. Seitz. Entrevista realizada por E. Peláez.

[16] "Software Pirates Find Italy's Law Has no Byte", en *Financial Times*, 28 de abril, 1988.

[17] *Op.cit.*, p. 410.

parecen causar daños gigantescos en términos financieros, e incalculables son también los gastos desembolsados para tratar de proteger de interferencias los sistemas de cómputo. Aunque el problema se plantea usualmente en términos de deshonestidad, la discusión de las propuestas recientes para penalizar el ingreso no autorizado en sistemas de cómputo ha puesto en evidencia que aquí también se trata de relaciones de poder. Algunas veces se establece una analogía entre el ingreso no autorizado en los sistemas y un allanamiento de morada, pero cualquier comparación semejante implica el endurecimiento del concepto de *software* como propiedad, lo que va en contra de una importante tradición en el uso de las computadoras. Muchos de quienes acceden sin autorización a los sistemas protestan contra esta impuesta privatización del conocimiento, que implica, en la práctica, medidas basadas en la analogía ya mencionada.

La existencia del *software* como "bien blando", esto es, como una mercancía de estatuto problemático, tiene importantes consecuencias para la forma en la cual se desarrolla. El asunto de la protección del *software* como propiedad no es simplemente una cuestión legal externa a su producción. Las contradicciones del concepto de *software* como propiedad se expresan inevitablemente en la producción real de paquetes de computación, como tensiones dentro de la programación o entre los actores involucrados. Esto se puede observar en muchas formas. Se expresa como tensiones entre las universidades y el "mundo real" de la industria en torno a la forma apropiada de desarrollar las técnicas de programación (como ocurre con la programación estructurada o ingeniería del *software*); se expresa en el interior de las universidades como tensiones en los proyectos en cuanto a las formas de relación apropiadas con el mundo del comercio; se expresa dentro de la industria como tensiones entre los expertos programadores, frecuentemente más interesados en cuestiones técnicas que en asuntos de ganancias, y sus administradores, quienes tienen que preocuparse más en consideraciones financieras.[18] Se expresan también en contradicciones en torno al sometimiento del trabajo intelectual a la disciplina del mercado. La idea de que el *software* podría producirse en la misma forma que cualquiera otra mercancía

[18] Peláez, *op.cit.*

fue duramente criticada en el libro de Fred Brooks, *The Mythical Man-Month,*[19] y el hecho de someter la programación a consideraciones comerciales normales fue agriamente satirizada por el Profesor Edseger Dijkstra en su serie de cartas acerca de los avatares de la ficticia *Mathematics Inc.* El tratamiento del *software* como una mercancía implica la imposición de criterios de eficiencia y productividad sobre aquéllos que lo producen, criterios que bien pueden resultar absurdos en términos de la actividad misma. Así, por ejemplo, Dijkstra señala cuán ridículo y cuán dañino resulta el medir la productividad en la programación en términos del número de líneas de código producidas, cuando el objetivo de la programación debe ser el contrario: resolver problemas en el menor espacio posible. Él pone en boca de su presidente de *Mathematics Inc.* las siguientes palabras:

> Regresamos a nuestro viejo método para medir la productividad: desde febrero de 1974 medíamos la productividad matemática por el número de nuevos resultados obtenidos cada mes; ahora regresamos a una técnica más realista y, sobre todo, más objetiva de contar el número de líneas de prueba producidas cada mes.[20]

Las contradicciones al tratar el *software* como una mercancía recorren la totalidad del desarrollo de la programación. Se reproducen tanto en el proceso de producción como en las fantasías de la legislación. El proceso de mercantilización del *software* es, y continuará siendo, un proceso contradictorio. Los intentos por definir el *software* como una propiedad han hecho poco para modificar la práctica de la gente de compartir ideas y programas. El espíritu de la programación original sigue estando vigente.

---

[19] F.J. Brooks. *The Mythical Man-Mont: Essays on Software Engineering,* Addison-Wesley, Reading, Massachusetts, 1974.

[20] E.W. Dijkstra. *Selected Writings on Computing: a Personal Perspective,* Springer Verlag, New York, 1982, p. 185.