



Por qué el software no debe tener propietarios*

RICHARD STALLMAN

Las tecnologías digitales de la información contribuyen al desarrollo del mundo haciendo que sea más fácil copiar y modificar información. Es lo que prometen hacer los ordenadores y, además, de una forma más fácil para todos.

Pero no todo el mundo quiere que sea más fácil. El sistema del *copyright* permite que los programas de *software* tengan «propietarios», la mayor parte de los cuales pretenden privar al resto del mundo del beneficio potencial del *software*. Los propietarios desearían ser los únicos con capacidad para copiar y modificar el *software* que usamos.

El sistema de *copyright* creció con la imprenta, una tecnología para la producción masiva de copias. El *copyright* se ajustaba bien a esta tecnología puesto que era restrictiva sólo para los productores masivos de copias. No privaba de libertad a los lectores de libros. Un lector cualquiera, que no poseyera una imprenta, sólo podía copiar libros con tinta y pluma, y a pocos lectores se les ponía un pleito por ello.

Las tecnologías digitales son más flexibles que la imprenta: cuando la información adopta forma digital, se puede copiar fácilmente para compartirla con otros. Es precisamente esta flexibilidad la que se ajusta mal a un sistema como el del *copyright*. Esa es la razón del incremento de medidas perversas y draconianas que se emplean en la actualidad para hacer cumplir el *copyright* del *software*.

Consideremos estas cuatro prácticas de la SPA¹: 1) Propaganda masiva en la que se afirma que está mal desobedecer a los propietarios para ayudar a un amigo. 2) Petición de que se convierta uno en chivato para delatar a sus colegas y compañeros de trabajo. 3) Redadas

* Copyright 1994, 1998, Richard STALLMAN. Se reproduce este artículo aparecido en el Nro. 47, 2001, de la Revista *Archipiélago*, Cuadernos de Crítica de la Cultura, www.archipiélago-ed.com, Madrid, de conformidad a la autorización que aparece en el mismo texto. El artículo que presentamos se publicó por vez primera en 1994, aunque fue revisado por su autor en 1998. Aun siendo ya un clásico en la Red, no había sido publicado antes en papel. Traducción de Stan Bark. Revisión y notas de Miguel Vidal.

1. SPA son las siglas de la estadounidense Software Publisher's Association («Asociación de Editores de *Software*»). Al igual que la BSA (Business Software Alliance) –que se comporta en términos estrictos como una organización parapolicial- es ya legendaria la beligerancia de la SPA contra la

(con ayuda policial) en oficinas y escuelas, en las que la gente debe probar su inocencia ante la acusación de hacer copias ilegales. 4) El proceso judicial por parte del gobierno de EE.UU., a petición de la SPA, contra personas como David LaMacchia, del MIT, no por copiar *software* (no se le acusa de copiar nada), sino sencillamente por dejar sin vigilancia equipos de copia y no censurar su uso.

Cada una de estas cuatro prácticas se asemeja a las usadas en la antigua Unión Soviética, donde todas las copiadoras tenían un vigilante para prevenir copias prohibidas, y donde las personas tenían que copia información en secreto y pasarla de mano a mano en forma de *samizdat*. Por supuesto hay una diferencia: el motivo para el control de información en la Unión Soviética era político: en los EE.UU. el motivo es el beneficio económico. Pero son las acciones las que nos afectan, no el motivo. Cualquier intento de bloquear el compartir información, no importa la causa, lleva a los mismos métodos y a la misma dureza.

Los propietarios utilizan diversos argumentos para que se les conceda el control del modo en que usamos la información.

Insultos. Los propietarios usan palabras difamatorias como «piratería» y «robo», al igual que terminología técnica como «propiedad intelectual» y «daño», para sugerir una cierta línea de pensamiento al público: una analogía simplista entre programas y objetos físicos.

Nuestras ideas e intuiciones acerca de la propiedad sobre los objetos materiales suelen referirse a si es justo *quitarle un objeto* a alguien. No se aplican directamente a *hacer una copia* de algo. Pero los propietarios nos piden que las apliquemos en cualquier caso.

Exageración. Los propietarios dicen que sufren un «daño» o «pérdida económica» cuando los usuarios copian programas por su cuenta. Pero el copiar no tiene un efecto directo sobre el propietario, y no hace daño a nadie. El propietario sólo puede perder si la persona que hizo la copia hubiese pagado por una del propietario.

Un mínimo de reflexión muestra que la mayoría de tales personas no habrían comprado esas copias. Aún así, los propietarios calculan sus «pérdidas» como si todos y cada uno hubiesen comprado una copia. Esto es una exageración, por decirlo de una manera suave.

La ley. Los propietarios a menudo describen el estado actual de la ley, así como las duras sanciones con las que nos amenazan. En este enfoque va implícita la sugerencia de que la ley actual refleja un punto de vista moral incuestionable —y así, al mismo tiempo, se nos insta a considerar estas sanciones incluso como hechos naturales por los que no se puede responsabilizar a nadie. Esta línea de persuasión no está diseñada para defenderse ante el pensamiento crítico; está concebida para reforzar un lugar común. Es evidente que las leyes no distinguen lo que está bien de lo que está mal. Todo estadounidense debería saber que, hace

llamada «piratería»: ejerce todo tipo de presiones, mentiras y amenazas que luego sirven de modelo a aprendices de policías como la SGAE española; *invita* a la gente a informar sobre sus compañeros y amigos, y promueve una política de «responsabilización», en la que los dueños de ordenadores deben hacer cumplir activamente las leyes de *copyright*, si no quieren ser castigados. En 1996, en su clásico «El derecho a leer», el propio Stallman ya avisaba de cómo la SPA estaba amenazando a pequeños proveedores de Internet, exigiéndoles que le permitieran espiar a sus usuarios. Muchos proveedores se rinden cuando les amenazan, porque no pueden permitirse litigar en los tribunales.

40 años, iba contra la ley que una persona de raza negra se sentase en la parte delantera del autobús; pero solamente los racistas dirían que sentarse ahí no estaba bien.

Derechos naturales. Los autores a menudo sostienen que existe una relación especial con los programas que han escrito y añaden que, en consecuencia, sus deseos e intereses respecto al programa simplemente prevalecen sobre los de cualquier otra persona –o incluso de los del resto del mundo. (Normalmente son las empresas, no los autores, los que retienen los *copyrights* sobre el *software*, pero se espera de nosotros que ignoremos esta discrepancia.)

A quienes proponen esto como un axioma ético –el autor es más importante que tú– sólo les puedo decir que yo mismo, un notable autor de *software*², lo considero una tontería. Pero la gente por lo general sólo suele sentir alguna simpatía hacia los derechos naturales por dos razones. Una razón es una analogía forzada entre el *software* y los objetos materiales. Cuando yo cocino espaguetis, me quejo si otra persona se los come, porque entonces yo ya no me los puedo comer. Su acción me duele exactamente tanto como lo que le beneficia a él; sólo uno de nosotros se puede comer los espaguetis, así que la pregunta es: ¿quién? La más mínima distinción entre alguno de nosotros es suficiente para inclinar la balanza ética. Pero el hecho de que tú ejecutes o modifiques un programa que yo he escrito te afecta a ti directamente y a mí indirectamente. Si tú le das una copia a tu amigo te afecta a ti y a tu amigo mucho más que lo que me afecta a mí. Yo no debería tener el poder de decirte que no hagas estas cosas. Nadie debería.

La segunda razón es que a la gente se le ha dicho que los derechos naturales de autor son una tradición aceptada e indiscutida de nuestra sociedad. Desde un punto de vista histórico, sucede justamente lo contrario. La idea de los derechos naturales de autor fue propuesta y decididamente rechazada cuando se concibió la Constitución de los Estados Unidos. Esa es la razón por la que la Constitución sólo *permite* un sistema de *copyright* y no *requiere* uno; por esa razón dice que el *copyright* debe ser temporal. Establece asimismo que el propósito del *copyright* es promocionar el progreso, no recompensar a los autores. El *copyright* recompensa a los autores en cierta medida, y a los editores más, pero se concibe como un medio de modificar su comportamiento.

La tradición realmente establecida de nuestra sociedad es que el *copyright* vulnera los derechos naturales del público, y que esto sólo se puede justificar por el bien del público.³

Economía. El último argumento que se emplea para justificar la existencia de los propietarios de *software* es que esto lleva a la producción de más *software*. A diferencia de los anteriores, este argumento por lo menos adopta un enfoque legítimo sobre el tema.

-
2. Evidentemente Stallman peca aquí de modestia pues es *mucho más* que un «notable programador»: suyas son algunas de las mejores piezas del *software* hoy existentes, como el editor Emacs, el compilador GCC y el depurador GDB.
 3. Stallman se dedicó extensamente a demostrar esta idea –que el *copyright* es un derecho artificial que viene a regular el derecho natural a la copia– en la conferencia que ofreció en julio del año 2000 en la Universidad de Burdeos, en el marco de la Conferencia de Debian, y que llevaba por título: «El *copyright* contra la comunidad en la era de los ordenadores». Existe traducción castellana en www.sindominio.net/biblioweb/telematica/stallman-copyright.html

Se basa en un objetivo válido: satisfacer a los usuarios de *software*. Y está empíricamente demostrado que la gente producirá más de algo si se le paga bien por ello.

Pero el argumento económico tiene un defecto: se basa en la presunción de que la diferencia es sólo cuestión de cuánto dinero debemos pagar. Asume que la «producción de *software*» es lo que queremos, tenga el *software* propietarios o no. La gente acepta gustosamente esta presunción porque está de acuerdo con nuestra experiencia acerca de los objetos materiales. Considérese un bocadillo, por ejemplo. Es posible que puedas conseguir un bocadillo equivalente gratis o bien por una cantidad de dinero. Si es así, la cantidad que pagas es la única diferencia. Tanto si lo tienes que comprar como si no, el bocadillo tiene el mismo sabor, el mismo valor nutricional y en ambos casos te lo puedes comer sólo una vez. El hecho de si el bocadillo lo obtienes de un propietario o no, no puede afectar directamente a nada más que a la cantidad de dinero que te queda después.

Esto es cierto para cualquier objeto material: el hecho de que tenga o no tenga propietario no afecta directamente a lo que *es*, o a lo que puedas hacer con ello si lo adquieres. Pero si un programa tiene un propietario, esto afecta en gran medida a lo que es, y a lo que puedes hacer con una copia si la compras. La diferencia no es sólo una cuestión de dinero. El sistema de propietarios de *software* incentiva a los propietarios de *software* a producir algo, pero no lo que la sociedad realmente necesita. Y causa una contaminación ética intangible que nos afecta a todos.

¿Qué es lo que la sociedad necesita? Necesita información que esté verdaderamente a disposición de sus ciudadanos, por ejemplo, programas que la gente pueda leer, arreglar, adaptar y mejorar, no solamente ejecutar. Pero lo que los propietarios de *software* típicamente ofrecen es una caja negra que no podemos ni estudiar ni modificar.

La sociedad también necesita libertad. Cuando un programa tiene un propietario, los usuarios pierden la libertad de controlar una parte de sus propias vidas. Y, sobre todo, una sociedad necesita incentivar el espíritu de cooperación entre sus ciudadanos. Cuando los propietarios de *software* nos dicen que ayudar a nuestros vecinos de una manera natural es «piratería», están contaminando el espíritu cívico de nuestra sociedad. Por ello decimos que el *software* libre es una cuestión de libertad, no de precio.⁴

El argumento económico para justificar la propiedad es erróneo, pero la cuestión económica es real. Algunas personas escriben *software* útil por el placer de escribirlo o por la

4. En inglés, la polisemia del término *free* obliga a insistir en este punto y deshacer la ambigüedad. En castellano disponemos de dos palabras, «libre» y «gratis», pero muchas traducciones se encargan lamentablemente de mantener la confusión y aún agravarla al traducir *free* por «gratis», totalmente erróneo en este contexto: existe *software* gratuito que es propietario (el navegador Microsoft Explorer, por ejemplo) y nada impide vender el *software* libre, aunque ciertamente se debe ofrecer algo extra –normalmente en forma de servicios añadidos– para que alguien compre algo que puede obtener legítimamente sin pagar por ello. La gratuidad en este caso es una consecuencia del modelo –en el que el programador puede que haya cobrado por su trabajo–, pero de ningún modo es lo que define al *software* libre.

admiración y amor al arte⁵; pero si queremos más *software* del que esas persona escriben, necesitamos conseguir fondos. Desde hace ya diez años, los desarrolladores de *software* libre han probado varios métodos para recabar fondos, con algo de éxito. No hay necesidad de hacer rico a nadie; los ingresos medios de una familia media, alrededor de 35.000 dólares, resultan ser un incentivo suficiente para muchos trabajos que son menos satisfactorios que programar.

Durante años, hasta que una beca lo hizo innecesario, yo me ganaba la vida realizando mejoras a medidas sobre *software* libre que yo había escrito. Cada mejora se añadía a la versión estándar lanzada y así, finalmente, quedaba disponible para el público en general. Los clientes me pagaban para que trabajase en las mejoras que ellos querían, en vez de en las características que yo habría considerado como de máxima prioridad.

La Fundación para el *software* Libre, una entidad sin ánimo de lucro exenta de impuestos para el desarrollo de *software* libre, consigue fondos mediante la venta de CD-ROMS de GNU⁶, camisetas, manuales y distribuciones *deluxe* (todo lo cual los usuarios son libres de copiar y modificar), así como mediante donaciones. Ahora cuenta con un equipo de cinco programadores y tres empleados que se encargan de los pedidos por correo.

Algunos desarrolladores de *software* libre ganan dinero mediante la venta de servicios de soporte. Cygnus Support⁷, que cuenta con alrededor de 50 empleados, estima que en torno al 15% de la actividad de su equipo es desarrollo de *software* libre –un porcentaje respetable para una compañía de *software*.

Algunas compañías, incluyendo Intel, Motorota, Texas Instruments y Analog Devices, han unido esfuerzos para financiar el desarrollo continuado del compilador GNU para el lenguaje C. Mientras, el compilador GNU para el lenguaje Ada está siendo financiado por la Fuerza Aérea de los EE UU, que cree que ésta es la manera más efectiva de conseguir un compilador de alta calidad.⁸

-
5. Aunque no resulte chocante a primera vista, no sólo programadores sino algunos teóricos consideran la programación «una de las bellas artes» (De la *programmation considerée comme une des Beaux arts* es precisamente el título de una obra de Pierre Lévy). Por su parte Franco Berardi, *Bifo*, afirma en «Trabajo Cero» que la programación puede ser valorada no sólo «como ejecución de un proyecto predefinido, no como simple elaboración de los procedimientos a través de los cuales se pone en funcionamiento un cierto proceso, sino como redefinición del contexto mismo y como elaboración de procedimientos afortunados».
 6. El Proyecto GNU (acrónimo recursivo que significa GNU'S Not Unix, o sea, «GNU No es UNIX») nació en 1984 de la mano de Richard Stallman, por entonces un *hacker* del emblemático Laboratorio de Inteligencia Artificial del Massachusetts Institute Technology (MIT), cuna de grandes *hackers*. El proyecto GNU se propuso a la sazón una tarea titánica: construir un sistema operativo libre completo. No es sencillo expresar en pocas palabras la enorme dificultad que comporta un proyecto así, en principio sólo al alcance de unas cuantas compañías con miles de programadores a sueldo.
 7. Cygnus fue la primera empresa importante que trabajó con *software* libre. Sus aportaciones a la comunidad del *software* libre –liberando código y manteniendo herramientas críticas como el compilador C de GNU- han sido numerosas e importantes. En 1999 fue adquirida por Red Hat, una gran compañía que basa por completo su modelo de negocio en el *software* libre.
 8. La financiación de la Fuerza Aérea se acabó hace algún tiempo; el Compilador GNU de Ada está ahora en servicio, y su mantenimiento se financia comercialmente.

Todos estos ejemplos son modestos⁹; el movimiento de *software* libre es pequeño y todavía joven. Pero el ejemplo de la radio «mantenida por la audiencia»¹⁰ en EE.UU. muestra que es posible mantener una actividad grande sin forzar a cada usuario a pagar.

Como usuario de informática hoy en día, puede que estés utilizando un programa propietario. Si tu amigo te pide hacer una copia, estaría mal negarse a ello. La cooperación es más importante que el *copyright*. Pero una cooperación clandestina, escondida, no contribuye a mejorar la sociedad. Una persona debería aspirar a vivir una vida honrada abiertamente con orgullo, y esto significa decir «no» al *software* propietario.

Te mereces ser capaz de cooperar abierta y libremente con otras personas que usan *software*. Te mereces ser capaz de aprender cómo funciona el *software*, y enseñar a tus estudiantes con él. Te mereces ser capaz de contratar a tu programador favorito para arreglarlo cuando se estropee.

Te mereces el *software* libre.

-
9. Desde que Stallman revisó este artículo por última vez, hace apenas tres años, la situación ha cambiado sobremanera y se han multiplicado las iniciativas comerciales, que ya no son tan «modestas» como las que citaba: en torno al *software* libre han surgido cientos de nuevas empresas, hasta el punto de convertirse en los dos últimos años en uno de los sectores más dinámicos del ya de por sí dinámico sector informático. Muchas de esas empresas mantienen modelos de negocio tradicionales basados en la prestación de servicios, pero otras están abriendo nuevas vías. No ha faltado incluso la incursión de capital financiero y especulativo en empresas del mundo Linux, como VA Linux y Red Hat, cuya salida a bolsa fue espectacular en ambos casos.
 10. Se refiere a las llamadas radios públicas, que tienen algún parecido con las radios libres. Para mantenerse sin necesidad de publicidad y sin control mediático reciben donaciones de sus oyentes, que no pagan por un servicio sino por mantener en antena y sin dependencias comerciales algo que cualquiera escuchará gratis.