

Dependencia Tecnológica y Software Libre

Technological Dependence And Free Software

Darío García R.*

RESUMEN

En este análisis se apunta a otras alternativas para entender el conocimiento – como orientador de la acción –, a distintas formas de vivir el avance tecnológico en esta época de globalización y de continuo cambio en la composición de competencias para el aprendizaje y el trabajo; de restricción y movilidad de fronteras, sin olvidar el veloz cambio en la generación y aceptación – adaptación – de nuevos conceptos en la técnica, la tecnología y la ciencia; de manera especial, la de información y la comunicación (llamadas TIC's), y así lograr una adecuada participación de mayor competencia y validez en el ámbito mundial contemporáneo para el reconocimiento y desarrollo de la nación.

ABSTRACT

This analysis points to other alternatives to understand the knowledgement - like guide of the action -, to different forms of living the technological advance in this globalization time and of continuous change in the composition of competitions for the learning and the work; of restriction and mobility of frontiers, without forgetting the speedy change in the generation and acceptance - adaptation - of new concepts in the technique, the technology and the science; in the special way, the information and the communication (named TIC's), and in this way to achieve an appropriate participation of more competition and validity in the contemporary world environmental for the recognition and development of the nation

Palabras clave: conocimiento, aprendizaje, competencias educativas

Key words: knowledge, learning, educational competitiveness

Fecha de recepción: 20 de noviembre 2006.

Fecha de aprobación: 6 de diciembre 2006.

* Ingeniero de Sistemas Universidad Nacional de Colombia. Estudios de Maestría en Tecnologías de la información aplicadas a la educación. Universidad Pedagógica Nacional. Docente de Educación Superior del I.T.C. garciarudar@gmail.com

1. Introducción

Un papel importante en las Instituciones de Educación Superior es el de fomentar encuentros y debates sobre los desarrollos tecnológicos que se relacionan con la industria, la sociedad y en general, los que afectan al conjunto de la población por sus alcances y repercusiones en su desarrollo personal como miembros de una sociedad democrática y libre, tanto directa como indirectamente.

Para tener un encuentro académico, enriquecedor e interesante al interior de las Instituciones, lo primero que se debe lograr, es el acceso libre al conocimiento, sin ambigüedades y con puntos de vista diferentes e interesantes, sustentados coherentemente con alcance académico global desde una perspectiva local, conocedora de las realidades propias y particulares del país. Entendido este acercamiento sin la antigua venerabilidad hacia los “dueños del saber” global, mercaderes usurpadores del conocimiento que forman y deforman el saber de acuerdo con sus intereses culturales, políticos y económicos.

2. Software

El conjunto de técnicas que hace funcionar una máquina es llamado en forma general: el “código”. Puede estar implementado en un programa de computador – el software – o grabado en un circuito electrónico – el hardware –, y contiene un conjunto de órdenes e instrucciones, que dirige el funcionamiento de las máquinas. Estas máquinas, en un caso particular son los computadores, los cuales definen y controlan cada vez más nuestras vidas. Determinan qué y cuál información se tiene de los ciudadanos y en dónde se guarda dicha información, cuándo se encienden y cambian los semá-

foros de una ciudad, cómo y cuánto dinero se puede retirar en los cajeros electrónicos de una red bancaria, qué datos se almacenan y presentan de una comunidad en la red Internet. Controlan si una imagen, una canción o una película se pueden enviar de Internet a un computador conectado a la red. Controlan la información que un programa o un computador transmiten a su fabricante cuando se navega por Internet.

Estas máquinas nos controlan y dirigen. Y en su parte más recóndita, el “código” dirige estas máquinas. ¿Sabemos que dicho “código existe”? ¿Qué deberíamos saber sobre el “código”? ¿Cómo nos controla y hacia dónde nos dirige? ¿Qué control deberíamos tener sobre éste “código”? ¿Qué poder dirige el “código” y hacia dónde? ¿Qué libertad debería haber para conocer y neutralizar el control que este “código” tiene?

Con estas preguntas, vemos la necesidad de ser cuidadosos en tener, entender y mantener la suficiente información y conocimiento para un análisis completo acerca del manejo dado por una "sociedad democrática y libre" en la implementación del “código”, para que sea verdaderamente “democrático y libre”. Código “libre” en el sentido de que el control, construido y elaborado por los codificadores o programadores y “ordenado” por sus jefes, sea aceptado y claro para todos; y en el que cualquier persona tenga derecho a tomar ese control y modificarlo.

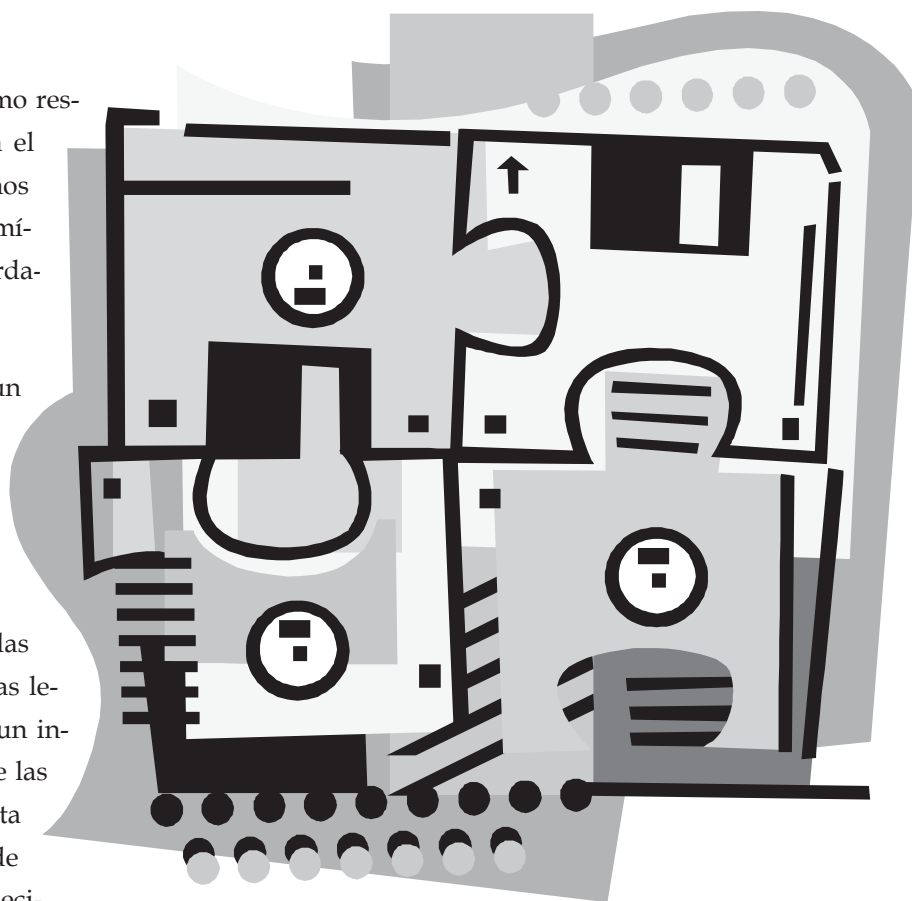
“Libre” (en inglés, Free) posee diversos significados, sólo uno de ellos se refiere a “precio”. Un significado esencial de “libre” es el de “libertad de expresión” o al de “trabajo no forzado”. El término aceptado y empleado es “libre en el sentido de limitado en cuanto a su control por los otros”, no libre como gratuito. No “libre” en el sentido de que los escritores del “código” no reciban una paga.

3. Software Libre

El “software libre” se da como respuesta a un mundo cimentado en el “código”. Un “código” que en muchos casos impide establecer las pautas mínimas para lograr una sociedad verdaderamente “democrática y libre”.

“Software libre” significa un control transparente y dispuesto a la modificación, de la misma forma que las leyes de una “sociedad libre”, son libres cuando hacen su control comprensible y abierto a la comunidad y pueden ser modificadas cuando no cumplan su objetivo. Las leyes son, en el mejor de los casos, un intento de alcanzar justicia; decir que las leyes definen la justicia o la conducta ética es una interpretación errónea de las cosas. La idea de que las leyes deciden qué está bien o qué está mal responde normalmente a una confusión.

Richard Stallman¹ inicia su trabajo en el Instituto de Tecnología de Massachusetts (en inglés, Massachusetts Institute of Technology, MIT), una de las principales universidades de investigación del mundo, situada en Cambridge (Massachusetts, Estados Unidos), como programador y desarrollador de software en sistemas operativos. Su carrera en la vida pública se da como programador y diseñador de software, además es fundador del “movimiento por la libertad del software”. Su intención es, producir “código” en la medida que pueda ser transparente y susceptible de modificación haciéndolo verdaderamente “libre”; acorde con la so-



iedad planteada en La Constitución e invocada por los dirigentes en el poder.

En 1971, Stallman pertenecía a un grupo de los Laboratorio de Inteligencia Artificial (AI Lab.) del MIT que compartía software desde varios años atrás. Se utilizó un sistema operativo de tiempo compartido llamado ITS (en inglés, Incompatible Timesharing System), diseñado y escrito por los programadores de planta del AI Lab en lenguaje Ensamblador para el Digital PDP-10 - (Procesador de Datos Programados Modelo 10), uno de los computadores más grandes de la época.

Cuando una persona de otra universidad o de cualquier empresa quería instalar y utilizar un programa, se le prestaba de buen grado. Si alguien utilizaba

1. Software libre para una sociedad libre, Stallman, Richard M. 2004. Recopilación y análisis

un programa desconocido e interesante, se le preguntaba por el código fuente para leerlo, modificarlo o “copiar” partes de él y montar un programa nuevo. Era la norma y la regla general para compartir y colaborar en el desarrollo de nuevo software. Esta situación cambió en 1981, la nueva empresa Symbolics contrató a casi todos los programadores del AI Lab, y los restantes miembros fueron incapaces de sobrevivir a esta situación. Luego el AI Lab compró un nuevo PDP 10 en 1982, sus administradores decidieron usar un sistema de digital de tiempo compartido no libre en lugar del ITS para instalarlo en la nueva máquina.

Los modernos computadores de la época, como el VAX o el 68020, contaban con su propio sistema operativo, pero ninguno utilizaba software libre. Había que firmar un acuerdo de confidencialidad incluso para obtener una copia ejecutable. Esto se fue generalizando.

Todo ello significaba que antes de poder utilizar un programa de computador, “se tenía que prometer no ayudar al vecino”. Quedaban así prohibidas las comunidades cooperativas y libres. Los contratos de software propietario establecieron la siguiente norma: “Si se comparte con el vecino, se convierte en un pirata. Si se quiere hacer algún cambio en el programa, se tendrá que rogar”.

El “software libre”, como las “sociedades libres”, tiene sus enemigos. Para las multinacionales productoras de software - comerciantes contemporáneos - en épocas de globalización y “libre mercado”, el “software libre” suena imposible, cuando mas es iluso. Nada, ni siquiera un vaso de agua, es libre (en inglés, de precio). ¿Cómo podrían ser “libres”, los conocimientos que gobiernan las máquinas, que a su vez dirigen el mundo?

¿Cómo podría una sociedad en su sano juicio, aspirar a semejante utopía de la libertad?

Una “sociedad libre” está gobernada por leyes. En toda sociedad libre hay límites que se ponen a la regulación legal: ninguna sociedad que mantenga sus leyes en secreto podría llamarse “libre”, mucho menos un Estado Social de Derecho. Ningún gobierno que esconda sus normas a los gobernados podría incluirse en una tradición de democracia, libertad y derecho. Los Acuerdos - el Derecho - gobiernan sólo cuando lo hacen a la vista de todos. Y el Derecho sólo está a la vista, cuando sus términos pueden ser establecidos y reconocidos por los gobernados y a su vez, ser acatados por sus gobernantes.

La idea de que el sistema social de derecho en torno al software propietario - un sistema que impide compartir o modificar el software - es antisocial, poco ético y sencillamente equivocado, puede sorprender a algunos. Pero ¿qué podemos decir acerca de un sistema que siembra la división y el egoísmo entre los usuarios y los abandona a la indefensión más absoluta en manos de los editores de software?

Cuando los editores de software hablan de “ejercer” sus “derechos” o de “acabar con la piratería”, lo que dicen es, de hecho, secundario. El verdadero mensaje de estas declaraciones se esconde en ciertas presunciones implícitas que dan por supuestas; creen que el público debe aceptarlas sin cuestionamiento. A analicemos algunas:

- Una suposición es que las empresas de software tienen el derecho natural e incuestionable a poseer el software y por ende, a detentar todo el poder sobre sus usuarios. El copyright no es una ley na-

tural, sino un monopolio artificial impuesto por el Estado que limita el derecho natural de los usuarios a copiar.

- Otra presunción implícita es que lo único importante en el software es la función que permite desempeñar, y no, los valores explícitos e implícitos que transmite y conlleva a los usuarios con su manera de “adquirirse” — y por lo tanto que, como usuarios de computadores, no deberíamos preocuparnos por el tipo de sociedad que se nos permite tener.
- Una tercera presunción es que no tendríamos de software de utilidad —o de un programa para realizar esta u otra tarea— si no cedemos el derecho de los usuarios sobre un programa a la empresa responsable del mismo. Esto resultaba convincente antes de que el movimiento del software libre demostrara que podíamos crear muchísimos programas, y muy útiles, sin necesidad de cadenas.

Si preferimos rechazar estas presunciones y analizamos estas cuestiones de acuerdo con los criterios morales y el sentido común del ciudadano de a pie, anteponiendo los usuarios a cualquier otra consideración, llegaremos a conclusiones muy diferentes. Los usuarios de computadores deberían ser libres para modificar los programas y ajustarlos a sus necesidades, libres para compartirlos, porque la cooperación con los demás consti-

tuye la base ética y moral de toda sociedad democrática y libre.

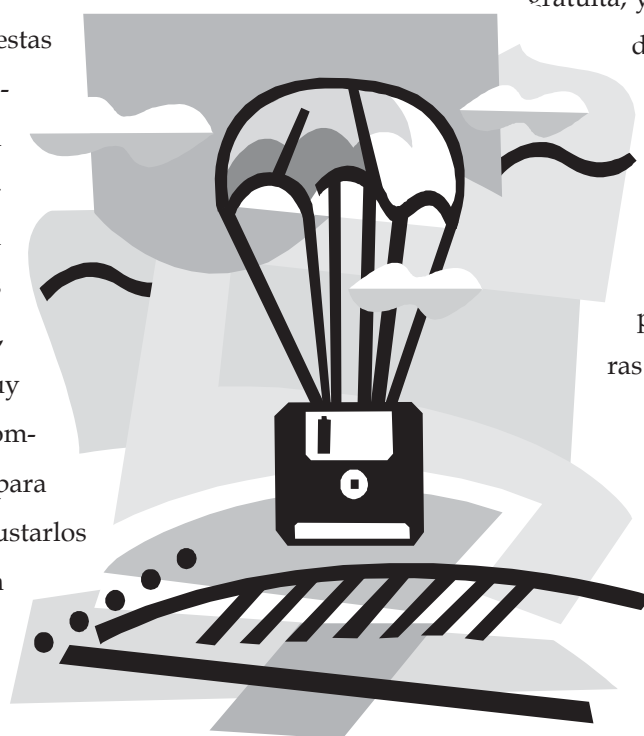
El término de “software libre” —para empezar, no tiene ninguna relación con el precio. Lo que nos interesa es la libertad., esta es la base para la definición de software libre.

Un programa es “software libre” siempre que, como usuario particular, se tenga:

- La libertad de ejecutar el programa para cualquier propósito.
- La libertad de modificar el programa para ajustarlo a las propias necesidades. (Para que se trate de una libertad efectiva en la práctica, se debe tener acceso al código fuente, dado que sin él la tarea de incorporar cambios en un programa es extremadamente difícil).
- La libertad de redistribuir copias, ya sea de forma gratuita, ya sea a cambio de un pago, de un precio.

- La libertad de distribuir versiones modificadas del programa, de tal forma que la comunidad pueda aprovechar las mejoras introducidas.

La libertad para vender copias es crucial: las colecciones de software libre a la venta en formato de CD-ROM o DVD son muy importantes para la

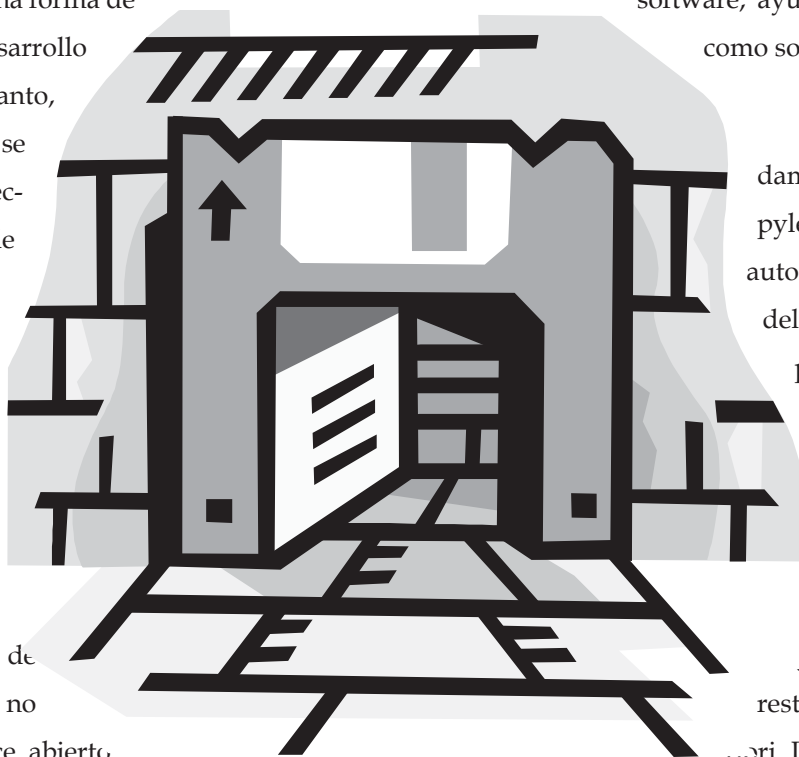


comunidad y venderlas es una forma de recaudar fondos para el desarrollo de software libre. Por lo tanto, cualquier programa que no se pueda incluir en estas colecciones no podrá calificarse de software libre. La filosofía del software libre rechaza una práctica empresarial concreta y muy generalizada, pero no rechaza el negocio en general.

Usando el poder del "copyright" - "derechos de autor" -, el "software libre" no sólo asegura que permanece abierto y susceptible de modificación, sino también que otro software, obtenido al incorporar y usar "software libre" - y que técnicamente se convierte en "obra derivada" - debe también, a su vez, ser "software libre".

Si uno usa y adapta un programa de software libre y distribuye públicamente esa versión adaptada, la versión distribuida debe ser tan libre como la versión de la que procede. Debe hacerse así, de lo contrario se estará infringiendo el "copyright". Se mantiene el anglicismo "copyright" frente al término jurídico de "derechos de autor", más correcto en español, no sólo por el uso amplio y extendido del término en inglés.

El mecanismo para este fin es un instrumento extraordinariamente inteligente llamado "copyleft" - "izquierdos de autor" - que se implementa a través de una licencia llamada GPL. "Copyleft" utiliza la ley de "copyright", pero dándole la vuelta para servir a un propósito opuesto al habitual: en lugar de privatizar el



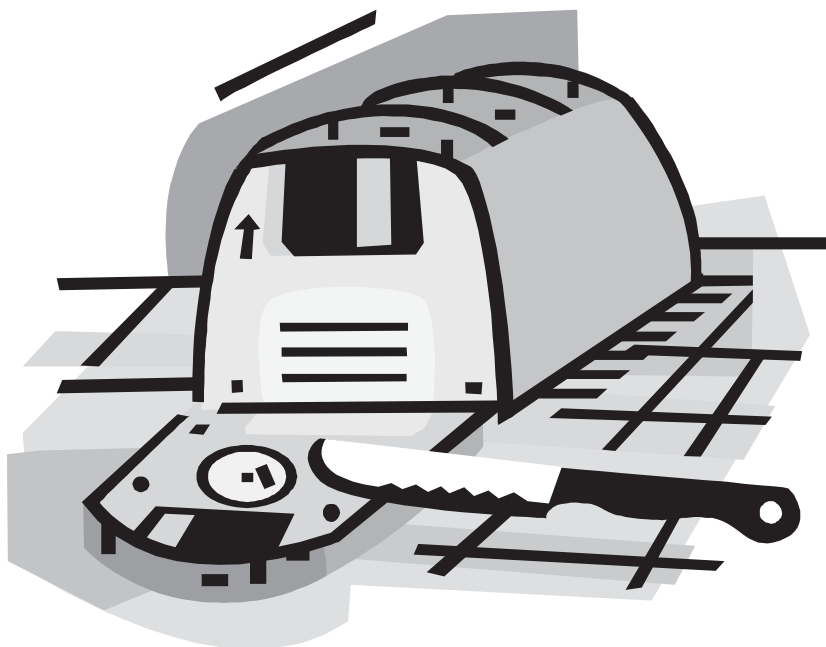
software, ayuda a preservarlo como software libre.

La idea fundamental del "copyleft" es que se autoriza la ejecución del programa, su copia, modificación y distribución de versiones modificadas, siempre que no se añada una clase de restricción a posteriori. De este modo, las

libertades cruciales que definen el "software libre" quedan garantizadas para cualquiera que posea una copia; estas libertades se convierten en derechos inalienables.

En 1985 se crea la Free Software Foundation, - FSF -, una organización sin ánimo de lucro dedicada al desarrollo de software libre. La FSF también distribuye copias del software, a lo que más tarde añadiría otros programas libres -no sólo de GNU- así como la venta de manuales libres. Por otra parte, en relación a las licencias GNU de la Free Software Foundation, se utiliza indistintamente tanto la traducción española, como Licencia Pública General (General Public License), como las siglas inglesas por las que normalmente son más conocidas, en este caso GPL o más correctamente GNU GPL.

La FSF acepta donaciones, pero la mayor parte de sus ingresos se dan al vender copias del software



libre y de otros servicios relacionados con éste. Actualmente, vende CD-ROM de códigos fuente, CD-ROM con los binarios, manuales cuidadosamente impresos –con total libertad para redistribuirlos y modificarlos– y Deluxe Distributions: colecciones enteras de software adaptadas a la plataforma elegida por el cliente.

La parte fundamental del software libre es la de tener un sistema operativo libre - no propietario - que fuera portable - se puede instalar en diferentes tipos de maquinas - y facilitar a los usuarios su migración - cambiarse al nuevo sistema operativo - pensando en el desarrollo futuro de los computadores; después de hacer unos análisis se vio la necesidad de que fuera compatible con Unix. El nombre de GNU fue elegido según una tradición de los programadores, como un acrónimo recursivo de "GNU's Not Unix" (en español, "GNU no es Unix"). El principal objetivo de GNU era ser software libre. Aun cuando GNU no entrañara ninguna ventaja técnica frente a Unix, sí tendría una ventaja social, al permitir que los usuarios cooperaran, y otra ética, al respetar su libertad.

En 1991, Linus Torvalds desarrolló un kernel compatible con Unix y lo llamó Linux. En el año 1992, la combinación de Linux con el incompleto sistema GNU, dio como resultado un sistema operativo libre. Gracias a Linux se puede ejecutar hoy una versión del sistema GNU. Esta versión se denomina: GNU/Linux, su composición fundamental es una combinación del sistema GNU con Linux como kernel.

4. El futuro

La FSF ha demostrado ser capaz de desarrollar una amplia gama de software libre. Esto no significa que sean invencibles e imparables. Existen diversos retos que plantean un futuro incierto para el software libre. Se nos exigirá un esfuerzo constante y mucha resistencia, a veces, durante muchos años. Se necesita la determinación que exhibe la gente cuando valora su libertad y la protege a toda costa.

La "Caja Negra" del Hardware: Los fabricantes de hardware tienden cada vez más a mantener en secreto las especificaciones del hardware. Esto dificulta enormemente la tarea de escribir drivers –controladores– libres que sean compatibles con el hardware nuevo. Hoy contamos con sistemas libres completos, pero no durarán mucho si no son compatibles con los computadores del mañana. Hay dos formas de enfrentarse a este problema. Los programadores pueden hacer ingeniería inversa para descubrir cómo crear programas compatibles con el hardware. El resto podemos elegir qué hardware será compatible con el software libre. A medida que aumente el número de usuarios de soft-

ware libre, el secretismo de estas especificaciones se convertirá en una política contraproducente.

Hacer ingeniería inversa es una labor colosal. ¿Contaremos con programadores lo bastante decididos para llevarla a cabo? Sí, siempre que les hayamos convencido de que el software libre es una cuestión de principios y de que los drivers no libres son intolerables. ¿Invertiremos dinero extra, e incluso tiempo extra, para poder utilizar drivers libres? Sí, siempre y cuando se generalice esta voluntad de recuperar nuestra libertad.

Las Bibliotecas NO Libres: que operan en un sistema operativo libre constituyen una trampa para los desarrolladores de software libre. Las atractivas funciones de la biblioteca son el cebo perfecto; al utilizar la biblioteca, se cae en la trampa, porque el programa no puede integrarse de forma útil en un sistema operativo libre. (Estrictamente hablando, se puede cargar el programa, pero éste no podría ejecutarse sin la biblioteca). Y, lo que es peor, en caso de popularizarse un programa que utilice una biblioteca propietaria, podría arrastrar a otros programadores desprevenidos hacia la misma trampa. ¿Cómo responder a la tentación que plantea la próxima biblioteca no libre? ¿Comprenderá la comunidad la necesidad de mantenerse alejada de cualquier trampa que se presente en el camino? ¿O renunciará a la libertad a cambio de comodidad y dar lugar así a un problema mucho mayor? Nuestro futuro depende de nuestra filosofía.

Patentes de Software: La amenaza más seria a la que se enfrenta el "software libre" procede de las patentes de software, que pueden introducir algoritmos y funciones fuera de su alcance al menos durante veinte años. Las patentes del algoritmo de compresión LZW

se aplicaron en 1983, y hasta hace poco se puede publicar software libre que produzca GIFs adecuadamente comprimidos. En 1998, se suspendió la distribución de un programa libre para producir archivos de audio MP3 comprimidos bajo amenaza de una demanda judicial por patente. Existen formas de abordar la cuestión de las patentes: buscar pruebas que demuestren la invalidez de una patente o buscar modos alternativos para realizar una tarea. Pero estos métodos funcionan sólo de vez en cuando; cuando fallan ambos, una patente puede obligar a todo el software libre a dejar de lado alguna función que desearían los usuarios.

La Documentación Libre: La mayor deficiencia de nuestros sistemas operativos no reside en el software, sino en la ausencia de buenos manuales libres para nuestros sistemas. La documentación es una parte esencial de cualquier paquete de software; un paquete importante de software libre sin un buen manual libre que lo acompañe constituye un lastre considerable. La documentación libre, al igual que el software, es una cuestión de libertad, no de precio. Los criterios para el manual libre son bastante parecidos a los del software libre: hay que darle a los usuarios ciertas libertades. Debe autorizarse la redistribución —incluida la venta comercial— en papel y on-line, de modo que el manual pueda acompañar a todas las copias del programa.

5. Terminología Adecuada

En la filosofía del "Software Libre", muchas veces se recurre a términos, palabras y conceptos de uso general no específico, los cuales conllevan a ideas ambiguas o distorsionadas - no siempre con buena intención - del ideario mismo de la FSF y el GNU, por lo tanto, hagamos algunas precisiones:

Comercial: Por favor no use “COMERCIAL” como sinónimo de “NO LIBRE”. Se estarían confundiendo dos asuntos completamente distintos. Un programa es comercial si se desarrolla como parte de una actividad empresarial. Un programa comercial puede ser libre o no libre, según su licencia. De la misma forma, un programa desarrollado por una escuela o un particular puede ser libre o no libre, según su licencia. Ambas cuestiones, qué clase de entidad desarrolló el programa y qué libertad tienen sus usuarios, son independientes. El software libre comercial es una contribución a nuestra comunidad, por lo que debemos promoverlo.

Creador: El término “CREADOR” aplicado a los autores, los equipara implícitamente a una deidad (“el creador”). Este término es utilizado por los editores para elevar la estatura moral de los autores por encima de la del ciudadano medio, justificando un mayor poder del copyright que los editores pueden ejercer en nombre de los autores.

Freeware: Por favor no use el término “freeware” como sinónimo de “software libre”. El término “freeware” fue utilizado con frecuencia durante la década de 1980 para hacer referencia a programas publicados sólo como ejecutables, con el código fuente no disponible. De la misma manera, si utiliza otro idioma diferente al inglés, por favor intente evitar palabras tomadas del inglés como “free software” o “freeware”. Intente utilizar palabras del propio idioma, por ejemplo:

Gestión de derechos digitales: El software para la gestión de derechos digitales está diseñado en realidad para imponer restricciones a los usuarios de computadores. El uso de la palabra “derechos”, en esta

expresión, responde a una propaganda pensada para que, sin darse cuenta, se considere la cuestión desde el punto de vista de los pocos que imponen las restricciones, mientras se ignora los derechos de los muchos a quienes las restricciones son impuestas. Son buenas alternativas: “Gestión de restricciones digitales” o “software con grilletes”.

Piratería: Los editores frecuentemente se refieren a la copia prohibida como “piratería”. De esta forma, expresan indirectamente que hacer copias ilegales es éticamente equivalente a atacar barcos en alta mar, y secuestrar y asesinar a la gente que viaja en ellos. Si no se cree que la copia ilegal sea como secuestrar y asesinar, se podría no usar la palabra “piratería” para describirla. Expresiones neutrales como “copia prohibida” o “copia no autorizada” pueden utilizarse en su lugar. Incluso algunos de nosotros podríamos preferir utilizar una expresión positiva tal como “compartir información con tu vecino”

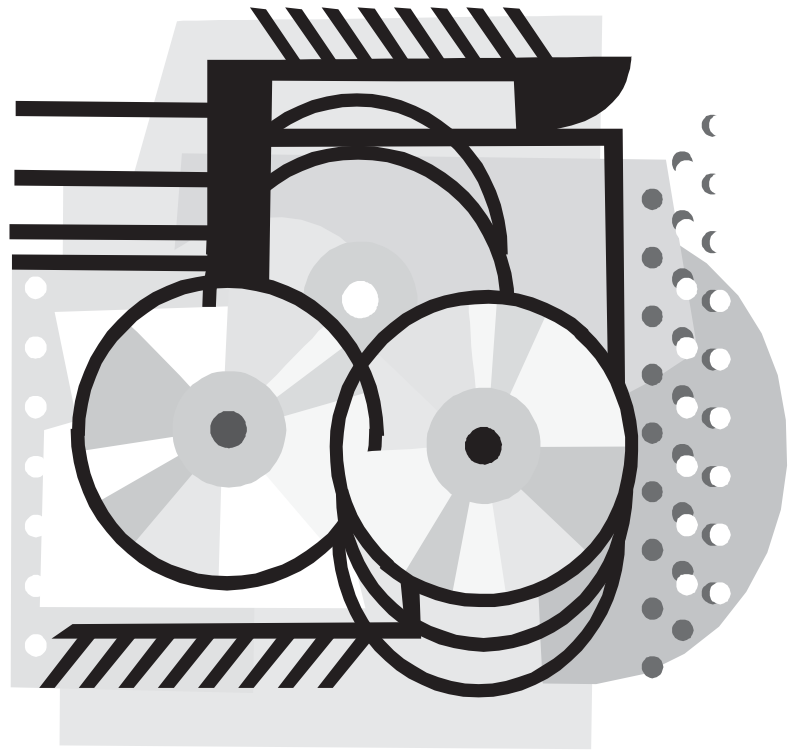
Propiedad intelectual: A los editores y a los abogados les gusta describir el “copyright” como “propiedad intelectual”. La expresión “propiedad intelectual” conlleva una presunción oculta: que la forma más normal de pensar respecto a la cuestión de la copia se basa en una analogía con los objetos físicos y en la idea que nos hacemos de ellos como propiedad. Pero esta analogía pasa por alto la diferencia crucial entre objetos materiales e información: la información puede ser copiada y compartida casi sin esfuerzo, mientras que los objetos materiales no. Fundar el pensamiento en esta analogía es ignorar esta diferencia.

Protección: A los abogados de los editores les encanta utilizar el término “protección” para describir el “copyright”. Esta palabra lleva implícita la idea de que

evita la destrucción o el sufrimiento; por lo tanto, impulsa a la gente a identificarse con el propietario y el editor, que se benefician del “copyright”, en lugar de identificarse con los usuarios que son restringidos por él. Es fácil evitar el término “protección” y utilizar expresiones neutrales en su lugar. Por ejemplo, en lugar de “la protección del copyright permanece vigente por un tiempo prolongado”, puedes decir, “el copyright permanece vigente por un tiempo prolongado”. Si se quiere criticar el copyright en lugar de apoyarlo, se puede emplear la expresión “restricciones del copyright”.

RAND: Los organismos de estandarización, que promulgan estándares restringidos de patentes que prohíben el software libre, suelen llevar a cabo una política para la obtención de licencias que requiere el pago de una tasa fija por cada copia hecha de un programa que cumpla esos estándares. A menudo se refieren a dichas licencias con el término “RAND”, que son las siglas de “razonable y no discriminatoria”. Este término no es más que un lavado de cara para un tipo de licencias de patentes que no son normalmente ni razonables ni no discriminatorias. Es cierto que estas licencias no discriminan a ninguna persona en particular, pero sí discriminan a la comunidad del software libre, lo que no las hace en absoluto razonable. Así que la mitad de “RAND” es engañosa y la otra mitad discriminatoria.

Robo: Los apologetas del copyright emplean con frecuencia palabras como “robo” y “hurto” para describir la violación del copyright. Al mismo tiempo, piden que consideremos el sistema legal como una au-



toridad ética: si copiar está prohibido, debe ser malo. De este modo, es pertinente mencionar que el sistema legal rechaza la idea de que la violación del copyright sea un “robo”. Los apologetas del copyright apelan a la autoridad, mientras tergiversan lo que la propia autoridad dice.

Software Gratuito: Si se quiere decir que un programa es software libre, por favor no diga que está disponible “de forma gratuita”. Esa expresión significa explícitamente que tiene un “precio cero”. El software libre es una cuestión de libertad, no de precio. Para evitar la confusión, puedes decir que el programa está disponible “como software libre”.

Software Regalado: Es erróneo utilizar el término “regalar” para referirse a “distribuir un programa como software libre”. Tiene el mismo problema que “software gratuito”: expresa que lo importante es el

precio, no la libertad. Una forma de evitar esta confusión es decir, “publicado como software libre”.

Vender Software: La expresión “vender software libre” es ambigua. Estrictamente hablando, intercambiar una copia de un programa libre por una cantidad de dinero es “vender”; pero la gente normalmente asocia el término “vender” con restricciones propietarias en el uso consecutivo del software. Puedes ser más preciso y evitar confusiones, diciendo “distribuir copias de un programa por una cantidad” o “imponer restricciones propietarias al uso de un programa”, dependiendo de lo que se quiera expresar.

6. Conclusión

Cuando nuestro mundo finalmente comprenda el poder y el peligro del “código”, cuando finalmente

vea que el “código”, como las leyes o como el gobierno, debe ser transparente para ser libre, comprenderemos la idea de un mundo donde la libertad y el conocimiento sobreviven al compilador.

Una sola persona o empresa NO puede retener algo con cuyo conocimiento la humanidad puede beneficiarse tremendamente. Ya que, el propósito de la ciencia y la tecnología es crear información útil para la humanidad, para ayudar a la gente a vivir mejor. Si prometemos que retendremos esa información – si la guardamos en secreto – entonces estaremos traicionando el objetivo fundamental de nuestra sociedad democrática, libre y en derecho.

El “software libre” asegura que el mundo gobernado por el código, debe seguir siendo tan “libre” como

lo era antes de la pretendida ley de software propietario: ley impuesta por unos pocos para su propio beneficio y en detrimento de los muchos usuarios de computador; que no hacen la ley – la padecen –, y pretender cambiar nuestra bien habida tradición de cooperación y libertad, al compartir en beneficio de la sociedad con nuestros semejantes, sin restricciones.

Por eso, hacer posible la libertad que la sociedad futura debe tener; no es una utopía, es una necesidad

