

Software Libre: introducción histórica

Revisión: 1.1

Jens Hardings, Alejandro Fuentes
(jens@hardings.cl, alejandro.fuentes@csol.cl)

4 de julio de 2003

1. Introducción

Esta sección ofrece una rápida vista histórica al desarrollo del software, a modo de introducir los conceptos de software libre (también llamado software de fuente abierta) y de dominio público. Se explica qué es el código fuente y el código objeto, así como la relación entre ambos para luego nombrar el dominio público y lo que se entiende bajo ese concepto.

En la segunda parte, en orden cercano al cronológico, se detallan algunos aspectos de cómo fueron surgiendo los conceptos de software propietario y software libre.

1.1. Código fuente y código objeto

Es importante conocer que por lo general, el código o programa que es escrito (y legible) por seres humanos no es el mismo código que se ejecuta en un computador¹. Para poder ejecutar un programa es necesario, por lo tanto, transformar el código fuente, que escriben los programadores, a algo que se pueda ejecutar en un procesador. Ese proceso se llama compilación y se hace por medio de un programa (compilador) especialmente diseñado para el lenguaje en el que fue escrito

¹la excepción es el código interpretado, el cual requiere a su vez de otro programa que sí debiera cumplir el requisito de ser código ejecutable

el código fuente. El resultado final de la compilación es el código objeto (también llamado código ejecutable), el cual es creado para un procesador específico².

Para poder ejecutar un programa, es necesario solamente el código objeto que es el que contiene las instrucciones para el procesador específico para el que fue compilado el programa. Sin embargo, el código fuente es la única forma de hacer modificaciones a un programa, ya que el proceso de compilación no es reversible³. Tampoco es posible crear código objeto para un procesador partir de código objeto creado para otro procesador, para ello es necesario tener acceso al código fuente.

1.2. Dominio Público y Propiedad Intelectual

El dominio público es el conjunto de bienes que cualquier persona tiene derecho a usar sin necesitar permisos ni pagar por el uso. En relación a bienes intangibles como lo son los protegidos por *propiedad intelectual*, el dominio público comprende todos los bienes que no están protegidos particularmente y forman un tesoro de información y recursos de libre disposición. La creatividad e innovación de una sociedad se basan en gran medida en el dominio público, el cual incluye la herencia de todas las generaciones anteriores. Todas las obras intelectuales, artísticas y otras que caen en categorías protegidas por *propiedad intelectual* necesitan de un dominio público para poder ser creadas.

Para una sociedad, es positivo contar con un dominio público lo más extenso posible, dado que cualquiera puede valerse de esta fuente para crear nuevas ideas y obras basándose en el conocimiento acumulado. Es por ello que es natural para una sociedad intentar maximizar el dominio público que tiene a su alcance. La propiedad intelectual es un conjunto de herramientas legales que si bien permiten a ciertos individuos abstenerse de agregar sus creaciones (las cuales, recordemos, se basan en gran medida en el dominio público existente) al dominio público, al menos temporalmente. Esto se hace con el fin de que en el largo plazo la cantidad de innovaciones creativas que pasan al dominio público sean mayores que en una situación sin esta protección. En otras palabras, parece razonable abstenerse

²o para una familia de procesadores, si se ignoran los beneficios de aprovechar las diferencias entre distintos procesadores de la misma familia

³Existe un proceso conocido como ingeniería reversa, pero éste es un trabajo extremadamente laborioso, inexacto y se debe recurrir a mucho estudio del comportamiento de un programa más que solamente inferir el código fuente a partir del código objeto. Por lo general, el costo de obtener código útil a partir de ingeniería reversa es comparable, si no mayor, al de escribir el código originalmente y además hay restricciones legales (contratos y otras) que prohíben esas prácticas en muchos casos.

de tener ciertos bienes en el dominio público de inmediato y esperar un tiempo, porque de lo contrario es probable que esa obra no hubiese sido generada siquiera, mucho menos podría haber pasado al dominio público.

2. Software y Libertades: el inicio

En un inicio, el software se distribuía junto con los pocos computadores que existían en el mundo, y tanto los clientes como los desarrolladores de estos computadores se encargaban de arreglar errores y agregar funcionalidades necesarias. Por ejemplo IBM entregaba código fuente a través del proyecto SHARE⁴, mientras que DEC hacía lo propio con la asociación DECUS⁵. El software no se compraba ni se vendía, simplemente se entregaba, pasando al dominio público (ver 1.2), y el verdadero negocio estaba en los costosos computadores, el software solamente era algo necesario para hacerlos funcionar y así darles un valor útil.

Cuando Bell Labs puso el código fuente de Unix a disposición de diversas universidades⁶, muchas mejoras y adiciones resultaron como consecuencia. La Berkeley Software Distribution (BSD, la primera versión creada en 1977), contenía los cambios a Unix y aplicaciones que se distribuían por el costo de una cinta de datos más el transporte. De esta misma manera también se agregó soporte de TCP/IP (el protocolo para comunicarse en Internet) a Unix. Incluso cuando AT&T (Bell Labs era en parte propiedad de AT&T), a través de Unix System Laboratories, comenzó a cobrar licencias por Unix, muchos pagaban esas licencias pero preferían usar el código de BSD⁷ el cual se distribuía en forma casi gratuita, pero sobre todo, libremente (cualquiera podía crear nuevas copias y distribuir las sin problemas).

Incluso en otro ámbito, relacionado con el Hardware, la publicación de los detalles del *Personal Computer* diseñado por IBM fue clave en su carácter revolucionario, al permitir a cualquier empresa a producir partes y piezas para esta nueva raza de computadores. Este hecho le permitió al computador personal un avasallador triunfo sobre otros sistemas, algunos de calidad muy superior. Si bien

⁴El proyecto SHARE, que es independiente de IBM pero cuenta con su apoyo, ha funcionado desde 1955 a la fecha y hoy en día se encuentra bajo <http://www.share.org/>

⁵Hoy en día, Encompass (<http://www.encompass.org/>) es el sucesor de DECUS-US

⁶El hecho que AT&T, principal propietario de Bell Labs, tuviese prohibición de participar en mercados fuera de su monopolio telefónico influyó en este hecho

⁷Este código proporcionaba, entre otras cosas, el Fast File System y la posibilidad de usar TCP/IP, atributos no disponibles en el *System V*, la versión oficial de USL.

claramente IBM podría haber lucrado mucho más optando por una decisión distinta⁸, es evidente que este hecho causó que el computador se volviera alcanzable para las masas y se convirtiera en un artículo de consumo (*commodity*).

3. Surgimiento de Software Propietario

Con el paso de la historia⁹, se dejó de lado la práctica de entregar el código fuente cuando se solicitaba, o bien se exigía la firma de un acuerdo de no divulgación. Este fue además el comienzo de la piratería de software, concepto que se inventó para intentar concientizar y dar una imagen profana a un problema que gozaba de la indiferencia de la mayoría.

La consecuencia de cerrar el código, junto con la poca cooperación que en parte resultó de ese hecho, hizo que proliferaran versiones derivadas de Unix, todas incompatibles entre sí. Las consecuencias no fueron necesariamente negativas para los proveedores de estos “sabores” de Unix: los clientes estaban cautivos, puesto que no podían cambiar fácilmente a otro proveedor de Unix, muchas veces un cambio así implicaba incluso el cambio de todo el hardware instalado, o bien era necesario manejar instalaciones híbridas y vivir con dos o más sistemas medianamente independientes.

Actualmente, una buena parte del software o programas de computadores que se usa diariamente en muchos lugares corresponde a software propietario: es software que no se compra, solamente se licencia el derecho de uso bajo ciertas condiciones y los costos de licencias dependen de variables como:

- la cantidad de computadores en los cuales se instala el software
- la cantidad de clientes que acceden un servidor
- la capacidad de cómputo de cada computador en que se instala el software

⁸No queda claro que la decisión de publicar el diseño del computador personal haya sido en realidad intencional. Más bien parece ser la conjunción de la inspiración revolucionaria de algunos empleados de IBM, casualidad y error.

⁹Hubo varios hechos significativos entre 1969 y 1974: IBM se ve obligada por un juicio anti-monopolios a fines de los '60 a separar el cobro por hardware, software y servicios, que hasta ese momento solamente se ofrecían en paquetes. El otro evento, conocido como el “software flap”, fue el que terminó por pavimentar el camino que seguirían futuros programas de computadores y está dado por una “Carta abierta a hobbyists” escrita por Bill Gates en 1976, en la cual se quejaba de que la mayoría de los usuarios de *Altair Basic* no habían pagado por su copia.

- la cantidad de usuarios de cada computador
- la duración de validez de la licencia

Comparado con el software libre, este tipo de software deja al usuario a merced del desarrollador de software. A cambio se recibe la promesa infundada (cualquier licencia de software se encarga antes que nada de quitar toda responsabilidad de parte del desarrollador, ver Hardings and Fuentes (2003)) de recibir soporte y servicios cuando se requieran. Si por ejemplo hay un error en el programa, el único capaz y legalmente autorizado a arreglarlo es el desarrollador. En el caso de que haya una demora en arreglar el error o cualquier otro problema, no existe posibilidad de buscar ayuda en otra parte.

4. Software (intencionalmente) Libre

Si bien en un inicio todo el software era libre, esto probablemente no era más que un hecho determinado por las circunstancias¹⁰. No existía una intención explícita ni consciente de que el código fuente fuera puesto a disposición del que lo necesitara y no se entregaban derechos o libertades de manera explícita.

Esto cambió en 1984, cuando Stallman crea la Free Software Foundation, su proyecto GNU y sobre todo la GNU General Public License (GNU GPL). Con esta licencia, Stallman acuña el concepto de Copyleft, el cual hace uso del derecho de autor (copyright) para conseguir su objetivo. Este objetivo es evitar que cualquier persona o institución pueda a futuro usar el código fuente licenciado en productos para luego distribuir éstos sin retribuir a la comunidad con sus modificaciones.

Paradójicamente, la GPL es por un lado más restrictiva que el dominio público (ver 1.2), en el sentido de que prohíbe ciertos comportamientos que serían válidos para obras en el dominio público. Por otro lado, la GPL otorga mayores beneficios al público general que el dominio público, puesto que se garantiza que nadie puede hacer uso del material bajo esa licencia sin otorgarle las mismas libertades a todo aquél a quien le distribuye versiones modificadas o textuales, así como obras derivadas.

En 1999, cuando Netscape decide distribuir el código fuente de su *Navigator*, surge una nueva iniciativa, *Open Source Initiative*, la cual evita el estilo confrontacional que caracteriza a Stallman para centrarse en puntos más pragmáticos del

¹⁰AT&T puso el código a disposición porque en ese tiempo no podía pensar en usar comercialmente el producto Unix, muchas de las universidades que generaban código encontraban natural publicarlo, siguiendo la tradición del trabajo científico, etc.

desarrollo de software libre. Según Stallman, la diferencia entre “Free Software” (software libre) y “Open Source Software” (software de fuente abierta) es que la primera representa una comunidad con fines políticos y principios básicos no transables, mientras que la segunda es una forma de desarrollar software independiente de ideales. En la práctica, proyectos de software libre y de software de fuente abierta no se pueden diferenciar sin entrar en detalles de intención y opinión personal de los impulsores y colaboradores del proyecto.

En términos simples, se puede decir lo siguiente sobre Software Libre:

- Software libre se refiere a programas computacionales sobre los cuales se permite explícitamente su utilización, copia, modificación y redistribución (tanto de copias textuales como modificadas), así como aprender de su funcionamiento.
- No se existe discriminación de ningún tipo sobre las libertades otorgadas.
- Un requisito para poder aprender cómo funciona un programa o realizar modificaciones, es la puesta a disposición del código fuente (ver 1.1), el cual no es necesario para simplemente usar el programa.
- Como consecuencia de la libertad de copiar el software y a que no se pagan licencias por cada copia, el precio tiende a equivaler al costo de la copia (prácticamente 0). (No así el costo, que incluye aspectos como adaptar el software a necesidades específicas, aprender a usarlo, instalación, etc.)
- Diariamente usamos software libre sin darnos cuenta: al enviar un mail, visitar sitios web o al recibir y emitir facturas digitales. Incluso toda la tecnología sobre la cual funciona Internet fue desarrollada siguiendo el mismo principio que inspira al software libre.
- Dado que se pueden usar desarrollos hechos por otros, es mucho más fácil crear innovaciones y por lo tanto la sociedad en suma gasta menos recursos en producir una y otra vez lo que ya se ha hecho antes, mejorando la eficiencia.

El software libre no es un movimiento homogéneo y sus integrantes, si se pueden llamar así, por lo general no comparten metas en común. Por ello, en general existe más de un proyecto de software libre en un mismo área, pero hay varios proyectos que se han consolidado en sus respectivos nichos. En particular, es difícil que pase un día sin que usemos software libre en alguna medida, incluso

sin darnos cuenta. Ejemplos en donde existe un uso mayoritariamente de software libre son:

- Apache httpd: servidor web que es usado en más del 60 % de los servidores web en el mundo, de acuerdo a Netcraft¹¹.
- servidores de mail: *sendmail* es usado en el 42 % de los servidores de mail, y *qmail* en el 17 %¹². Ambas aplicaciones son software libre, por lo que el 59 % de los servidores de mail en Internet funcionan gracias a software libre.
- T_EX: procesador de textos con énfasis en fórmulas matemáticas y buen manejo de bibliografías, es usado en el 90 % de las revistas académicas y libros especializados en matemáticas y física¹³.
- bind: servidor DNS, el servicio que se encarga de traducir nombres de dominio a números IP y por lo tanto permite la existencia de Internet como la conocemos hoy en día. La totalidad de los servidores raíz (que son indispensables para que funcione todo el resto de los servidores de nombre) funcionaban con Bind hasta comienzos de 2003, momento en que se comenzó a usar NSD (que también es software libre) en uno de ellos¹⁴. Se estima que un 85 % a 95 % de los servidores de nombre usan bind para proveer sus servicios¹⁵.

Referencias

Grassmuck, V. (2002). *Freie Software – Zwischen Privat- und Gemeineigentum*. Bundeszentrale für politische Bildung, Bonn, Germany.

Hardings, J. and Fuentes, A. (2003). Discusiones sobre Aspectos Legales en torno a Software Libre y Formatos Digitales. <http://www.hardings.cl/publications/hardings2003legales.pdf>.

Raymond, E. S. (2001). *The Cathedral & The Bazaar*. O'Reilly & Associates.

¹¹Fuente: <http://www.netcraft.com/>

¹²Fuente: <http://cr.yip.to/surveys/smtpsoftware6.txt>

¹³Fuente: <http://lpf.ai.mit.edu/Patents/knuth-to-pto.txt>

¹⁴ <http://www.ripe.net/ripe/mail-archives/dns-wg/2003/msg00044.html>

¹⁵Fuente: http://www.nominum.com/content/documents/Foundation_wp.pdf,
<http://www.isi.edu/~bmanning/in-addr-versions.html>