Sofware Libre versus Software Propietario

Javier Smaldone (`javier@dc.exa.unrc.edu.ar`)

1. Conceptos preliminares

En un sentido restringido, el software es el conjunto de programas informáticos que al ser ejecutados por una computadora (o un conjunto de ellas), lleva a cabo el procesamiento, transmisión o almacenamiento de información.

Para su desarrollo, el programador escribe sentencias en algún *lenguaje de programación*. La característica común de este tipo de lenguajes es que tratan de asemejarse al lenguaje natural para facilitar la escritura y comprensión de los programas. El producto resultante se denomina *"código fuente"* del programa.

En una etapa posterior, y utilizando a tal efecto otro programa llamado *"compilador"*, se traduce el *código fuente* al lenguaje que es capaz de manipular la computadora que luego deberá ejecutarlo. El resultado de este proceso es el denominado "programa ejecutable" o *"código objeto"*.

En un desarrollo de mediana envergadura, la complejidad del *programa ejecutable* es tal que es virtualmente imposible su análisis por parte de un ser humano, razón por la cual para realizar correcciones, extensiones o cualquier tipo de modificación sobre el programa, se hace indispensable recurrir al *código fuente*, realizar allí los cambios requeridos y luego volver a compilarlo, generando un nuevo programa ejecutable.

La figura 1 muestra un esquema del proceso de creación de un programa.

2. El modelo tradicional de desarrollo y distribución

En esta sección analizaremos la forma tradicional de distribución y comercialización de software a gran escala, esto es, de aquellos productos de amplia difusión tales como sistemas operativos, procesadores de texto, bases de datos, compiladores, etc.

Cuando una empresa productora de software distribuye un producto de este tipo, solamente entrega al comprador una copia del programa ejecutable, junto con la autorización de ejecutar dicho programa en un número determinado de computadoras.

En el contrato que suscriben ambas partes, comúnmente denominado *"licencia"* del producto, queda expresado claramente que lo que el cliente adquiere es simplemente la facultad de utilizar dicho programa en determinada cantidad de computadoras (dependiendo del monto que haya abonado). En este sentido, la licencia deja en claro que el programa sigue siendo propiedad de la empresa productora del mismo y que el usuario no está facultado a realizar ningún cambio en él (lo que técnicamente sería muy difícil, por lo expuesto anteriormente) ni tampoco a analizarlo para determinar como realiza sus funciones. Entre otras cosas, el usuario también tiene vedada la difusión de cualquier fallo o error que pudiese descubrir en el programa. Un punto que es particularmente llamativo es que, en la mayoría de los casos, el comprador no puede transferir la titularidad de la licencia a un tercero.

Por otra parte, la empresa productora deslinda toda responsabilidad respecto a las pérdidas que pudiera ocasionar para el comprador el uso del software en cuestión. Esto incluye los daños por fallas del producto, debidas a errores del mismo. En este sentido, la licencia aclara que el vendedor no garantiza la aplicabilidad del producto a ningún fin determinado.

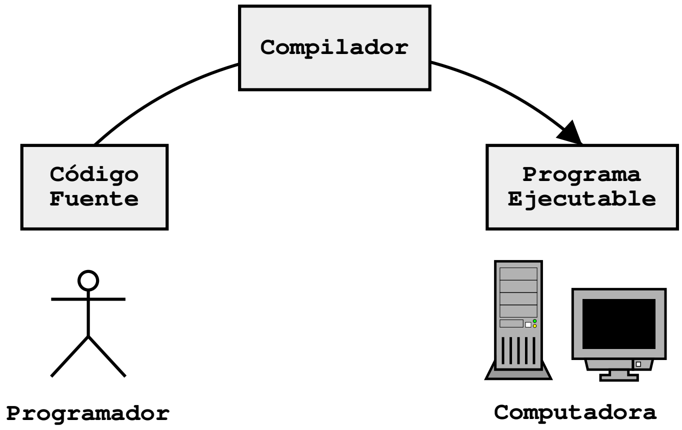

Figura 1: Proceso de creación de un programa

3. Consecuencias del modelo tradicional para el usuario

3.1. Dependencia de un proveedor

Como dijimos anteriormente, la corrección de errores o el agregado de nuevas funciones en un programa solamente puede hacerse si se dispone del código fuente. Es claro que, al ser la empresa proveedora la única que dispone de dicho código, sólo esta puede atender a los requerimientos de un cliente insatisfecho con el producto del cual ha adquirido una licencia de uso.

Esto pone al usuario en una clara situación de dependencia del proveedor. Si el programa presenta algún defecto, éste debe aceptar las condiciones de la empresa productora del programa (en el supuesto caso de que dicha empresa reconozca el error y acceda a repararlo).

3.2. Falta de garantías

Al no responsabilizarse la empresa productora del programa por los defectos que este pudiera presentar, ni garantizar su aplicabilidad para ningún fin determinado, el usuario queda imposibilitado de realizar ningún tipo de acción contra el proveedor ante la eventualidad de que dichos errores o defectos le ocasionen perjuicios, ya sea por indisponibilidad de sus sistemas informáticos, pérdida de información, pérdida de la confidencialidad de sus datos, etc.

El usuario ni siquiera puede reclamar por el cumplimiento de lo que el proveedor dice en sus anuncios publicitarios o en los mismos manuales del producto acerca de sus características.

Cabe recordar en este punto las cuantiosas pérdidas que se producen cada año por ataques de virus informáticos.

3.3. Falta de soporte

En la mayoría de los casos, los programas propietarios de amplia difusión son producidos por empresas transnacionales, que cuentan con representantes o distribuidores en nuestro país. Uno de los argumentos más fuertes esgrimidos por dichas empresas a la hora de justificar el alto costo de las licencias, es la disponibilidad de soporte técnico especializado. Dicho servicio supondría un respaldo hacia el usuario, de cara a la solución de posibles problemas que pudieran presentarse en el uso del producto adquirido por este último.

La realidad nos muestra que en la mayoría de los casos, todo lo que el usuario puede esperar del servicio de soporte técnico son instrucciones de instalación, configuración y uso. La reparación de errores detectados en el programa por lo general no está al alcance de quien brinda el soporte: una subsidiaria o un asociado a la empresa productora del software, que no dispone del código fuente del programa. Aún en el caso de poder reportar el fallo a los encargados del desarrollo del producto, en muchas oportunidades la empresa no reconocerá que se trata de un defecto del programa, ya que esto puede afectar a la imagen comercial del producto. Recordando nuevamente el caso de los virus informáticos, este problema se acrecienta año tras año debido a la resistencia de las empresas a reconocer y reparar los errores que posibilitan el accionar de este tipo de programas.

4. El modelo de desarrollo y distribución del Software Libre

El movimiento del Software Libre tuvo su origen en el mundo académico. Desde hace más de treinta años, numerosos programadores de distintas universidades han desarrollado herramientas de forma cooperativa y abierta, intercambiando libremente su código fuente. De esta manera se ha logrado construir productos de software de gran envergadura y excelentes cualidades técnicas.

Con el crecimiento de Internet, dicho movimiento ha sumado adeptos en todo el mundo, trascendiendo el ámbito académico. En la última década han surgido empresas que basan su negocio en el desarrollo, la distribución y el soporte de Software Libre. En la actualidad numerosas empresas que tradicionalmente han utilizado el modelo propietario para el desarrollo y la comercialización de sus productos, comienzan ha incorporar productos de software libre. A modo de ejemplo podemos citar a IBM, que basa todos sus productos orientados a la web en el servidor libre *Apache*.

4.1. ¿Qué se considera “Software Libre”?

Un programa puede considerarse *“Software Libre”* si de distribuye bajo una licencia que garantice a quien lo recibe las siguientes libertades:

1. Libertad de ejecutar el programa, con cualquier propósito.
2. Libertad de estudiar cómo funciona el programa y de adaptarlo a sus necesidades.
3. Libertad de redistribuir copias del programa.
4. Libertad de mejorar el programa y redistribuir dichas modificaciones.

Éstos últimos dos puntos deben tener una restricción: Si redistribuye el programa, modificado o no, debe hacerse bajo las mismas condiciones. Esta cláusula es la que garantiza la libertad del programa, impidiendo que alguien tome un desarrollo libre y lo transforme (o incluya) en un desarrollo propietario, limitando alguna de las libertades citadas.

Nótese no se menciona (y por ende, no se limita) la posibilidad de que quien desarrolla o distribuye un programa libre cobre por ello un monto de dinero.

4.2. El negocio basado en Software Libre

Gran parte del Software Libre existente en la actualidad ha sido desarrollado por programadores durante su tiempo libre, o como parte de sus actividades académicas. Este hecho ha sido utilizado en reiteradas oportunidades por sus detractores para calificar al software libre de “amateur”, carente de soporte y, por ende, no apto para el uso en aplicaciones reales.

Esta situación ha comenzado ha cambiar desde hace unos años debido principalmente a tres factores:

1. Grupos de desarrollo inicialmente informales o poco organizados, se han constituido en empresas debido al éxito de sus productos libres.
2. Algunas empresas productoras o comercializadoras de software han comenzado a cambiar sus políticas de desarrollo y distribución, utilizando licencias libres para sus productos.
3. Otras empresas han tomado desarrollos libres (producidos por programadores independientes) para utilizarlos como base de sus soluciones, invirtiendo en su mejoramiento y brindando soporte a sus usuarios.

4.2.1. ¿Por qué una empresa productora de software elegiría el modelo de Software Libre?

Básicamente son tres razones principales por las cuales una empresa podría inclinarse por este modelo:

1. Al optar por el modelo de desarrollo y distribución libre, la empresa queda habilitada a utilizar la gran cantidad de herramientas libres disponibles en la actualidad. Esto no sólo implica la ejecución de dichas herramientas, sino también la modificación de las mismas para adaptarlas a casos particulares y la exploración de sus mecanismos de funcionamiento para luego reutilizarlos en futuros desarrollos. De esta forma se obtiene una ventaja significativa respecto de aquellas empresas que basan su negocio en el modelo propietario o cerrado, que no pueden utilizar esta base de herramientas y conocimiento.
2. El liberar un programa facilita enormemente su distribución y publicidad. De esta manera no es necesario invertir enormes sumas de dinero en campañas publicitarias y en marketing para poder competir con productos establecidos en el mercado.
3. Si el producto en cuestión tiene suficientes méritos técnicos, con seguridad despertará el interés de un gran número de desarrolladores, usuarios y otras empresas en todo el mundo, lo cuales comenzarán a contribuir en su desarrollo, extensión y depuración. Muchos son los casos en que pequeños emprendimientos han engendrado productos de gran nivel técnico y de una envergadura impensada por sus creadores originales. Basta citar a modo de ejemplo productos como el sistema operativo *Linux*, el servidor web *Apache*, el manejador de base de datos *MySQL*, entre otros.

Con respecto al primer punto, una muestra de la importante base de Software Libre existente en la actualidad puede encontrarse en el *Proyecto Debian*, una recopilación de más de 2.800 paquetes de programas libres. Un análisis de dicha distribución, usando *COCOMO* (un modelo ampliamente utilizado en la industria para calcular el costo de desarrollo de software) arroja los siguientes resultados:

- Esfuerzo estimado: 171.141 hombres-mes (14.261 hombres-año)
- Costo estimado del desarrollo: 1.848.225.000 dólares

Esto indica, por ejemplo, que un equipo de desarrollo integrado por 400 personas demoraría 35 años en desarrollar todo el software incluido en Debian, con un costo de 1.800 millones de dólares.

Aunque como ya dijimos, Debian incluye sólo 2.800 programas, este ejemplo basta para demostrar la enorme base de productos y de conocimiento que puede utilizar quien desarrolle Software Libre.

4.2.2. El Software Libre y la propiedad intelectual

Muchos objetan, en contra del modelo de distribución libre, que el hecho de proveer el código fuente de los programas, autorizando su uso y redistribución, implica la pérdida de la propiedad intelectual. Esto posibilitaría a una empresa de mayor envergadura el tomar como propio dicho desarrollo, relegando a sus autores originales.

A través de los treinta años de existencia de Software Libre, nunca se ha dado una situación como esta. Por el contrario, este tipo de problemas se ha planteado en innumerables oportunidades (muchas de las cuales han llegado a la justicia) en el mundo del Software Propietario.

La razón por la cual los desarrolladores de un programa libre no pierden el control del mismo es muy simple: a sus usuarios y al resto de los programadores que contribuyen con el mismo no les conviene. Los autores originales son el factor aglutinante del proyecto. Ellos son quienes dirigen el rumbo del desarrollo, quienes recopilan los reportes de errores y los requerimientos de los usuarios, ellos reunen los aportes y contribuciones de los demás desarrolladores. De esta forma, si la actitud de los líderes del proyecto es la adecuada, sus usuarios y colaboradores seguirán manteniéndolos como referentes.

En virtud de esto, a una empresa que le interese incorporar dicho programa dentro de su oferta de productos le convendrá colaborar con el grupo de desarrollo existente (liderado por sus autores originales), antes que iniciar un proyecto paralelo.

Por el contrario, si los desarrolladores iniciales por algún motivo pierden interés en el producto, la disponibilidad del código fuente y de la documentación del mismo posibilitará la creación de nuevos grupos

Empresa	Ingresos	**G**anancias	% de ganancia	**E**mpleados	**G/E**	**E/G**
IBM	$ 81.667 M	$ 6.328 M	7 %	290.000	$ 21.820	45,83
Oracle	$ 7.143 M	$ 955 M	13 %	40.000	$ 23.875	41,88
Microsoft	$ 20.000 M	$ 8.000 M	40 %	29.000	$ 275.862	3,63

Cuadro 1: Algunas cifras sobre el negocio del software

que continuarán con la evolución del mismo y darán respuesta a los usuarios existentes. Esto marca otra gran diferencia con respecto a los programas propietarios, en donde la continuidad de un producto depende de la conveniencia y las condiciones económico-financieras de la empresa desarrolladora.

4.2.3. Acerca de las garantías y el soporte

Como ya dijimos, en general las empresas productoras de Software Propietario no ofrecen ningún tipo de garantía por sus productos. Además hemos marcado las deficiencias en materia de soporte técnico, brindado muchas veces por quienes no tienen la posibilidad real de solucionar problemas.

En el caso del Software Libre, la situación es radicalmente distinta. Aunque los creadores originales en general tampoco ofrecen ninguna garantía por el producto, la misma (junto con el soporte) puede ser asumida por una empresa que elija brindar dicho servicio a sus clientes. Dicha situación sería posible ya que se dispondría del código fuente del programa en cuestión, junto con el derecho de efectuar las modificaciones que el cliente requiriera para la satisfacción de sus necesidades, con lo cual estaría en posición de ofrecer un servicio garantía y asistencia técnica de gran calidad.

4.3. El Software Libre como modelo basado en servicios

El Software Libre promueve un modelo de negocio basado en servicios, en contraposición al modelo basado en licencias impulsado por las empresas de desarrollo y distribución de Software Propietario.

El cuadro 1 muestra las cifras de tres empresas líderes en el mercado.

IBM es una empresa que comercializa principalmente hardware. Si bien posee una amplia oferta de productos de software, no es éste el eje principal de su actividad comercial. *Oracle* es una empresa orientada exclusivamente al mercado de los manejadores de base de datos. *Microsoft* es exclusivamente una empresa de producción y comercialización de software, abarcando prácticamente todas las áreas: computación personal, herramientas desarrollo, bases de datos y software de base (sistemas operativos).

En el caso de *Microsoft*, las cifras muestran una clara desproporción en la relación entre ganancias y la cantidad de empleados. La columna **G/E** nos dice que por cada empleado, la empresa tiene una ganancia anual de 275.000 dólares (cifra que ronda los 20.000 dólares en los otros dos casos). La última columna de la tabla puede interpretarse en el sentido de cuántos empleados pone a disposición la empresa a un cliente que le signifique un millón de dólares en ganancias.

Estas cifras ponen en clara evidencia que la justificación del pago de altas sumas de dinero en concepto de “licencias de uso” de software, por el soporte técnico y los servicios que el proveedor brinda al cliente es inválida.

El Software Libre plantea una modalidad diferente: un modelo basando en servicios. El cliente no debe pagar por usar el software (lo que al productor del mismo no le ocasiona ningún gasto adicional), sino por los servicios de asistencia técnica, de capacitación y por la implementación de nuevas características y la corrección de errores o defectos.

De esta forma se ofrece al usuario un trato más justo teniendo en cuenta que, además, el código fuente del programa no está bajo el dominio absoluto del productor original del mismo, pudiendo otra empresa ofrecer los mismos servicios y fomentándose de esta manera la libre competencia.

5. Conclusiones

En el presente artículo han sido expuestas las ventajas del modelo de Software Libre en contraposición con el modelo propietario. Los efectos negativos del software propietario quedan de manifiesto teniendo en cuenta la situación actual, en la que algunas grandes corporaciones monopolizan gran parte del mercado informático mundial y el costo de los programas (y costos colaterales), ascienden año tras año.

Por contraparte, el Software Libre ha dejado de ser ya una promesa, debido a tres décadas de constante desarrollo y al reciente aporte de algunas grandes empresas, otrora comprometidas con el modelo propietario, que han visto la potencialidad de la nueva forma de negocio que se plantea.

Referencias

[1] *"Trampa en el Ciberespacio"*. Roberto Di Cosmo.
`http://www.pps.jussieu.fr/~dicosmo/Piege/trampas/`

[2] *"Por qué el software no debe tener propietarios"*. Richard Stallman.
`http://www.gnu.org/philosophy/why-free.es.html`

[3] *"Vender Software Libre"*. Richard Stallman.
`http://www.gnu.org/philosophy/selling.es.html`

[4] *"Proposición para el uso de Tecnologías Abiertas en el Estado"*.
`http://www.grulic.org.ar/proposicion/index.html.es`

[5] *"La Encrucijada Digital"*. Marcelo Baldi, Fernando Cuenca, Daniel Moisset.
`http://www.grulic.org.ar/proposicion/doc/encrucijada.html.es`

[6] *"Razones por las que el Estado debe usar Software Libre"*. Federico Heinz.
`http://www.grulic.org.ar/proposicion/doc/razones.html`

[7] *"Software Libre en la enseñanza de la Informática"*. Maximiliano Eguaras, Javier Smaldone.
`http://www.smaldone.com.ar/opinion/sl_informatica.html`

[8] *"Contando patatas: el tamaño de Debian 2.2"*. Jesús M. González-Barahona, Miguel A. Ortuño, Pedro de las Heras, José Centeno, Vicente Matellán.
`http://www.ati.es/novatica/2001/154/154-30.pdf`

Copyright