

## Implicaciones y modelos de negocio



Jesús M. González Barahona  
Profesor en la Universidad Rey Juan Carlos

### Implicaciones del trabajo con software libre

Las características del software libre y las posibilidades que ofrece son consecuencias de las libertades descritas en el apartado anterior. Por ejemplo, la libertad de redistribución proporciona en la práctica una canal de distribución de gran eficiencia económica y de muy bajo coste para el productor. Las posibilidades de modificación y de redistribución de las modificaciones facilitan la evolución y mejora técnica de los programas. En general, de la aplicación de todas las libertades simultáneamente se deducen importantes sinergias, que hacen que el software libre se comporte de una forma tan especial. A continuación se expone, a modo de ejemplo, una visión rápida de lo que supone el software libre en algunos escenarios.

### Usuarios finales

Probablemente el principal cambio que experimentan los usuarios finales al entrar en el mundo del software libre es que pueden encontrar un alto nivel de competencia en un mercado con gran tendencia al monopolio de facto. El usuario puede usar el producto que desee, y buscar soporte para él en cualquier empresa que quiera dárselo. Cambiar de una a otra es fácil, sin tener que cambiar de producto, ya que todas ellas trabajan con el mismo programa: el código libre que tiene el usuario. Así, el

usuario podrá elegir a la que mejor relación calidad/precio le ofrezca según sus necesidades, como hace siempre en cualquier otro mercado libre. Las empresas que quieran dar este tipo de servicio (que incluye en principio cualquier tipo de garantía, servicio o modificación sobre el programa) pueden hacerlo con la única limitación de su conocimiento sobre el programa y la cantidad de recursos que quieran poner a disposición de sus clientes.

Cuando se usa software libre ya no se depende de la fiabilidad del fabricante para intentar deducir la calidad de un producto, sino que la guía nos la dará la aceptación de la comunidad y el estudio del programa por terceras partes (que podrá hacer en detalle, al estar el código fuente disponible). Nos olvidamos además de "cajas negras", en las que hay que confiar sin más remedio, y de las estrategias de los fabricantes que pueden decidir unilateralmente dejar de mantener un producto.

La evaluación de productos antes de adoptarlos es también mucho más sencilla, ya que basta instalar los productos alternativos en nuestro entorno real y probar. Antes, usando software propietario, era preciso confiar únicamente en informes externos, o negociar pruebas con los proveedores, lo cual no es siempre posible ni económicamente viable.

Dada la libertad de modificar el programa para uso propio, el usuario

puede personalizarlo o adaptarlo sus las necesidades, corrigiendo errores si los tuviera (o contratar a alguien que lo haga por él). También podemos, directamente o contratando servicios, integrar el programa con otro, o auditar su calidad (por ejemplo la seguridad). El control pasa, en gran medida, del proveedor al usuario.

### Administraciones públicas

Las administraciones públicas son grandes usuarios con características especiales, ya que tienen ciertas obligaciones con respecto al ciudadano. Por ejemplo, proporcionarles servicios accesibles y neutrales respecto a los fabricantes, o garantizar la integridad, utilidad, privacidad y seguridad de sus datos a largo plazo. Esto les obliga a estar especialmente interesadas en los estándares y los formatos de datos abiertos, y en la posibilidad de realizar detalladas inspecciones sobre el software que usan. En el mundo del software propietario esto ha sido siempre un problema para las administraciones, especialmente en presencia de mercados con situaciones de monopolio de facto. Afortunadamente el software libre, con su gran respeto por los estándares abiertos y su directa disponibilidad para la inspección de código, ofrece muchas soluciones en este campo.

Por otra parte, las posibilidades que ofrece el software libre de creación de



una industria de alto valor tecnológico, a nivel local o regional, y sin dependencias estratégicas, son sin duda interesantes para cualquier administración que esté interesada por el desarrollo de su ámbito de actuación.

### El desarrollador y el integrador de software

Para el desarrollador y productor de software, el software libre cambia mucho las reglas del juego. Con él es más fácil competir siendo pequeño y acceder tecnología punta. Puede aprovechar el trabajo del resto de la comunidad del software libre y quizás (si se sigue el modelo de desarrollo adecuado) conseguir la colaboración libre y desinteresada de otros desarrolladores (quizás desinteresados, pero seguro también interesados por la calidad del programa que se está desarrollando), del que el iniciador será la referencia. La distribución del software creado es de bajo coste, y su escala global.

No obstante, no hay que perder de vista que al renunciar al cobro de licencias de uso (la fuente de ingresos más habitual en el mundo del software), habrá de buscar nuevos modelos de negocio (de los que se habla más adelante).

Para el integrador el software libre es un paraíso. No habrá más cajas negras que intentar encajar con grandes dificultades y sin disponer de toda la información necesaria. En el mundo del software libre, el integrador puede limar asperezas e integrar fragmentos de programas para conseguir el producto integrado necesario, disponiendo de un acervo ingente de software de donde extraer las piezas.

### La industria de mantenimiento y servicios

El disponer del fuente lo cambia todo, situando a las terceras partes casi en las mismas condiciones que el productor. Lo único que le separará de poder ofrecer cualquier servicio o modificación sobre un programa será su nivel de conocimiento sobre él y la dedicación de suficientes recursos para la tarea. Podrán ofrecerse servicios con un valor añadido mucho más apreciado, y sin dependencias de la estrategia de los proveedores. En la actualidad esta es la categoría de negocios más clara en el mundo del software libre (quizás junto con la del integrador), y en la que está apareciendo más rápidamente un mercado maduro de ofertas.

### Principales modelos de negocio

Puede decirse que son muchos los modelos de negocio que se están explorando alrededor del software libre, algunos más clásicos y otros más innovadores. Hay que tener en cuenta que entre los más habituales en la industria del software no es fácil usar los basados en la venta de licencias de uso de programas producidos, pues dadas las libertades ya comentadas, ese es un mecanismo de financiación muy difícil de explotar. Sin embargo, sí se pueden utilizar los basados en el servicio a terceros, con la ventaja de que sin ser necesariamente el productor de un programa puede darse soporte completo sobre él.

Dadas estas limitaciones (y estas ventajas) desde hace unos años se están probando variantes de los modelos de negocio habituales en la industria del software, a la vez que se buscan otros más innovadores que explotar las posibilidades que ofrece el software libre. Sin duda en los próximos años veremos aún más experimentación en este campo, y también tendremos más información sobre qué modelos pueden funcionar bien, y en qué circunstancias.

En este apartado vamos a ofrecer una panorámica de algunos de los que

nos encontramos hoy día, centrándonos en los modelos basados en el desarrollo y los servicios alrededor de un producto de software libre. Los ingresos en este caso provienen directamente de estas actividades, pero no necesariamente implican desarrollo de nuevos productos.

En cualquier caso, y aunque aquí se ofrece una clasificación relativamente clara, no hay que olvidar que casi todas las empresas usan en realidad combinaciones de los modelos que describimos, entre ellos y con otros más tradicionales.

### Modelos basados en mejor conocimiento

La empresa que utiliza este modelo de negocio trata de rentabilizar su conocimiento de un producto (o conjunto de productos) libres. Sus ingresos provendrán de clientes a los que venderá servicios relacionados con ese conocimiento: desarrollos basados en el producto, modificaciones, adaptaciones, instalaciones e integraciones con otros. La ventaja competitiva de la empresa estará en gran medida ligada al mejor conocimiento del producto: por ello estará especialmente bien situada si es la productora, o participa activamente en el proyecto lo produce.

Esta es una de las razones por la que muchas empresas participan activamente en los proyectos relacionados con el software sobre el que tratan de vender servicios: es una forma muy eficiente de obtener conocimiento sobre él, y lo que es más importante, de que ese conocimiento sea reconocido. Desde luego, explicarle a un cliente que entre los empleados hay varios desarrolladores del proyecto que produce el software que, por ejemplo, se quiere modificar, puede ser una buena garantía.

Este tipo de empresas tiene un gran interés en dar la imagen de que poseen un buen conocimiento de determinados productos libres. Una consecuencia es que su apoyo a ciertos proyectos de software libre no es algo meramente filantrópico. Por el contrario, puede ser

uno de los activos más rentables de la empresa, ya que sus clientes lo valorarán muy positivamente como una muestra clara de que conocen el producto en cuestión. Además, de esta forma podrán seguir muy de cerca el desarrollo, tratando de asegurarse, por ejemplo, de que mejoras demandadas por sus clientes pasan a formar parte de producto desarrollado por el proyecto.

Los servicios que proporcionan este tipo de empresas pueden ser muy amplios, pero normalmente consisten en desarrollos a medida, adaptaciones o integraciones de los productos en los que son expertas, o bien servicios de consultoría donde aconsejan a cómo utilizar mejor el producto en cuestión (especialmente si es complejo, o si su correcto funcionamiento es crítico para un cliente dado).

#### Ejemplos

- LinuxCare. Fundada en 1996, proporcionaba en sus orígenes servicios de consultoría y soporte para GNU/Linux y software libre en EE.UU., y su plantilla estaba compuesta fundamentalmente por expertos en GNU/Linux. Sin embargo, en 1992 cambió sus objetivos, y desde entonces se ha especializado en proporcionar servicios casi exclusivamente a Linux ejecutando sobre máquinas virtuales VM en grandes ordenadores de IBM. Su modelo de negocio ha cambiado también a mejor conocimiento con limitaciones, al ofrecer como parte fundamental de sus servicios una aplicación no libre, Levanta.
- Alcové. Fundada en 1997 en Francia, proporciona fundamentalmente servicios de consultoría, consultoría estratégica, soporte y desarrollo para software libre. Desde su fundación, Alcové ha mantenido en plantilla a desarrolladores de varios proyectos libres, y ha tratado de rentabilizarlo en términos de imagen. También ha tratado de ofrecer una imagen, en general, de empresa vinculada a la comunidad de software libre, por ejemplo, colaborando con asociaciones de usuarios y dando publicidad a sus colaboraciones con



proyectos libres (por ejemplo, desde Alcové-Labs).

### Modelos basados en mejor conocimiento con limitaciones

Estos modelos son similares a los expuestos en el apartado anterior, pero tratando de limitar la competencia a que pueden verse sometidos. Mientras que en los modelos puros cualquiera puede, en principio, entrar en competencia, ya que el software utilizado es el mismo (y libre), en este caso se trata de evitar esa situación poniendo barreras. Estas barreras suelen consistir en patentes o licencias propietarias, que normalmente afectan a una parte pequeña (pero fundamental) del producto desarrollado. Por eso estos modelos pueden considerarse en realidad como mixtos, en el sentido que están a caballo entre el software libre y el propietario.

#### Ejemplos

- Caldera. La historia de Caldera es complicada. En sus inicios, creó su propia distribución de GNU/Linux, orientada a las empresas: Caldera OpenLinux. En 2001, compró la división de Unix de SCO, y en 2002 cambió su nombre a "SCO Group". Su estrategia empresarial ha dado tantos vuelcos como su nombre, desde su total apoyo a Linux hasta sus demandas contra IBM y Red Hat en 2003, y su abandono de su propia distribución. Pero en lo que se refiere a este apartado, el negocio de Caldera, al menos hasta 2002, es un claro exponente del mejor conocimiento con limitaciones.

Caldera trataba de explotar su conocimiento de la plataforma GNU/Linux, pero limitando la competencia a que podía verse sometida mediante la inclusión de software propietario en su distribución. Esto hacía difícil a sus clientes cambiar de distribución una vez que la habían adoptado, pues aunque las demás distribuciones de GNU/Linux incluían la parte libre de Caldera OpenLinux, no encontraban en ellas la parte propietaria.

- Ximian. Fundada en 1999 con el nombre de Helix Code por desarrolladores muy vinculados al proyecto GNOME, fue adquirida en agosto de 2003 por Novell. La mayor parte del software que ha desarrollado ha sido libre (en general, parte de GNOME). Sin embargo, en un ámbito muy concreto Ximian decidió licenciar un componente como software propietario: el Connector for Exchange. Este es un módulo que permite a uno de sus productos estrella, Evolution (un gestor de información personal que incluye correo electrónico, agenda, calendario, etc.) interactuar con servidores MS Exchange, muy utilizados en grandes organizaciones. De esta manera trata de competir ventajosamente con otras empresas que proporcionen servicios basados en GNOME, quizás con los productos desarrollados por la propia Ximian, que no podrán interactuar tan fácilmente con Exchange. Salvo por este producto, el modelo de Ximian ha sido de mejor conocimiento, y también basado en ser la fuente de un programa (ver más adelante).

### Modelos basados en ser la fuente de un producto

Este modelo es similar al basado en el mejor conocimiento, pero especializándolo de forma que la empresa que lo utiliza es productora, de forma prácticamente íntegra, de un producto libre. La ventaja competitiva aumenta al ser los desarrolladores del producto, controlar su evolución y

En el mundo del software libre es difícil cobrar licencias de uso, pero no imposible. En general, no hay nada en las definiciones de software libre que impida que una empresa cree un producto y sólo lo distribuya a quien pague una cierta cantidad. Por ejemplo, un determinado productor podría decidir distribuir su producto con una licencia libre, pero sólo a quien le pague 1.000 euros por copia (de forma similar a como se hace en el mundo clásico del software propietario).

Sin embargo, aunque esto es teóricamente posible, en la práctica es bastante difícil que suceda. Una vez que el productor ha "vendido" la primera copia, quien la recibe puede tratar de recuperar su inversión vendiendo más copias a menor precio (algo que no

puede prohibir la licencia de un programa libre). En el ejemplo anterior, podría tratar de vender 10 copias a 100 euros cada una. Es fácil deducir cómo este proceso continuaría en cascada hasta la venta de copias a un precio cercano al coste marginal de copia, que con las tecnologías actuales es prácticamente cero.

Aún así, hay modelos de negocio que "hacen eso". Un ejemplo es el de las distribuciones de GNU/Linux, que se venden por un precio muy bajo comparado con sus competidores propietarios, pero superior (y normalmente claramente superior) al coste de copia. Por supuesto en estos casos entran a jugar otros factores, como la imagen de marca o la comodidad para el consumidor.

tenerlo antes que la competencia. Así la empresa desarrolladora se coloca en un lugar muy bueno de cara a los clientes que quieran servicios sobre el programa. También es un modelo muy interesante en términos de imagen, ya que la empresa ha demostrado su potencial desarrollador con la creación y mantenimiento de la aplicación, lo que puede ser muy interesante de cara a convencer a posibles cliente de sus capacidades.

Ejemplos

- Ximian. Ya se mencionado cómo en parte ha usado el modelo de mejor conocimiento con limitaciones. Pero en general, Ximian ha seguido un claro modelo basado en ser la fuente de programas libres. Sus principales productos, como Evolution, RedCarpet o Mono se han distribuido bajo licencias de GNU, y Ximian los ha desarrollado casi en exclusiva desde el comienzo. La empresa ha tratado de rentabilizar este desarrollo consiguiendo contratos para hacerlos evolucionar en ciertos sentidos, para adaptarlos a las necesidades de sus clientes, y ofreciendo personalización y mantenimiento.
- Zope Corporation. En 1995 se funda Digital Creations, que desarrolla un producto propietario para la gestión de anuncios clasificados vía web. En 1997 recibió una inyección de capital por parte, entre otros, de una empresa de capital-riesgo, Opticality Ventures. Lo extraño (en aquella época) de esta inversión es que como condición le pusieron que distribuyera como software libre la

evolución de su producto, lo que más adelante fue Zope, uno de los gestores de contenidos más populares en Internet. El modelo de negocio de la empresa desde entonces fue producir Zope y productos relacionados con él, y ofrecer servicios de adaptación y mantenimiento para todos ellos. Zope Corporation ha sabido, además, crear una dinámica comunidad de desarrolladores de software libre alrededor de sus productos, y colaborar activamente con ellos.

## Modelos basados en ser la fuente de un producto con limitaciones

Son similares a los modelos anteriores, pero tomando medidas para limitar la competencia o maximizar los ingresos. Entre las limitaciones más habituales podemos considerar las siguientes:

- Distribución propietaria durante un tiempo, luego libre. Con o sin promesa de distribución libre posterior, cada nueva versión del producto se vende como software propietario. Pasado un tiempo (normalmente, cuando se empieza a comercializar una nueva versión, esa versión pasa a distribuirse con una licencia libre. De esta manera la empresa productora obtiene ingresos de los clientes interesados en disponer lo antes posible de nuevas versiones, y a la vez minimiza la competencia, ya que cualquier competidor sólo podrá disponer de

la versión libre (disponible sólo cuando ya hay una nueva versión propietaria, supuestamente mejor y más completa).

- Distribución limitada durante un tiempo. En este caso, el software es libre desde que se comienza a distribuir. Pero como nada en una licencia libre obliga a distribuir el programa a todo el que lo quiera (esto es algo que quien tiene el software puede hacer o no), lo que hace el productor es distribuirlo durante un tiempo sólo a sus clientes, que le pagan por ello (normalmente en forma de contrato de mantenimiento). Al cabo de un tiempo, el productor lo distribuye a cualquiera, por ejemplo poniéndolo en un archivo de acceso público. De esta manera, el productor obtiene ingresos de sus clientes, que perciben esta disposición preferente del software como un valor añadido. Naturalmente, el modelo sólo funciona si los clientes a su vez no hacen público el programa cuando lo reciben.

En general, en estos casos las empresas desarrolladoras obtienen los beneficios mencionados, pero no sin costes. Debido al retraso con el que el producto está disponible para la comunidad del software libre, es prácticamente imposible que ésta pueda colaborar en su desarrollo, por lo que el productor se beneficiará muy poco de contribuciones externas.

Ejemplos

- artofcode LLC. Desde el año 2000, artofcode comercializa Ghostscript en tres versiones (anteriormente lo hacía Alladin Enterprises, con un modelo similar). La versión más actual la distribuye como AFPL Ghostscript, bajo una licencia propietaria (que permite el uso y la distribución no comercial). La siguiente (con un retraso de un año, más o menos) la distribuye como GNU Ghostscript, bajo la GNU GPL. Además, artofcode ofrece una tercera versión, con una licencia propietaria que permite la integración en productos no compatibles con la

GNU GPL (en este caso usa un modelo dual, que será descrito más adelante).

- Ada Core Technologies. Fue fundada en 1994 por los autores del primer compilador de Ada 95, cuyo desarrollo fue financiado en parte por el Gobierno de EE.UU., y que estaba basado en GCC, el compilador de GNU. Desde el principio sus productos han sido software libre. Pero la mayoría de ellos los ofrecen primero a sus clientes, como parte de un contrato de mantenimiento. Por ejemplo, su compilador, que sigue estando basado en GCC y se distribuye bajo la GNU GPL, se ofrece a sus clientes como Gnat Pro. Ada Core Technologies no ofrece este compilador al público en general de ninguna manera, y normalmente no se encuentran versiones de él en la red. Sin embargo, con un retraso variable (en torno a un año), Ada Core Technologies ofrece las versiones públicas de su compilador, muy similares pero sin ningún tipo de soporte, en un archivo de ftp anónimo.

## Modelos basados en licencias especiales

En estos modelos, la empresa produce un producto que distribuye bajo dos o más licencias. Al menos una de ellas es de software libre, pero las otras típicamente son propietarias, y le permiten vender el producto de una forma más o menos tradicional. Normalmente, estas ventas se complementan con la oferta de consultoría y desarrollos relacionados con el producto. Por ejemplo, una empresa puede distribuir un producto como software libre bajo la GNU GPL, pero ofrecer también una versión propietaria (simultáneamente, y sin retraso para ninguna de las dos) para quien no quiera las condiciones de la GPL, por ejemplo, porque quiere integrar el producto con uno propietario (algo que la GPL no permite).

Ejemplo

- Sleepycat Software. Esta empresa fue fundada en 1996, y anuncia que

desde ese momento ha tenido beneficios (lo que desde luego es notable en una empresa relacionada con el software). Sus productos, incluyendo Berkeley DB (un gestor de datos muy popular y que puede empotrarse fácilmente en otras aplicaciones) se distribuyen bajo una licencia libre que especifica que en caso de empotrarse en otro producto, ha de ofrecerse el código fuente de ambos. Sleepycat ofrece servicios de consultoría y desarrollo para sus productos, pero además los ofrece bajo licencias que permitan empotrarlos sin tener que distribuir el fuente. Naturalmente, esto lo hace bajo contrato específico, y en general en régimen de venta como software propietario.

## Modelos basados en la venta de marca

Aunque puedan conseguirse productos muy similares por menos dinero, muchos clientes están dispuestos a pagar el extra por comprar una "marca". Este principio es utilizado por empresas que invierten en establecer una marca con buena imagen y bien reconocida, que les permita luego vender con suficiente margen productos libres. En muchos casos no sólo venden esos productos, sino que los acompañan de servicios que los clientes aceptarán también como valor añadido.

Los casos más conocidos de este modelo de negocio son las empresas que comercializan distribuciones GNU/Linux. Estas empresas tratan de vender algo que en general se puede obtener a un coste bastante menor en la red (o en otras fuentes con menos imagen de marca). Por ello han de conseguir que el consumidor reconozca su marca, y esté dispuesto a pagar el sobreprecio. Para ello no sólo invierten en publicidad, sino que también ofrecen ventajas objetivas (por ejemplo, una distribución bien conjuntada, o un canal de distribución que llegue hasta las cercanías del cliente). Además, suelen ofrecer a su alrededor una gran cantidad de servicios, tratando de

rentabilizar lo más posible esa imagen de marca (desde formación hasta programas de certificación para terceras partes).

Ejemplo

- Red Hat. La distribución Red Hat Linux comenzó a distribuirse en 1994 (la empresa empezó a conocerse con el nombre actual en 1995). Desde entonces, Red Hat ha conseguido posicionar su nombre como el de la distribución de GNU/Linux por excelencia. Hoy día, en torno a ese nombre, Red Hat comercializa todo tipo de servicios relacionados con la distribución, con GNU/Linux, y con software libre en general.

## Todo un futuro por delante

Aunque los comienzos del software libre pueden trazarse hasta los años 1970, su entrada por la puerta grande en muchas empresas sólo acaba de ocurrir. Sin embargo, las consecuencias que puede tener para la industria del software, y para la de las tecnologías de la información en general, son tan grandes, que conviene prestarle atención y comprender cómo funciona. Con este artículo se ha tratado de acercar al lector a una de las facetas, que a pesar de ser relativamente desconocida es, una vez que se estudia desde cerca, apasionante. Sólo el futuro dirá si todo lo que hoy promete el software libre se convertirá en realidad, o si por el contrario dentro de unos años sólo será una nota al pie de la historia de la informática. Pero mientras este futuro llega, quizás convenga prestarle atención, aunque sólo sea por si acaso...

### Aviso Legal

© 2003 Jesús M. González Barahona.

Se otorga permiso para copiar y distribuir este documento completo en cualquier medio si se hace de forma literal y se mantiene esta nota.

Este artículo está basado en material incluido en el libro "Introducción al software libre", de Jesús M. González Barahona, Joaquín Seoane Pascual y Gregorio Robles, con permiso de los autores. El libro puede consultarse en <http://curso-sobre.berlios.de/introsobre>