

## F.6. Licencias de software (debate de IweTel)

Por Tomàs Baiget

**Baiget, Tomàs** (ed.). Licencias de software (debate de IweTel).

En: *IweTel*. <http://listserv.rediris.es/archives/iwetel.html> En: *Anuario ThinkEPI*, 2008, pp. 180-190.



**Resumen:** Por juzgarlas de interés recogemos las clarificadoras intervenciones de varios miembros de la lista de correo electrónico IweTel sobre las licencias de software.

**Palabras clave:** Licencias de software, Externalización, Subcontratación, Contratación de servicios, Programas informáticos, Servicios profesionales, Programación.

**Title:** Debate on software licenses

**Abstract:** As we believe they are very interesting and enlightening, we collected the messages of several members of the email list IweTel on software licenses.

**Keywords:** Software licenses, Outsourcing, Service contracts, Computer programmes, Professional services, Programming.

### Isidro Aguillo

Planteó el problema: "Se necesitaba un software para una cierta tarea. Se contactó con un programador al que se le detallaron las funciones que debe cumplir el programa, y que aceptó el encargo pues es capaz de hacerlo. Al serle solicitado el precio preguntó el número de instalaciones que se van a hacer del mismo. Se le responde que esto no es relevante, ya que se le pagará por horas de trabajo, y que si bien el número actual de instalaciones es limitado, se pretende aumentarlo en el futuro. Él indica que no está conforme y que quiere cobrar por número de licencias".

*isidro@cindoc.csic.es*

### Jesús Tramullas

Tú contratas, tú pones las condiciones...

Esa persona te aplica el modelo clásico de cobro de desarrollador de soft que muchas pequeñas y medianas empresas aplican en nuestro país. Pretende cobrar por licencias para ganar dinero como con un modelo *Microsoft/SGAE*: "trabajo una vez, cobro muchas..."

Además, por tu mensaje no me queda claro de quién sería el código.

Yo creo que el código fuente es del que contrata, y más si se trata de una aplicación a medida.

Por lo tanto en el contrato se debería establecer que:

1. El código es del que paga, y se debe entregar bien documentado.
2. Los derechos de propiedad del código son del que paga (otra cosa es que luego lo cedas o decidas liberarlo como soft libre con alguna de las licencias disponibles, etc.).

Siendo previsor con problemas posteriores de ampliación o modificación del programa, yo buscaría otro desarrollador...

*tramullas@unizar.es*

### Óskar Calvo

Obviamente está claro que ese programador piensa que sois unos pardillos: Si aceptáis sus condiciones y él realiza la aplicación vosotros no seréis los dueños del software, sino que estaréis pagando las licencias del mismo, y el programador tendrá el derecho a comercializarlo libremente.

En el "contrato" tenéis que indicar que estáis pagando al programador, o empresa, por hacer un software que será propiedad del *Cindoc* (supongo), y que el *Cindoc* podrá disponer del mismo cómo y cuándo quiera.

Si firmáis un contrato en el que adquirís las licencias de una aplicación, no seréis dueños del producto, sino clientes finales.

Mi sugerencia: pedir presupuestos a diferentes personas, no sólo a una, y replantear

la aplicación, porque una aplicación escritorio hay que instalarla en diferentes equipos, una aplicación servidor, tiene que instalarse en un equipo servidor y los usuarios acceden de forma remota (navegador web, gestor de archivos, etc.).

Tiene que quedar muy claro que todo el desarrollo que se haga del mismo, testeo de calidad, fuentes de la aplicación, etc., pertenece al dueño y no al programador. Y lo mejor es explicarlo y dejarlo bien claro en el contrato, para que no haya malentendidos luego.

[oskartsu@yahoo.es](mailto:oskartsu@yahoo.es)

### José Ramón Pérez Agüera

Desde luego a cualquiera que se le ocurra vender licencias de un software hecho por encargo se merece la pérdida del proyecto. Es como si al comprar la tele nos preguntan que cuánta gente la va a ver y que en función de eso pues pagamos más o menos. Son modelos de negocio sin sentido en la sociedad del conocimiento y, en mi opinión, el surgimiento de licencias tipo GPL no hacen sino cumplir con la necesidad de un cambio a este respecto. Software libre implica conocimiento libre y eso nos hace avanzar a todos.

Espero que lo que hoy está ocurriendo con el software ocurra también a la misma escala en unos pocos años con la cultura y la ciencia en general, nos irá a todos mucho mejor, no sólo a unos pocos.

[jose.aguera@fdi.ucm.es](mailto:jose.aguera@fdi.ucm.es)

### Iulius

La única opción es deshacerse de ese programador y contratar otro. La alternativa implicaría:

– No ser dueños de vuestra aplicación jamás: no la estaríais “comprando”, tan sólo se la estaríais alquilando a ese programador. Si la intentáis mejorar, os cobrará; si queréis actualizarla, os cobrará, si queréis “mirarla” (código) os cobrará... Contratad otra persona que entienda desde el principio que está desarrollando una aplicación de la que vosotros seréis dueños, y que se le paga por ese trabajo.

[iuliuse@gmail.com](mailto:iuliuse@gmail.com)

### Raúl San José

No estando muy de acuerdo con los mensajes que se están recibiendo y que por supuesto respeto, apporto mi opinión desde el prisma del desarrollador... Estoy de acuerdo

## GNU y GPL



*GNU* (que se pronuncia ñu) es un sistema operativo compuesto enteramente de software libre. Su nombre es un acrónimo recursivo de *Gnu's Not Unix*, porque su diseño es como *Unix*, pero se diferencia de *Unix* por ser software libre y por no contener código de *Unix*. *GNU* fue fundado por **Richard Stallman** y ha sido el objetivo inicial de la *Free Software Foundation (FSF)*.

La *General public license* (licencia pública general) de *GNU* (*GNU GPL* o simplemente *GPL*) es una licencia de software libre ampliamente utilizada, escrita originalmente por **Richard Stallman** para el proyecto *GNU*. Es la licencia usada por el kernel de *Linux*. La *GPL* es el ejemplo más popular y conocido de licencia *copyleft*, que exige que los trabajos derivados estén disponibles también

bajo la misma licencia *copyleft*. En virtud de esta filosofía, la *GPL* garantiza a los receptores de un programa de ordenador los derechos definidos por el software libre, y el *copyleft* se utiliza para garantizar la preservación de las libertades, incluso cuando el programa es modificado o se añade a otro. Esto es a diferencia de las licencias de software libre permisivas (*permissive free software licenses*), de los que las licencias *BSD* (*Berkeley Software Distribution*) son un ejemplo, que permiten copiar pero no modificar.

Fuente: *Wikipedia*



100% en que el contrato que se firme es lo que va a misa y es donde debe dejarse claro todo esto. Pero entiendo que el contrato aún no se ha firmado.

Cualquier producto de software original (no pirata) se mueve por licencias. En muchas ocasiones también se contrata soporte de ese producto, y no es lo mismo dar soporte a 1 usuario que a 10, y puede que haya que contratar más personal.

A nivel de diseño y desarrollo del software hay muchas opciones en el mercado: si me pides una base de datos para un puesto lo mismo me planteo hacerla con *Access*, si me lo pides para 10 o más usuarios lo mismo pienso en meter *SQL Server* por medio... y hacer el front en *Visual Studio...*, por tanto lleva más trabajo. Como desarrollador no me cuesta lo mismo la licencia de *Microsoft Access* que la de *Visual Studio*, y por tanto tampoco cuesta lo mismo hacer una aplicación con una u otra herramienta... (estoy poniendo situaciones muy distintas pero que sirven para el tema tratado).

Luego también, desde el punto de vista de un desarrollador, vender por licencias es una forma de controlar la distribución del software, ¿quién me dice a mí que no me compra el software una biblioteca y luego acaba instalado en toda una red de bibliotecas. Si se hace extensible a otras bibliotecas que no han pagado por ello es lógico que pasen por caja y paguen.

Si se trata de una licencia monopuesto se puede vender a bajo-medio coste, porque al final es una licencia más que vendo y entiendo que el negocio puede ser pequeño y el presupuesto reducido...

Como comentáis, *Microsoft* cobra por licencia que distribuye; un ordenador nuevo lleva *Windows* y por tanto se paga por ello, porque va a haber un usuario nuevo que a partir de ese momento va a disfrutar de las ventajas que aporta ese software. Me parece clarísimo que pague por ello...

La otra opción es que diga que no está de acuerdo con el contrato y que se le retire el software *Windows* del PC. Esto se puede hacer pero casi nadie lo hace.

Para resumir, si el grupo de usuarios/centros beneficiados se amplía, es lógico que exista un control de la distribución, pero es que todas las empresas de software lo

hacen... Compras una cámara fotográfica y el soft para tratar las imágenes viene con un número de serie que sólo se puede utilizar en esa cámara que es de la persona que ha pagado por ello al comprarla. Bajas un software de prueba de la web y metes una licencia que es única para la persona que lo ha bajado... No sé, miles de ejemplos...

*raul.san.jose@gmail.com*

## Iulius

Una comparación:

Encargo que me construyan una casa, pero el constructor me dice que de eso nada, que como mucho él se construye una casa y me la alquila por un precio que dependerá del número de habitantes que la ocupen...

Estoy seguro de que este modelo de desarrollo de aplicaciones está bastante implantado y es asumido por muchas personas como "normal". No sé si eso lo hace más justificable. Desde luego no lo hace más justo. Software libre para una sociedad libre.

*iuliuse@gmail.com*

## Óskar Calvo

Hay una diferencia clara, Raúl. *Microsoft* vende un producto cerrado, y lo vende bajo licencias, con un proceso de análisis funcional, orgánico, etc. En cambio en este caso el programador va a plasmar en código lo que le están pidiendo que haga, y no tiene desarrollado dicho producto final. Tienes razón en que el informático puede firmar un contrato en el que quiera poder vender licencias. Pero ojo, como el informático, el autor intelectual del aplicativo tampoco puede venderlo sin que le pueda caer por ello una demanda por plagio por parte de su institución.

Respecto a las licencias, no es cierto que cualquier software original se vende por licencias. Mira si no el software *open source* (no confundir con software libre), donde te entregan el código fuente para que hagas con él lo que desees.

Por otro lado están las herramientas de software libre (*GNU*), que no se venden y no se mueven por licencias, y tampoco son piratería. Hay que darse cuenta de que un producto cerrado sí se puede vender por licencias, tú realmente cuando pagas por un producto a *Microsoft* u otras compañías, lo haces por el derecho de usar su software.

Cuando contratas a una persona, lo haces para que te haga un software específico, que no existe en el mercado, y en ese caso se entiende que el software pertenece a la empresa que contrata, y no al informático.

*oskartsu@yahoo.es*

### **Pedro Martínez Campos**

¿Nosotros desarrollamos software, “libre”? Pues no lo sé. Utilizamos *PHP*, *MySQL*, etc., pero...

En mi empresa, para dimensionar el precio antes utilizábamos el concepto de licencias, pero ahora no lo hacemos. Utilizamos parámetros del tipo: configuración, parametrización, formación, nuevas necesidades, etc.

Cuando un cliente pide un software concreto y específico, es relativamente fácil, nosotros no le cobramos por licencias ni de uso ni de instalación, medimos tiempo de realización y valoramos.

Pero si nos movemos con presupuestos reducidos (pymes, pequeños órganos de la administración) no instalarías ni un solo software de éstos, es por lo que los programadores o las empresas intentamos “amortizar” el software (vendiendo o instalándolo) más de una vez.

El problema surge cuando se hace un desarrollo que se desea que evolucione en el tiempo, que lo usen muchos usuarios, etc., vamos, un software sobre el que se hace un esfuerzo en horas superior a lo normal, y que por supuesto deseas amortizar, ¿cómo lo haces?

O consigues un buen cliente que esté dispuesto a utilizar el software y pagar el desarrollo (alguna administración) o no tiene sentido su construcción.

A pesar de todo yo sigo apostando por el desarrollo y entrego todas las fuentes; imagino que si lo protejo con la licencia oportuna nadie hará negocio con mi esfuerzo. Creo que éste es el dilema de los constructores de software.

*pedrolistas@ya.com*

*pmartinezcambos@102novadoc.es*

### **Raúl San José**

El programador que acepte eso habrá pecado de inexperto y para próximas ocasiones será peor que la persona que ahora están

pensando contratar, porque se sentirá que ha pagado la novatada.

Entremezclando otro tema que hay en el foro:

– ¿no se cobra canon por cada libro prestado?

– ¿se cobra dos veces el canon por un mismo libro a dos personas distintas, verdad?

– pues entonces no entiendo por qué en software se debe hacer distinto...

*raul.san.jose@gmail.com*

### **Celso Gonzales**

En nuestra profesión creemos mucho en el respeto a los derechos de autor, y por ende pensamos que la actividad de creación y desarrollo debe estar amparada bajo las normas que rigen la ley. Minimizar el esfuerzo de creación intelectual o artística de las personas es fomentar la falta de reconocimiento social de la producción de nuevo conocimiento.

Creo que esta persona que nos describe **Isidro Aguillo** ha actuado profesionalmente, pues todo proyecto serio de tecnología de información se maneja sobre licencias.

Podemos pensar en la mano de obra barata de programación, pero me pregunto: ¿no estaremos fomentando un estado de injusticia social de aprovecharnos de la creación intelectual de otros? Además las licencias de *Open Source*, aunque son libres, no debe ser asociadas a la gratuidad. El copyright nace de la necesidad de proteger la creación intelectual, y por ello coexistirán, queramos o no, ambas opciones.

Pienso que esta persona se merece todo el respecto del mundo, y no debemos criticar sus exigencias, pues depende de la persona que lo contrate establecer las condiciones. Éste es sólo un tema de contrato laboral, el cual se rige por la oferta y demanda.

Yo soy bibliotecólogo y también desarrollador de aplicaciones y sistemas de información, y estoy de acuerdo con **Raúl**. Extrapolando el ejemplo, con algo más común: ¿es lo mismo vender los derechos de un libro del que se publiquen 500 ejemplares, que otro que del que se haga una tirada de 1.000.000 de ejemplares (con ilimitadas reediciones), y además se vendan sus derechos a las empresas de televisión y cine, y se cambie el nombre del escritor fantasma por otra persona que

vende más pero no escribe nada. Realmente me parece un negocio redondo, pero ¿cuál es nuestro compromiso ético al respecto?

*celso.gonzales@gmail.com*

### **Jesús Tramullas**

...con respuestas como ésta, cada vez veo más claro que el futuro es desarrollar software de manera cooperativa y libre.

Pero creo que el problema ya es de concepto. Resumo, y me refiero al caso del software a medida:

1. La empresas que programan software privativo pretenden:

- Que les des la idea
- Quedarse el código.
- Cobrar por licencia.
- Venderte servicios de valor añadido.

2. Olvidan que el que encarga y paga es el dueño del producto resultante, no ellos.

– No se les paga por una licencia, se les paga por un producto.

– La jurisprudencia ha dejado claro en varias ocasiones, en problemas entre programador y empresa, que el código es de la empresa.

3. Seguramente todos (usuarios finales) estamos de acuerdo en:

- Yo pago, yo soy el dueño
- Yo pago por un producto a medida, no por una licencia
- Si me quieres vender valor añadido, de acuerdo, pero eso es a posteriori, y si te lo ganas.

– Con mi producto hago lo que quiero, como si lo quiero regalar en CDs un domingo con un periódico.

Uno de los problemas de este país con la informática es el empeño de aplicar modelos de negocio caducos, que cada vez va a ser más difícil mantener, con justificaciones tan peregrinas como el coste del desarrollo... Evidentemente desarrollar es costoso, eso no lo niega nadie, pero hay numerosas empresas que desarrollan en libre y ganan dinero, se mantienen y se expanden.

*tramullas@unizar.es*

### **Raúl San José**

La ley del mínimo esfuerzo, la ley del mínimo coste, la ley del máximo provecho...

¿Quién hace el software libre? ¿Gente que disfruta con su trabajo, que le gusta progra-

mar porque es su hobby, y que lo pone en la web, pidiendo una cita cada vez que usen el código?

¿Así vamos a ser competitivos? ¿Así vamos a crecer?

¿Por qué no estamos los primeros en los rankings de I+D en software?

Para ser más grandes hay que invertir, hay que dar de comer a la gente que nos facilita el trabajo día a día (en este caso los informáticos), y no esperar a que alguien se le ocurra hacer algo que me sirva en un 40% pero que como es libre pues lo cojo...

Software libre o gratis = ahorro, concepto rebajas = no me vale para lo que quiero, pero que como es barato lo cojo.

Así no vamos a ser nunca competitivos, innovadores, punteros.

Estamos siempre dependiendo de lo que otros hagan. Entiendo que dependiendo de donde trabajamos (sector público, privado) contamos con más o menos presupuesto para hacer más o menos inversiones. Ahí está la lucha que debemos tener con nuestros superiores cada vez que se habla de partidas presupuestarias. Si queremos crecer hay que invertir.

*raul.san.jose@gmail.com*

### **Daniel Vidal**

El tema no es en absoluto baladí.

Según qué modelo de contrato os vincule, os podríais encontrar incluso que el programa dejara de funcionar (por expiración de tiempo, falta de nuevas funciones -adaptación a *Windows Vista*, por poner un ejemplo burdo-) y que luego "nadie" lo pudiera modificar/corregir.

Si uno quiere encargar un programa a medida, debe firmar un contrato en el que MUY CLARAMENTE se indiquen las funciones que deben realizarse, así como el lenguaje o código que se va a utilizar. La "autoría" podrá ser del programador, pero la "propiedad" debiera ser del cliente. Así, el código fuente, la documentación técnica, etc., debe estar en manos del cliente, a quien interesaría poder tener la opción (legal y real –en cuanto a viabilidad–) de recurrir a cualquier otro "informático / programador" para realizar futuras actualizaciones, adaptaciones, etc.

La posibilidad o no de utilizar el software resultante (o incluso porciones del mismo

–rutinas, etc.–) por parte del “informático/programador” para otros clientes debe ser así mismo acordada y reflejada en el contrato.

Igualmente, temas como garantías, mantenimientos, etc. (“soporte”) es conveniente que queden reflejados también en el contrato. Todo ello no obsta a que, en realidad, aspectos como el número de instalaciones o la identidad del cliente puedan ser tenidos en cuenta por parte de algunos “informáticos / programadores” para determinar el precio de su producto/servicio, más allá de otras variables más tradicionales como son/eran “la dificultad”, “el número de horas empleadas”, etc.

En cualquier caso, no cabe confundir un “informático / programador” que crea y desarrolla “algo” por su cuenta y riesgo, y, en consecuencia, lo comercializa bajo licencias o cualquier otro sistema que estime oportuno, con otro supuesto muy diferente que es el de un “informático / programador” que es contratado a los efectos de hacer “x” para “y”. Como decía al inicio del mensaje, el tema es muy importante, ya que la “confusión” del párrafo anterior está muy extendida y, de no hacer las cosas bien (un buen contrato), los problemas pueden ser muchos. Y lo dicho es aplicable casi con independencia del tema del software “libre” o no libre, que si bien tiene sus ventajas, también tiene sus contras.

*Martí & Associats, Abogados y Economistas*

*dalvi@retemail.es*

*vidal@martilawyers.com*

### **Jesús Tramullas**

Vaya por delante que estamos hablando del caso particular que ha planteado **Isidro**, y no de la generalidad.

Si una empresa tiene una idea brillante y hace una herramienta, estoy completamente de acuerdo en que quiera protegerla y explotarla. Está en su derecho... Por eso en USA ya se están planteando revisar las patentes de software, vistas las tonterías que se patentan, los abusos que se cometen, y la limitación para la innovación y la libre competencia que suponen.

Quiero pensar que la mezcla de churras con merinas que se está haciendo entre el planteamiento de **Isidro** y el software libre no es interesada. Para mí queda meridiana-

mente claro que si la idea es mía, yo la defino, la encargo y la pago, ningún desarrollador puede venir a decirme que el código es suyo y que me vende licencias.

No se puede llenar la boca de “éticas” y demás, y aprovechar el trabajo que te paga uno para revendérselo a muchos, so excusa para sobrevivir y del coste de desarrollo, autoría, derechos de propiedad intelectual, ni pamplinas.

El modelo de licencias privativas se agotará, sólo es cuestión de tiempo.

En lo que respecta al software libre, hablar de “barato”, “gratuito”, “gente que lo hace en su tiempo libre” y otras lindezas, lo único que demuestra es una soberana ignorancia de los modelos de negocio subyacente, que funcionan, y que se están extendiendo cada vez más. Por cierto, seguro que *OpenOffice*, *Dspace* o *Fedora* son ejemplos de softs que no hacen lo que quiero, pero los cojo porque son gratis...

Acusar al soft libre de la falta de I+D en ingeniería del soft en España es una falacia, típica de los negocios que viven precisamente del mínimo esfuerzo y la máxima rentabilidad. Casualmente España es uno de los países donde la industria informática vive más cómodamente como revendedora de productos foráneos. Conozco empresas punteras en España que desarrollan soft libre, y se dan vida.

Recordemos los casos internacionales de *MySQL*, *Alfresco*, *Magnolia*... Que nadie me diga que comprar licencias es invertir en tecnología y en I+D, por favor...

Tan claro es que este modelo de licencias no va ningún sitio que basta revisar las últimas convocatorias de proyectos europeos para Sociedad de la Información, o el *Profit*, para ver que los programas de innovación a escala nacional y europea potencian el software libre con líneas propias y específicas. El paso es dejar de vivir de licencias, para vivir de servicios de valor añadido.

*tramullas@unizar.es*

### **Óskar Calvo**

Contestando a **Raúl**, decir que desconoce que es el software libre. Hay que tener en cuenta que el software libre hace hincapié en la libre distribución, modificación y en la mejora constante del mismo.

Eso no quiere decir que si una entidad contrata a una empresa para realizar un software se le cobre por el mismo. Y si esa entidad pide una modificación o llevar a cabo una mejora, la simple instalación de software libre, la empresa contratada para tal premisa tiene todo el derecho del mundo a cobrarle su trabajo realizado.

A menudo se confunde el software libre con el software gratuito, sobre todo en inglés, donde la palabra "free" se usa para los dos significados.

La traducción correcta de *free software* no es software gratuito, sino software en el que se puede modificar el código fuente.

Respecto al software libre y gratuito de la red, la mayoría está en la misma porque es más sencillo de adquirir que si tuviésemos que solicitarlo por cd-rom o dvd, incluso así y todo puedes pedir de forma gratuita distribución de ubuntu /kubuntu/ xubuntu/ edubuntu si gasto alguno.

Pensar que una empresa de software libre no es viable, o no es rentable recordemos que *RedHat (Mandriva), Novell (SuSe), MySQL, Ubuntu, etc.*, son empresas que ofrecen servicios profesionales a empresas, y tienen su producto también de forma gratuita.

¿Por qué hacen esto?,

– Porque a diferencia del código cerrado, las comunidades de software libre (código abierto) participan en la mejora del mismo de forma activa. Y cuando hablamos de participar no sólo hablamos de programar, también de procesos de calidad, marketing, traducción, propuestas de mejoras, etc. No hay que ser un programador para trabajar en proyectos de software libre.

Volviendo al tema de derechos de autor, autoría. Si yo me compro una canción en una tienda online resulta que pago por la compra de la misma al autor, pago por tenerla en el reproductor de mp3 al autor, por tenerla en el disco duro al autor, por tenerla en la tarjeta de memoria del teléfono móvil, y en un futuro por haberla bajado por internet, etc.

¿Y esto es correcto?

– No señor. Si yo compro un producto no es lógico que tenga que pagar por el x veces.

Y en el caso que presenta **Isidro**, no se quiere adquirir un producto cerrado, sino que se le presenta a un programador un análisis funcional de la aplicación, y éste únicamente hará el análisis orgánico y la programación del código. Imaginemos que el software fuera para visualizar noticias en internet: el programador tendría derecho a cobrar por cada persona que se conecta a la web para ver noticias. Así nos haríamos ricos todos. Yo al menos, en los proyectos que he llevado a cabo nunca he facturado por licencias, sino por desarrollo únicamente.

*oskartsu@yahoo.es*

### Federico Gramage

Tratándose de una aplicación a medida: se le da la idea, cómo tiene que ser y cómo tiene que funcionar, se le paga por las horas dedicadas a crear y por las horas que le sirvo de conejillo para que funcione...

El programador no puede quedarse con el código fuente. Sería de risa.

*fdgramage@yahoo.es*

### Ailé Filippi

– Contesto a Raúl San José: "¿La ley del mínimo esfuerzo?"

– Te invito a que formes parte de alguna comunidad de software libre (SL), no de los que bajan los programas si no de los que desarrollan.

"La ley del máximo provecho". Esto sí es cierto. Sacar el mayor provecho de las comunidades de SL es lo que buscamos todos los que a ellas pertenecemos, eso sí, "provecho", no "aprovechamiento".

Proyectos como *Debian* o *Ubuntu* no son hechos por hobby o por personas que no quieren dejarlo en un cd. Antes de afirmar algo como eso, deberías conocer cómo se distribuyen las aplicaciones y cómo son las reglas de juego para los desarrollos.

Existen cantidad de pasos y de niveles para poder generar un código, un paquete y que éste sea aceptado.

Estás muy lejos de la realidad.

*ailefi@yahoo.com*

### Débora Fernández

Para aclarar conceptos pego el extracto correspondiente de la Ley de la Propiedad Intelectual:

## **Artículo 97**

Titularidad de los derechos.

1. Será considerado autor del programa de ordenador la persona o grupo de personas naturales que lo hayan creado, o la persona jurídica que sea contemplada como titular de los derechos de autor en los casos expresamente previstos por esta Ley.

2. Cuando se trate de una obra colectiva tendrá la consideración de autor, salvo pacto en contrario, la persona natural o jurídica que la edite y divulgue bajo su nombre.

3. Los derechos de autor sobre un programa de ordenador que sea resultado unitario de la colaboración entre varios autores serán propiedad común y corresponderán a todos éstos en la proporción que determinen.

4. Cuando un trabajador "asalariado" cree un programa de ordenador, en el ejercicio de las funciones que le han sido confiadas o siguiendo las instrucciones de su empresario, la titularidad de los derechos de explotación correspondientes al programa de ordenador así creado, tanto el programa fuente como el programa objeto, corresponderán, exclusivamente, al empresario, salvo pacto en contrario.

5. La protección se concederá a todas las personas naturales y jurídicas que cumplan los requisitos establecidos en esta Ley para la protección de los derechos de autor.

Por consiguiente, si contratas un programador independiente para que te haga un programa de ordenador es él, como su autor, el que posee todos los derechos y por lo tanto, el que fija las condiciones en las que se ha de realizar el desarrollo y posterior implantación o distribución del mismo. Estas condiciones, por supuesto, pueden ser objeto de negociación entre el programador y "el que paga" y si éste no está conforme, pues se busca otro programador y todos contentos. Es la ley del mercado.

Habría que recordar el hecho de que un programa de ordenador tiene la misma consideración que una obra literaria. No olvidemos que hay "autores" que consiguen "negros" sin nombre para que les redacten la novela que luego llevará su firma y su foto. Pues con los programadores pasa igual, sólo hay que buscar un programador que trague con las condiciones que le dicten, claro que en ese caso, quizás su calidad como progra-

mador no alcance el nivel requerido. Los buenos programadores se cotizan. Es la ley del mercado.

Y si realmente lo que se quiere es tener la propiedad tanto del programa como del código fuente, pues según la ley sólo caben dos opciones:

– Comprarle los derechos a su autor.

– Asalarar un programador, esto es, meterlo en nómina.

*debo@mundo-r.com*

## **Jesús Tramullas**

...citar la ley está bien, si se sabe de que se está hablando. Porque un programa de ordenador no lo hace un programador. Eso se dice por desconocimiento de los procesos de desarrollo de software...

Y en segundo lugar, asalariado es el que recibe un salario por hacer algo. Los programadores, por muy *freelance* que sean, cobran y emiten facturas, etc. Se puede demostrar la relación contractual entre ambas partes, de empleador y empleado, o de solicitante y prestador de servicios. Punto 4 del propio artículo.

*tramullas@unizar.es*

## **Celso Gonzales**

Quisiera aportar a lo dicho que en muchos países hay una distinción entre la idea y la expresión de la idea. Se otorgan derechos de autor a las expresiones de las ideas y no a la idea misma o a los métodos de funcionamiento. Este es un mecanismo necesario dentro de la sociedad de crear la libertad de utilizar los conocimientos, sin restricción alguna, lo cual evita el estancamiento y el monopolio de intereses. Por ejemplo, la arquitectura y la forma de funcionar de un sistema de préstamo se pueden utilizar sin restricción alguna, pues es parte de un conocimiento adquirido por la sociedad. Pero lo que no se puede es utilizar la expresión de esas ideas plasmadas en códigos de programación, logotipos, diseño gráfico, etc.

*celso.gonzales@gmail.com*

## **Débora Fernández**

Si el programador no es el que hace el programa me pregunto quién lo hace.

Sí, ya conozco los procesos de desarrollo de software, al menos en su planteamiento



académico, pero en el mundo real, si le pides a un programador *freelance* que te haga un programa, será él el que ejerza de analista, documentador, programador, etc., y las fases de diseño, desarrollo e implantación se solaparán unas con otras para atender a los requerimientos cambiantes del cliente que, aunque es el que “pone la idea”, en realidad esta idea nunca está clara del todo, por eso es por lo que las técnicas de “programación extrema” suscitan tanto interés hoy en día.

Me parece evidente que el punto 4 del artículo 97 de la *LPI* se refiere a la relación entre el trabajador asalariado y su empresario, es decir, que presupone una relación laboral entre ambos, para que si yo tengo una empresa de desarrollo de software que contrata programadores en nómina, éstos no puedan reclamar la autoría del software.

El caso que nos ocupa no es una relación trabajador-empresario, sino empresa-empresa, o b2b (*business to business*), como dicen ahora. Un trabajador *freelance* es empresa y como tal te emitirá una factura por su trabajo, para el desarrollo del cual puede existir o no un contrato, que en ningún caso se consideraría un contrato laboral a efectos de, por ejemplo, una inspección de Trabajo. Un asalariado recibirá su nómina según su contrato laboral. El Estatuto de los Trabajadores lo define claro.

Por otro lado, el punto 4 sólo habla de asalariado-empresario, lo del solicitante-prestador de servicios no lo veo por ningún lado, y

si no está, es pura invención y las invenciones no son ley.

Y, de todas formas, para evitar confusiones en los términos habría que aclarar que cuando hablamos de “software libre” no estamos hablando de “software gratuito”, porque parece que los programadores son unos canallas que tienen la osadía de pretender cobrar por su trabajo, el cual además ha de ser público y libremente disponible.

Pues bien, no. Eso será así si el programador -autor- quiere y si no, no.

*debo@mundo-r.com*

### Jesús Tramullas

Por partes:

1. La propiedad intelectual protege ideas y su plasmación por parte de los autores.

Imaginad un programador que dijera: “como tú eres torpe y no sabes programar, me aprovecho de tu idea, porque yo sí sé, aunque la idea a mí ni se me haya pasado por la cabeza hasta que apareciste tú...”.

Sería lo mismo que reconocer autoría intelectual sobre una pintura o una foto a los que crean los pigmentos para hacer un cuadro, o los pinceles, o el fabricante de rollo fotográfico...

Por algo será que los programadores usan “herramientas software”.

2. Técnicamente, hoy en día la mayor parte del desarrollo del código, función principal del programador, está automatizada en un 60-70%. Cualquiera que haya usado un

## IDE

Un entorno de desarrollo integrado o en inglés *integrated development environment (IDE)* es un programa compuesto por un conjunto de herramientas para un programador. Puede ser exclusivamente para un sólo lenguaje de programación o bien, poder utilizarse para varios.

Un *IDE* es una aplicación compuesta de un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI. Los *IDEs* pueden ser aplicaciones por sí solas o ser parte de aplicaciones existentes. El lenguaje *Visual Basic* por ejemplo puede ser usado dentro de las aplicaciones de *Microsoft Office*, lo que hace posible escribir sentencias *Visual Basic* en forma de macros para *Microsoft Word*.

Los *IDEs* proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como *C++*, *Java*, *C#*, *Delphi*, *Visual Basic*, *Object Pascal*, *Velneo*, etc.

Fuente: *Wikipedia*

IDE tipo *Eclipse*, o *Ruby on Rails*, o un IDE para *PHP* conoce la cantidad de código que se genera automáticamente...

Quiero suponer que los sufridos programadores reconocerán la autoría de los que idearon la generación automática de código, y les pagarán la parte correspondiente de instalaciones, de venta de licencias, etc.

Me encanta la referencia a la "programación extrema"

[http://en.wikipedia.org/wiki/Extreme\\_programming](http://en.wikipedia.org/wiki/Extreme_programming)

que, resumida y superficialmente, es hacer algo muy adaptable que funcione en breve tiempo, para satisfacer al cliente, acelerando los procesos. Más que nada porque no sé a qué viene aquí (luego sí que vienen los problemas con los clientes, que, casualmente tienen claro lo que quieren cuando ven que la aplicación no hace lo que debería).

3. Lo de los *freelance*/autónomos me queda clarísimo... Cuando un autónomo alquila mi baño, como puede hacer una composición artística con las baldosas siguiendo mi idea, pues resulta que si deseo cambiarlo, no puedo, porque puede alegar derechos... y si alguien entra en mi casa, le gusta, y decide hacer algo similar en su baño, pues tiene que pagar un derecho al alicatador original... Es a todas luces ridículo.

Por cierto. Si una empresa contrata a otra, que a su vez subcontrata autónomos... ¿de quién es el código? Llévelo a los tribunales, y creo que lo dejarán muy, muy claro. Y no creo que recurran al Estatuto de los Trabajadores para definir las relaciones comerciales y contractuales entre ellos.

4. Precisamente el follón con las patentes de software viene del mismo enfoque que aquí, con una mal entendida "propiedad intelectual exclusiva", sobre una actividad que, hoy en día, es en su mayor parte técnica y productiva. Quizá en el momento de redacción de la ley tenía su sentido, pero hoy en día, cuando el software es una *commodity* (mercancía) no lo tiene en el 80% de las situaciones.

5. En cualquier caso, a mí me parece bien que los autores originales del soft puedan solicitar su reconocimiento. Nosotros lo hemos hecho..., para acto seguido liberarlo como soft libre. El que una herramienta sea soft libre no quiere decir que su autor no

pueda establecer y ver reconocida su autoría original. El soft libre se caracteriza por las cuatro libertades que ya redactó **Stallman** tiempo ha.

### Las cuatro libertades del software libre, según Richard Stallman

1. Ejecutar el programa como se desee.
2. Estudiar el código fuente del programa y cambiarlo para que el programa haga lo que uno desea.
3. Distribuir copias exactas a los demás, cuando se desee.
4. Distribuir copias de versiones modificadas a los demás, cuando se desee.

6. De nuevo mezclando, quiero pensar que no interesadamente, churras con merinas. El que sea soft libre no quiere decir que todos los servicios de valor añadido existentes a su alrededor sean gratuitos. De la misma forma, muchos desarrollos en soft libre son pagados a programadores en nómina: *IBM*, *RedHat*, *MySQL*, la librería *Libc* del propio *GNU*... Aquí nunca hemos dicho que los programadores no cobren adecuadamente por su trabajo, ni mucho menos. Ni que todos sean unos delincuentes, que deberían liberar obligatoriamente sus productos. Sólo hemos resaltado el abuso que se produce en determinadas situaciones, amparados en una interpretación torticera de la tan traída propiedad intelectual. Y hemos defendido que hay otros modelos de negocio que, pese a quien pese, funcionan.

[tramullas@unizar.es](mailto:tramullas@unizar.es)

### Referencias

**Morgan, Jason V.** Chaining open source software: the case against software patents.

<http://lpf.ai.mit.edu/Patents/chaining-oss.html>

**Ríos, Mauro D.** Licencias, software y autoría, una mirada objetiva.

<http://www.noticias.com/articulo/22-10-2005/mauro-d->

[rios/licencias-software-y-autoria-mirada-objetiva-4mgi.html](http://teoriapolitica.mforos.com/123957/2171473-el-software-libre-y-sus-limites-bajo-el-capitalismo/)

**Vázquez, Leonardo.** El software libre y sus límites bajo el capitalismo.  
<http://teoriapolitica.mforos.com/123957/2171473-el-software-libre-y-sus-limites-bajo-el-capitalismo/>

**Merit.** Study on the economic impact of the open source software on innovation and the competitiveness of the Information and Communication Technologies (ICT) sector in the EU.  
<http://ec.europa.eu/enterprise/ict/policy/doc/2006-11-20-flossimpact.pdf>

**Documentados.** Linux le come terreno a Microsoft.  
[http://oskar.documentados.com/index.php?option=com\\_content&task=view&id=62&Itemid=27](http://oskar.documentados.com/index.php?option=com_content&task=view&id=62&Itemid=27)

**Documentados.** Problemas con Windows Vista, suma y sigue.  
[http://oskar.documentados.com/index.php?option=com\\_content&task=view&id=57&Itemid=27](http://oskar.documentados.com/index.php?option=com_content&task=view&id=57&Itemid=27)

**Documentados.** Windows Vista se hunde.  
[http://oskar.documentados.com/index.php?option=com\\_content&task=view&id=64&Itemid=27](http://oskar.documentados.com/index.php?option=com_content&task=view&id=64&Itemid=27)

**Baquía.** A Windows Vista le llueven las críticas.  
[http://oskar.documentados.com/index.php?option=com\\_content&task=view&id=43&Itemid=27](http://oskar.documentados.com/index.php?option=com_content&task=view&id=43&Itemid=27)

**MimetistWebLoG.** Ubuntu vs Windows.  
<http://www.elrinconde.com/mimetist/2006/06/20/ubuntu-vs-windows/>

**Gallego, Norberto.** La disputa sobre 'protegitibilidad' del software enfrenta intereses económicos.  
<http://www.lavanguardia.es/gen/20070308/51312624361/noticias/la-disputa-sobre-protegitibilidad-del-software-enfrenta-intereses-economicos-microsoft-windows-estados-unidos-parlamento-europeo-comision-europea-bell.html>

**Senso, José Antonio.** Trabajo colaborativo: Basic support for cooperative work (BSCW). En: Software libre para servicios de información digital. Esteban Navarro, M. A. [et. al.]. Madrid: Pearson, 2006.

**Díaz Varela, Mar.** El Gobierno aplicará un canon sobre DVD, ordenadores e impresoras.  
<http://www.internautas.org/html/2832.html>

**Van Peteghem, Luc.** Peerless.  
[http://www.freeband.nl/freenovation/index.cfm?mag\\_id=1054&art\\_id=1065&language=en](http://www.freeband.nl/freenovation/index.cfm?mag_id=1054&art_id=1065&language=en)

**Proyecto I-Share.** I-Share: sharing resources in virtual communities for storage, communications, and processing of multimedia data.  
<http://www.freeband.nl/project.cfm?language=nl&id=520>



## MÁSTER EN DOCUMENTACIÓN DIGITAL

### Máster y Programas de posgrado online

Arquitectura de la información en sitios web. ■ Organización de la información y sistemas de gestión de bases de datos. ■ Buscadores y posicionamiento web. ■ Documentación audiovisual. ■ Gestión de proyectos en centros de información. ■ Archivística digital. ■ Gestión de contenidos y gestión del conocimiento. ■ Bibliotecas digitales. ■ Usabilidad.

**Dirección:** Cristòfol Rovira y Lluís Codina

**Calendario:** oct 07 - jun 08

[www.documentaciondigital.org](http://www.documentaciondigital.org)

## MÁSTER EN BUSCADORES

### POSICIONAMIENTO, MARKETING ONLINE Y VIGILANCIA TECNOLÓGICA

#### Máster y Programas de posgrado online

El objetivo general del curso es proporcionar una sólida formación sobre todos los aspectos relacionados con los buscadores de Internet en sus diferentes facetas: herramientas de recuperación de información, medios privilegiados para el marketing online, instrumentos para implementar programas de vigilancia tecnológica e inteligencia competitiva.

**Dirección:** Cristòfol Rovira y Lluís Codina

**Calendario:** nov 07 - jun 08

[www.masterenbuscadores.com](http://www.masterenbuscadores.com)

#### Colaboran:

Observatorio de Comunicación Científica · UPF  
Instituto Universitario de Lingüística Aplicada · UPF  
DOC 6 Consultores de Recursos de Información

Infórmate

[www.idec.upf.edu](http://www.idec.upf.edu) | [info@idec.upf.edu](mailto:info@idec.upf.edu) | Tel. 93 542 18 50 | Balmes 132, Barcelona