

# Licencias de Software: Antecedentes

M. en C. Eduardo Vega Alvarado  
Lic. Silvia Toledo Molano  
M. en C. Ma. Aurora Molina Vilchis  
Profesores del CIDETEC-IPN

**S**oftware es un término general que se aplica a las diversas clases de programas (*conjuntos de instrucciones*) que dirigen a una computadora y a sus dispositivos relacionados para llevar a cabo una tarea específica [1]; eventualmente, esta definición se puede ampliar para incluir a la documentación relativa al funcionamiento y uso de tales programas (*manuales*) [5].

Cuando se desarrollaron las primeras computadoras electrónicas, los pioneros en el campo no previeron que aparecería una actividad denominada programación de computadoras; sin embargo, a partir de la década de 1990 el desarrollo y mercadeo del software superó por mucho al del hardware, el cual para entonces se había convertido en un producto barato de producción en masa [2]. Como resultado de esta enorme expansión, las licencias de derechos son cada vez más importantes como mecanismos para protección del software, y en muchas ocasiones se vuelven un punto de conflicto, debido principalmente al desconocimiento general sobre las mismas.

En el presente artículo se describe de manera objetiva a los antecedentes y a las diferentes opciones para

licenciar software, considerando tanto la óptica del programador como la visión de los distribuidores de programas y el sentir del usuario final.

---

## HISTORIA

---

Las primeras computadoras electrónicas digitales surgen a finales de la segunda guerra mundial, básicamente como calculadoras de alto rendimiento; como consecuencia de esta especialización, la programación tanto de sistemas como de aplicaciones simplemente no existía. En 1951, la aparición de la *UNIVAC (Universal Automatic Computer)* marca un hito al ser el primer equipo pensado para su venta como una computadora de propósito general (universal); a partir de entonces comienza a tomar fuerza la idea de que una misma computadora pueda realizar tareas totalmente diferentes, únicamente cambiando su secuencia de órdenes (programa). Así van surgiendo los sistemas operativos, los lenguajes de programación y los compiladores, dando origen a la programación de sistemas, en una dinámica que se mantiene hoy en día.

El enfoque de comercialización aplicado inicialmente consideraba a los programas y a los equipos como sistemas completos, bajo un esquema propietario, por lo que sus partes no se distribuían de forma independien-

te; no existían entonces proveedores de aplicaciones ni se pensaba siquiera que un equipo ejecutara programas de otro fabricante. Sin embargo, en diciembre de 1968, bajo la presión del gobierno de Estados Unidos, IBM anunció que al año siguiente *desincorporaría* el software; esto es, lo vendería por separado en lugar de combinar su costo con el de los sistemas físicos. El software permanecería etéreo, pero ahora podría comprarse y venderse. El efecto de esta decisión era abrir el campo del software a la competencia comercial, la cual se vería alimentada por la fuerza del mercado libre [2].

Como consecuencia de esta medida, comenzó un debate aún vigente sobre si el software debe considerarse como propiedad intelectual y, en consecuencia, en que términos deben generarse las licencias para su uso, distribución, modificación y copia, entre otros aspectos; incluso la discusión se extiende al análisis sobre la patentabilidad del software, ya que algunos lo consideran como un desarrollo y para otros es una invención.

La selección de un tipo de licencia de software no es un asunto simple, dado que hay muchos detalles involucrados, e incluso el decidir que licencia utilizar para un proyecto particular puede ser un proyecto por sí mismo [8]. Es fundamental considerar el punto de vista del interesado, esto es, si es creador, distribuidor o

usuario ya que incide directamente en su percepción respecto a que aspectos del software se deben proteger mediante una licencia. En general, las opciones para licenciar software contemplan dos alternativas:

1. Las licencias propietarias
2. El software libre y de fuente abierta (*FOSS, Free and Open Source Software*).

---

### SOFTWARE PROPIETARIO

---

El término *propietario* describe una tecnología o producto que pertenece exclusivamente a una compañía, la cual conserva en secreto el conocimiento acerca de dicha tecnología o del funcionamiento interno del producto [1]. Incluso la mayoría de los programas propietarios solo funcionan adecuadamente si se utilizan con otros productos de la misma compañía; Microsoft es considerado como el mejor ejemplo de ello, siendo el caso Windows – Office totalmente representativo.

En el caso del software propietario, lo único que generalmente permiten las licencias es el uso de la aplicación, prohibiendo o limitando severamente cualquier otro permiso sobre el producto. Por lo mismo, la copia o distribución ilegal de los programas se ha convertido en una preocupación permanente para los poseedores de las licencias, tal como lo muestra la afirmación siguiente [4]:

*La magnitud de la piratería cometida contra toda clase de contenidos digitales tales como música, software y películas se ha vuelto monstruosa. Este problema tiene enormes repercusiones económicas y ha sido causa de cierta parálisis en la creatividad ¿porqué crear si no se recibe recompensa por el esfuerzo?.*

La polémica respecto al software propietario aumenta cuando se consideran aspectos tales como:

- el valor de la propiedad intelectual, suponiendo que dicha propiedad sea éticamente correcta, en otra palabras ¿cuánto valen los conocimientos aplicados en un programa?
- la creación y sostenimiento de monopolios y oligopolios en una época donde la ley de la oferta y la demanda, la libre competencia y la apertura de mercados son la base de la organización económica mundial, y
- la falta de capacidad de compra en sectores amplios de usuarios, especialmente en los países en vías de desarrollo, que impide la adquisición de estos productos y propicia la piratería.

---

### SOFTWARE LIBRE

---

Richard Stallman, un investigador del Laboratorio de Inteligencia Artificial del MIT, inició en 1984 el proyecto *GNU (GNU is Not Unix)*, con la idea de desarrollar un sistema operativo bajo el esquema de software libre. Stallman inició su proyecto porque sentía que el conocimiento que constituye a un programa ejecutable (esto es, el código fuente) debe ser libre; mientras los fabricantes de software veían un conjunto de secretos industriales que debían protegerse celosamente, Stallman veía conocimiento científico que debía ser compartido y distribuido [3].

Stallman provenía de un ambiente de programadores, acostumbrados a compartir sus desarrollos entre sí. Estos grupos de trabajo no eran un concepto nuevo; si bien se habla de ellos como un movimiento que

aparece en la década de los setenta, sus orígenes se remontan a 1955 cuando surge el grupo *SHARE*, formado por usuarios de sistemas IBM 701 con la finalidad de compartir sus experiencias en la programación de tales equipos [2].

Inicialmente, el proyecto GNU avanzó hasta la implementación de herramientas para la construcción de aplicaciones, tales como bibliotecas de funciones, editores de texto y compiladores; sin embargo, el desarrollo del sistema operativo base no tuvo mayor progreso hasta la aparición del kernel de Linux. El sistema operativo libre Linux inició una de las mayores revoluciones en la historia de la computación cuando brotó de los dedos de un estudiante finlandés llamado Linus Torvalds, hace ya 15 años. Actualmente, Linux encabeza ahora ventas anuales de hardware, software y servicios por al menos 15 mil millones de dólares [6]. Este hecho demuestra lo establecido por el propio Stallman, respecto a que software libre no es sinónimo de gratuito, así como tampoco existe contraposición entre el mismo software libre y el software comercial [7]:

*Contrastar el software libre (o fuente abierta) con el software “comercial” es un segundo malentendido [...] Software comercial significa software desarrollado por un negocio. Muchos programas libres (y de fuente abierta) importantes, incluyendo a OpenOffice y MySQL, son comerciales.*

A pesar de su éxito, el desarrollo futuro del software libre y de fuente abierta enfrenta algunos obstáculos, entre los que destacan:

- Hardware secreto: los fabricantes de HW tienden a mantener en secreto las especificaciones de sus productos, dificultando la escritura de manejadores (*drivers*) para dar soporte a nuevos dispositivos.

- La existencia de bibliotecas de desarrollo que no son software libre, principalmente para programar interfaces con el usuario.

- Las patentes de software, que impiden utilizar estructuras y algoritmos en aplicaciones fuera del software propietario hasta por veinte años.

---

### LICENCIAS DE PRODUCTOS DERIVADOS

---

Un punto que normalmente no se toma en cuenta respecto a las licencias es el estatus de los productos derivados o generados al utilizar algún software en particular, cuando tales productos tienen la forma de programas, bibliotecas de funciones o incluso estructuras de datos con formatos especiales. La mayoría de las herramientas para diseño de estas aplicaciones incluyen provisiones al respecto en sus licencias, por lo que las alternativas para los productos generados están ya definidas, por lo que es muy fácil infringir alguna ley por ignorancia. Es conveniente recordar que, de acuerdo a las normas generales del derecho, el desconocimiento de una ley no justifica su incumplimiento.

---

### CONCLUSIONES

---

Es importante destacar que el debate en el caso del software no es sobre la aplicación o no de licencias, sino sobre las características que estas deben cumplir; el pensamiento general es que todo programa debe estar cubierto por una licencia derivada de los derechos de autor. Cada tipo de licencia tiene un enfoque especial orientado a la protección de aspectos específicos del software; así, mientras algunos piensan en conservar la propiedad intelectual, otros se avocan a asegurar la libertad en todos los aspectos relacionados con la interacción usuario – producto: uso, copia, distribución y modificación, sin que esta libertad conduzca a una pérdida de derechos para el autor.

Evidentemente cada tipo de licencia tiene ventajas y desventajas, y es imposible determinar si alguna es mejor que otras sin considerar una serie de factores, que en muchos casos incluyen aspectos económicos, legales e incluso sociales de gran complejidad. La selección de licencias debe hacerse analizando caso por caso, y dadas las características de la economía actual, el enfoque más adecuado parece ser la incorporación de sistemas abiertos, donde el aspecto comercial no se anteponga a la difusión del conocimiento, sino que se complementen.

En este artículo se presentó un panorama general sobre las licencias de software, con la idea de iniciar una serie de trabajos al respecto; posteriormente se tratarán en detalle las licencias específicas y sus implicaciones económicas, legales y sociales, así como una serie de criterios para la selección de la licencia adecuada en un caso específico.

---

### REFERENCIAS

---

1. Thing, L., *The Whatis?com's Encyclopedia of Technology Terms*. Que Publishing, 2002.
2. Ceruzzi, P., *A History of Modern Computing, 2<sup>nd</sup> Edition*. Cambridge: MIT Press, 2003.
3. DiBona, C., *Open Souces, Voices from the Open Source Revolution*. O'Reilly & Associates, 1999.
4. Eilam, E., *Reversing, Secrets of Reverse Engineering*. John Wiley & Sons, 2005.
5. Téllez, J., *Derecho Informático, 3<sup>a</sup> Edición*. McGraw Hill, 2004.
6. Lyons, D., *Toppling Linux, Forbes Global 2.18*, Octubre 30, 2006, p92.
7. Stallman, R., A Note from Richard Stallman (Letter to the Editor), *Technology Review 110.3*, Cambridge, Mayo – Junio 2007, p8.
8. St. Laurent, A., *Understanding Open Source & Free Software Licensing*, O'Reilly & Associates, 2004.