



Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

## El Impacto del Software Libre en las Empresas: P&R (19530 lectures)

Per Ricardo Galli Granada, [gallir](http://mnm.uib.es/gallir/) (<http://mnm.uib.es/gallir/>)

Creado el 11/12/2003 19:53 modificado el 11/12/2003 19:53

*En este artículo hago un resumen, en forma de preguntas y respuestas, de los puntos principales que he tocado en mi [conferencia del impacto del software libre en las Empresas](#)<sup>(1)</sup>, analizado desde un punto de vista sobre todo económico. Espero que os sirva al menos para convencer algún jefe, sobre todo porque es difícil explicar que en realidad el modelo económico del software propietario es tan ineficiente que debería desaparecer de la faz de la tierra ;-).*

### ¿Es económicamente viable el software libre?

El software libre no necesita dinero para que sea desarrollado. Aunque todas las compañías y administraciones dejen de invertir en software libre (mediante programadores, código, donaciones, etc.), siempre habrá programadores que lo harán por motivaciones distintas a las económicas directas.

El software es una creación intelectual, aunque se incrementen los consumidores no significa que el o los autores originales tengan una pérdida patrimonial. Éste aspecto es el que se ha tenido en cuenta para la elaboración de las leyes de protección intelectual.

Pero además de ello el software tiene una característica única a diferencia de otras creaciones humanas: puede ser desarrollado y mejorado por miles de personas, como se ha demostrado el núcleo Linux o el escritorio KDE.

### ¿Pero puede ser el coste del SL cercano o igual a cero? ¿No viola leyes económicas básicas?

Todo lo contrario. Si no hay una motivación económica para el desarrollo no puede considerarse como gasto imputable. Además los costes de replicación (costes marginales) son casi cero, por lo tanto su precio de “etiqueta” debería ser también próxima a cero.

### ¿Es económicamente viable el software propietario?

Los costes del software propietario son muy elevados y no están cercanos a los costes de producción (sólo hay que observar los márgenes de ganancias y salarios de los directivos de Microsoft), por lo tanto el modelo de desarrollo del software propietario es un modelo muy ineficiente. El software propietario sólo será viable si los precios se ajustan al coste real de producción, pero el que finalmente decide es el consumidor.

### ¿Pero el propio consumidor, al elegir el software libre por los costes bajos, no hará que las empresas de software propietario desaparezcan y por lo tanto se perjudiquen los mismos consumidores a largo plazo?

Según los teóricos del capitalismo, la existencia de las empresas está justificada porque es el medio conocido más eficiente de llevar bienestar a los consumidores (mercado). Por lo tanto el objetivo de las empresas debe ser solamente el beneficiar a los consumidores.



## ¿Debería el consumidor dejar de ahorrarse unos pocos cientos de euros y así beneficiar a toda la cadena de producción, que a su vez reportaría beneficio a los consumidores?

No.

Este sí que es un argumento más marxista que capitalista, ya que según dicho argumento “el beneficio del estado está por encima del beneficio de los consumidores”. Es todo lo contrario, el consumidor debe sólo pensar en su beneficio inmediato al adquirir un producto. La supervivencia de las empresas es sólo responsabilidad de sus directivos. Adaptarse o perecer.

## ¿Se morirá la industria del software?

El software siempre deberá ser desarrollado por alguien. Seguramente como *hobby* de muchos programadores pero lo más probable es que las empresas sean las que más inviertan. Además, independientemente de que el software sea libre o propietario, **los costes de desarrollo inicial no son los más significativos en el coste total del ciclo de vida de un programa, sino los costes de adaptación y mantenimiento.**

Si un programa no es modificado después de la liberación de su versión inicial, el modelo que representa se alejará (*gap*) cada vez más de la realidad. Esa diferencia se mantiene al mínimo con el mantenimiento posterior, cuyo coste total se estima en un 75% del coste total del ciclo de vida.

En todo caso la pregunta es: ¿morirá el la industria del software propietario?

## Si no hay ganancias de ventas de software ¿que incentivos tendrán las empresas para desarrollar software libre?

Para la inmensa mayoría de las empresas, el software es un gasto, no produce ganancias.

Para más del 90% de esa minoría de empresas que sí desarrollan software, **dicho software también es un gasto**, ya que ellas no se dedican a producir software para vender y ganar dinero, sino para uso *in-house* o personalizaciones a otras empresas. Para todas esas empresas el software libre sí significa ganancia de dinero: **dinero no gastado es dinero ganado.**

En pocas palabras, las empresas seguirán necesitando software, y ese software deberá seguir evolucionando y adaptándose al modelo real que evoluciona constantemente. Pero lo que puede cambiar radicalmente es el de pasar desde un “modelo de grandes inversiones”, donde se invierten grandes cantidades de dinero con la esperanza de obtener algún beneficio aunque sea de posicionamiento estratégico, a un “modelo de gasto”, donde se invertirá dinero en muchas menores cantidades. Al ser el gasto muy inferior, no habrá que justificar ese dinero con beneficios económicos futuros.

Si todas las empresas hacen este tipo de análisis y gasto el software libre seguirá avanzando y evolucionando con mayor velocidad a la vez que las empresas que desarrollen programas notarán una reducción de gastos importantes.

Dado que las empresas que desarrollan y obtienen ganancias de la venta de paquetes de software son una ínfima minoría del total de empresas y usuarios, los efectos negativos del software libre serán casi imperceptibles.

Pero hay que recalcar, en los últimos 25 años se ha invertido muy poco dinero en software libre, **la inversión de la empresa privada en software libre es casi inexistente comparada con el dinero que se ha gastado en desarrollo de software propietario.** Aún así el software libre ha llegado a tener la mayoría de los servidores de Internet y está ganando rápidamente mercado en los “escritorios”.

Es decir, haya o no haya inversión de las empresas para el desarrollo del software libre, éste seguirá creciendo.



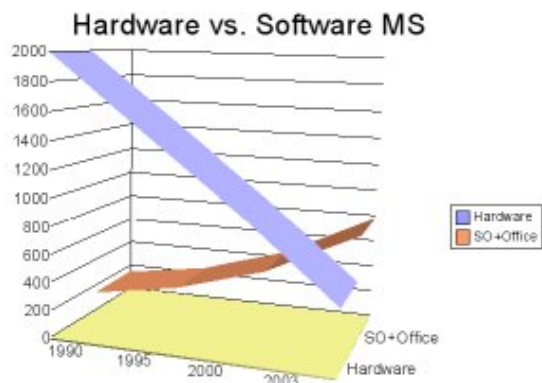
## ¿Morirá el la industria del software propietario?

Sólo podrá sobrevivir si se convierte en un modelo eficiente y haya consumidores que deseen pagar unas funcionalidades que no le ofrecen otros productos cuyos precios son mucho más ajustados. En todo caso, serán los consumidores (el mercado) los que lo decidirán.

## ¿Pero no hay formas de hacer más eficiente al mercado del software propietario?

Seguramente sí, pero lo primero que deberían hacer las grandes corporaciones de desarrollo de paquetes de software propietario es reducir sus precios, al estar muy lejos de los costes de producción. Pero ello hará que no pueden mantener su monopolio (“monopolio inestable”) al carecer de los medios económicos que le facilitan el mantenimiento del predominio en el mercado.

Según algunos economistas, especialmente los de la nueva generación de la Escuela de Chicago, la mejor forma es la que aplica Microsoft y la denominan “internalización de externalidades” o “integración vertical” (este argumento fue usado por la defensa de Microsoft en el juicio “antimonopolio”), o más popularmente conocida como *embrace and extend*. Según esta teoría es la mejor forma de reducir costes de desarrollo al no tener que duplicar trabajo de desarrollo en las diferentes áreas del negocio (tened en cuenta que evitar la duplicación de desarrollo de funcionalidades similares es uno de los fundamentos del software libre)



Si se produce esa integración vertical, lo único que se hará es mantener la condición de monopolio y aumentar el nivel de concentración de toda la industria. Aunque algunos analistas afirmen que Microsoft es un “monopolio natural”, una de las condiciones que debería cumplir para que sea ese “monopolio natural” es que sus precios no sean monopolísticos y que no tenga competidores o productos sustitutos. Si se analiza la evolución de los precios de los productos de Microsoft se observará que sus precios no han parado de subir, tal como fue demostrado en el mismo juicio.

## ¿Pero el coste tan bajo del software libre no hará que suba la del hardware?

Esta afirmación, aunque escuchada muchas veces, tiene la misma lógica que decir que la construcción de nuevas carreteras hará aumentar el precio de los coches. O que la reducción del precio de los peajes tendrá el mismo efecto sobre el precio de los transportes. O que si se baja el precio de las llamadas telefónicas subirán la de los terminales. ¿Ridículo no?

En todo caso el efecto es todo lo contrario, la gente comprará más hardware y el negocio para los informáticos aumentará, no sólo por la mayor cantidad de ordenadores y usuarios, sino porque surgirán nuevas aplicaciones.

No hay que olvidar que la industria del hardware de “infraestructura” está mucho más diversificada que la de los sistemas operativos y suites ofimáticas y desde hace años que ya tiene precios de bien de consumo homogéneo (marca blanca o *commodity*), lo que significa que es un mercado de competencia pura (que por cierto es la pesadilla de los productores).

*Smart companies try to commoditize their products' complements.*

Hay que tener en cuenta el que el hardware es el complementario del software. Y no hay que olvidar que el hardware y software son complementarios de los servicios de los informáticos.



Desde el egoísmo y aplicando reglas elementales de microeconomía, **los informáticos deberíamos fomentar el software libre para obtener más beneficios.**

## ¿No hará que los programadores pierdan sus puestos de trabajo?

[Se estima que en USA sólo el 5% de los programadores desarrolla software propietario para venderlos como paquetes<sup>\(2\)</sup>](#). El restante 95% hace desarrollos para software que será usado en la empresa (*in-house*), mantenimiento y personalizaciones. Para este 95% el software libre es el paraíso, ya que evitan que programen una y otra vez funcionalidades que ya están disponibles en otros programas. Por otro lado, tal como lo demuestra la experiencia de IBM con Websphere y Apache, la mejor forma de reducir el coste de mantenimiento de un programa es hacerlo libre.

## ¿Continuarán programando sin cobrar dinero?

Se ha llegado al estado actual de la informática del software libre con muy poca inversión (comparada a las hechas en software propietario) de las empresas o administraciones. Como ya se ha dicho antes, la económica no es la única motivación. Y lo más probable es que en el futuro las inversiones y gastos en software libre se incrementen, por lo que la pregunta pierde todo su valor. Seguramente habrá muchos que lo hagan por vocación y otros muchos porque le pagan para hacerlo.

## ¿Pero y los costes de oportunidad de los programadores de software libre?

Es una pregunta típica de los “puristas” economicistas. La barrera de entrada al mercado del software propietario es muy alta, las posibilidades de tener éxito son ínfimas y se necesita mucha inversión, especialmente de comercialización. Es mucho menos problemático liberar un programa como software libre, y esperar contribuciones externas para mejorarlo, que intentar comercializarlo.

## ¿Es legal la GPL?

Sí, es perfectamente legal, similar a cualquier licencia de software válida y sustentada sobre las leyes de propiedad intelectual. Cualquier creación de software está protegida por dichas leyes desde el momento mismo de la creación. En particular entran en juego los derechos patrimoniales: reproducción, distribución, comunicación pública y transformación (gracias al Dr. Javier González por la explicación).

La licencia GPL se consideraría en España como “Condiciones generales de un producto”. Básicamente es una autorización de los autores que permite la reproducción (incluye la ejecución, ya que obliga a copiar el programa en la memoria), distribución, comunicación pública (disponibilidad en Internet) y transformación (modificación y mejoras). Para poder hacer cualquiera de las actividades anteriores se deben aceptar y cumplir por las condiciones expuestas en la licencia, que es la única autorización por parte de sus propietarios.

## ¿Que es eso que sea viral? ¿Es malo?

No es malo, es natural.

Se denomina viral a la implementación en una licencia del concepto *copyleft* que exige que cualquier modificación a un programa GPL sea también liberada como GPL. Con ello se asegura que todo el código siga siendo libre. Es sencillamente una protección para evitar que el código se convierta en propietario y por lo tanto sus autores no vean recompensado su trabajo o su trabajo sea usado **sólo** para beneficio de unos pocos.

Las licencias de software propietario son aún más “virales” si cabe. El hecho que uno haya recibido (legal o ilegalmente) una copia de software propietario no significa que la pueda copiar y distribuir libremente sin autorización expresa de los propietarios.



## ¿No pone la licencia GPL en peligro todo mis programas si por error uno de los programadores introduce código GPL?

La afirmación a la respuesta anterior es muy común entre los “analistas” especializados: si por error se introduce código GPL en un programa propietario, debido a la característica viral de la licencia, el programa completo debería ser liberado como GPL. Pero es totalmente falso.

Como ya expliqué antes, se trataría de una violación de la ley de propiedad intelectual y no de violación de un contrato. Hay varias formas de solucionarlo, desde no distribuir el programa con ese código (recordad que la GPL pone restricciones sólo a la distribución y comunicación pública), quitarlos totalmente y posiblemente indemnizar a los autores. Quizás, y a elección del propietario del resto del código, liberar todo con licencia GPL acabaría con la violación de la misma, pero debe ser con conocimiento y voluntad expresa de los propietarios del código.

Ningún autor o juez podría obligar a liberar un software con licencia GPL si no hay esa voluntad expresa. Hay que recordar que cualquier tipo de software, aunque sea no GPL, también está protegido por las leyes de propiedad intelectual.

## ¿Es el software una *commodity*?

No lo es, al menos en la industria del software propietario. No lo es por varias razones, una es que si lo fuese cualquiera podría cambiar de sistema operativo o procesador de textos sin problemas. Ya sabemos que eso es imposible actualmente. Si el software fuese una *commodity* su precio de venta sería muy cercano al de los costes reales de producción. Lo último tampoco es verdad.

## ¿El verdad que el coste total de propiedad (TCO) del SL es más alto que el software propietario?

No, no está demostrado porque es muy difícil hacerlo, varía mucho con cada caso particular. Por ejemplo IBM ha demostrado que usando su arquitectura *mainframe* S390, con cientos de máquinas virtuales Linux es mucho más barato que comprando cientos de PCs diferentes.

Aún así el TCO es una mala excusa para impedir una migración, porque el TCO se fija en los costes de administración individuales.

Por ejemplo, ¿porqué las empresas de autobuses no compran coches utilitarios que tienen un TCO infinitamente menor al de los caros autobuses de 50 pasajeros? Evidentemente la respuesta está en que pocos autobuses hacen el trabajo de cientos de coches. La respuesta se denomina **retorno de inversión**. Debería aplicarse el mismo concepto al hardware y software.

## ¿Pero acaso las licencias de distribuciones como RedHat no son más caras que la de Windows?

RedHat está vendiendo no sólo un producto empaquetado, sino también el soporte y garantía para miles de aplicaciones, no sólo el sistema operativo básico. Además hay que tener en cuenta que RedHat está especializado en el mercado de empresas y que hay otras opciones, desde SUSE a Debian, que es totalmente gratuito.

## Si el software libre es tan bueno y eficiente, ¿porqué no se adoptó o extendió antes?

Porque el software libre no es un competidor del software propietario, sino un sustituto (recordad además que el mercado del software no es una *commodity*). Para hacer el cambio se necesita un cambio de actitud y tomar una decisión importante. Para hacer el cambio hace falta ver ejemplos exitosos de los *early adopters*.

De todas formas hay nichos que el software libre ha sido más exitoso que el software propietario desde hace muchos años, por ejemplo el correo electrónico, servidores web, servidores de ficheros, DNS, etc. En estos casos no tiene



justificación el retraso de algunas organizaciones, especialmente universidades, para adoptar el software libre (más del 90% de los servidores DNS de Internbet son libres, más del 85% de los correos electrónicos pasan por servidores de software libre). Especialmente por la calidad demostrada, el ahorro de costes y flexibilidad de la infraestructura básica de la organización. Quizás la respuesta es que se han dejado “engañar” (o han sucumbido a los “regalos”) de las estrategias comerciales de las empresas.

## El software libre distribuye la riqueza, ¿pero la genera?

En realidad como se está demostrando, el software propietario es un modelo económicamente ineficiente y que por lo tanto con él se pierde “valor”. Así que la pregunta debería hacerse a los que defienden dicho modelo, ya que es muy difícil de justificar económicamente que genere riquezas, al menos más que el software libre.

Pero intentando contestar a la pregunta, según estudios de economistas, las inversiones en tecnologías de la información no aseguran el éxito económico de las empresas, de hecho en muchos estudios no hay ningún tipo de correlación entre inversión tecnológica y rendimiento en los mercados. Por lo tanto no se puede asegurar que el software (sea libre o propietario) automáticamente genere riquezas a las empresas.

## ¿No innova más el software propietario que el software libre?

Es otro argumento repetido hasta la saciedad por los defensores del software propietario, especialmente de grandes empresas. Pero habría que pedirles que enumeren quienes han sido los inventores de las siguientes tecnologías: redes locales, Internet, navegador web, correo electrónico, *thin clients*, administración remota, vídeo digital en el PC, música, sistema gráfico basado en ventanas... ninguna de ellas ha sido inventada por Microsoft ni empresa productora de software propietario. De hecho la mayoría de esas tecnologías son anteriores a 1990 y surgieron en entornos que hoy consideraríamos de “código abierto”.

## ¿No afectarán negativamente las patentes al software libre?

Si, como también al software propietario. Cualquier sistema de software de unos pocos miles de líneas seguramente está afectada por unas cuantas patentes. La única forma de estar seguro es borrando todo el software que haya en los discos duros de los ordenadores.

## ¿Cómo será el negocio informático?

Seguramente muy distinto en lo que se refiere a los paquetes cerrados de software, muy pocos cambios en el mercado de mantenimiento, soporte y personalizaciones y precios más bajos, sobre todo si hay una alternativa de software libre.

En cualquier caso el mercado lo definirán los consumidores, y eso parece no agrandar a algunos empresarios.

---

## Comentarios

Lista de enlaces de este artículo:

1. <http://bulma.net/body.phtml?nIdNoticia=1917>
2. <http://www.catb.org/~esr/writings/magic-cauldron/magic-cauldron-3.html>

---

E-mail del autor: [gallir\\_ARROBA\\_uib.es](mailto:gallir_ARROBA_uib.es)

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=1939>