

Software libre para la gestión documental

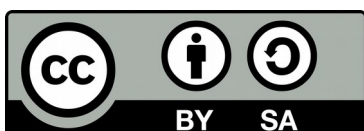
Módulo 1

La revolución del software libre y de código abierto

Guillermo Castellano

ācal
asociación de archiveros de castilla y león





La revolución del software libre y de código abierto

Guillermo Castellano Casas

Septiembre 2017

Maquetado por Guillermo Castellano Casas con LibreOffice 5.0 en Linux Mint 17.3.

Fuente Alegreya Sans de Juan Pablo del Peral.

Imagen de portada de [Pexels](#).

Infografías diseñadas por Ana Amelia Patiño Esteo con Inkscape en Ubuntu 16.04.

Puedes compartir, es decir, copiar y distribuir el material en cualquier medio o formato y adaptar, es decir, mezclar, transformar y crear a partir del material. Las condiciones para esto son reconocer adecuadamente la autoría, proporcionando un enlace a la licencia e indicando si se han realizado cambios (haciéndolo de cualquier forma razonable, sin sugerir que se tiene el apoyo del licenciador o lo recibe por el uso que hace), y compartiendo con la misma licencia si se mezcla, transforma o crea a partir del material.

La revolución del software libre y de código abierto

1. Las cuatro libertades del software libre

El proyecto GNU define el software libre como “el software que respeta la libertad de los usuarios y la comunidad”.

Uno de los falsos mitos más arraigados sobre el software libre es que significa software gratuito. No es la gratuidad lo que define el software libre, sino la libertad de usarlo, estudiarlo, distribuirlo y mejorarlo. Esta confusión proviene de que en inglés “free software” puede significar tanto “software libre” como “software gratuito”, lo que llevó a Richard Stallman (2004, p. 59) —de quien hablaremos en el siguiente epígrafe— a explicar que “debemos pensar en la acepción de libre como en 'libertad de expresión' y no como en 'barra libre de cerveza’”.

Para comprender mejor qué es software libre y qué no lo es, se suele hablar de las cuatro libertades esenciales:

1. La libertad de usar el programa con cualquier propósito (Libertad 0).
2. La libertad de estudiar cómo funciona el programa (Libertad 1).
3. La libertad de distribuir copias del programa (Libertad 2).
4. La libertad de mejorar el programa y hacer públicas estas mejoras (Libertad 3).

Así, un programa que reúne estas cuatro libertades es software libre, mientras que uno que no lo hace es software privativo, porque priva a los usuarios de su libertad. El software privativo también se suele conocer como software propietario o software de código cerrado.

Las libertades 1 y 3 requieren acceso al código fuente del programa, por lo que otra forma de definir el software libre es como aquel programa cuyo código fuente está publicado deliberadamente por el desarrollador para que la comunidad lo utilice, estudie, distribuya y mejore. La importancia del acceso al código fuente hace posible que también nos refiramos al software libre como software de código abierto.

De acuerdo con esta definición, nos encontraremos software gratuito que no es libre y software de pago que es libre. El ejemplo más característico de la primera situación es el software pirata, ya que el programa se utiliza y distribuye sin el consentimiento del propietario, pero también podemos mencionar el freeware, que es un tipo de software sin coste que no se puede estudiar ni mejorar libremente ya que el desarrollador no ha publicado el código fuente.

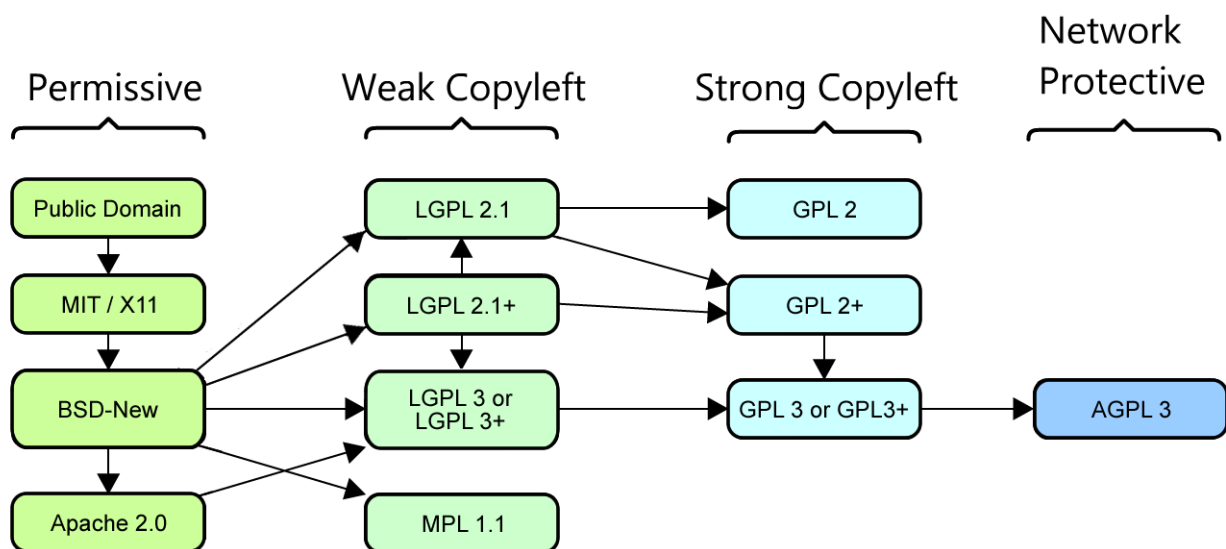
A la inversa, tenemos programas que reúnen las cuatro libertades y se pueden adquirir en modalidades de pago. El caso más típico es cuando pagamos por servicios adicionales, como la instalación del programa en la nube o soporte técnico. De hecho, existen muchos modelos de negocio basados en software libre, como veremos más adelante.

Comúnmente, el software libre se ha distribuido bajo la versión 2 de la licencia GNU General Public License (GPLv2), creada en 1992 por Stallman para garantizar la libertad de los usuarios. Para ello, la licencia declara que el programa es libre y obliga a que cualquier versión modificada del programa se distribuya bajo la misma licencia. La práctica de utilizar los derechos de autor para asegurar la libre distribución de la obra en vez de para restringirla se conoce como copyleft y tenemos otro ejemplo muy conocido en las licencias del tipo compartir-por-igual o share-alike de Creative Commons.



Logos de GPLv3 y la licencia Creative Commons BY-SA (Reconocimiento-Igual).

Existen decenas de licencias de software libre y no todas exigen que las versiones derivadas del programa original se distribuyan bajo la misma licencia. Son las conocidas como licencias permisivas, entre las cuales destacan la licencia Apache y la licencia MIT. Estas licencias permiten, por ejemplo, que alguien distribuya como software privativo su versión de un programa de código abierto. Su popularidad ha crecido en los últimos años y, según un estudio de Ben Balter (2015) sobre el uso de licencias en la comunidad GitHub, la licencia GPL ha sido desplazada del primer puesto por la licencia MIT, utilizada en un 44,69% de los proyectos que cuentan con algún tipo de licencia.



Relaciones de compatibilidad entre licencias libres y de código abierto populares.

Fuente: Daffara (2011).

2. Historia del software libre

2.1. Orígenes del movimiento del software libre (1976-1991)

El inicio de la historia del software libre se suele remontar a 1985, fecha en la que Richard Stallman constituyó la Free Software Foundation (FSF) y publicó el Manifiesto GNU. Otros sitúan su inicio cinco años antes, cuando Stallman quiso mejorar el controlador de la impresora con la que trabajaba y su compañero Carnegie Mellon no le proporcionó una copia del código fuente porque había firmado un acuerdo de confidencialidad con Xerox. Por mi parte, estoy de acuerdo con Slavoj Žižek (2011, pp. 433-434) cuando afirma que “la fecha más importante en la historia del ciberespacio fue el 3 de febrero de 1976” y éste será nuestro punto de partida.

Módulo 1. La revolución del software libre y de código abierto

El 3 de febrero de 1976 Bill Gates publicó su famosa Carta abierta a los aficionados, que debe ser leída, de nuevo en palabras de Žižek (2011, pp. 433-434), como “la afirmación de la propiedad privada en el mundo de los programas informáticos”. En ella, el co-fundador de la recién creada Microsoft arremetía contra la mayoría de los usuarios de ordenadores domésticos por “robar el software” que empleaba e impedir, según él, que se hiciese buen software. Su razonamiento era que compartir el software desincentivaba a los profesionales a seguir produciendo software (Gates, 1976).

El desarrollo colaborativo de software por las comunidades de usuarios fue una práctica habitual hasta finales de la década de 1970 y se tenía la costumbre de distribuir el código fuente del programa para que otros usuarios pudieran estudiarlo y mejorarlo. Stallman (2004, p. 226), que en aquel tiempo era programador en el laboratorio de inteligencia artificial del MIT, recuerda que “todo el sistema operativo era software desarrollado por la gente de nuestra comunidad y compartíamos cualquier parte de él con cualquiera. [...] La cooperación era nuestro modo de vida. Y estábamos seguros dentro de ese modo de vida. No luchábamos por él. No teníamos que luchar por él. Simplemente vivíamos así. Y, hasta donde nos concernía, habríamos seguido viviendo así. Así que había software libre, pero no un movimiento del software libre”.

La carta de Bill Gates supuso el comienzo de una nueva época, marcada por mayores restricciones a los usuarios de software. Los fabricantes adoptaron la idea de que compartir el software equivalía a robar y tardarían más de 20 años darse cuenta de que había modelos de negocio compatibles con la libertad de los usuarios.

El movimiento para promover la libertad de compartir y modificar el software nació como reacción a esta privatización del software. El incidente con la impresora Xerox convenció a Stallman de los problemas del software privativo y en 1984 abandonó su empleo en el MIT para trabajar a tiempo completo en un sistema operativo libre compatible con Unix, que bautizó como GNU (GNU's Not Unix). En 1985 tenía lista la primera pieza de su nuevo sistema: el editor de textos GNU Emacs.

GNU Emacs es el claro ejemplo de que el software libre es una cuestión de libertad, no de gratuidad. Aunque el programa se podía descargar libremente vía FTP, también se podía adquirir una copia física por 150 \$. Stallman utilizó el dinero recaudado por esta vía para continuar financiando el desarrollo de GNU.

1985 también fue el año en el que Stallman (1993) constituyó la FSF y publicó el Manifiesto GNU, en el que explicaba, entre otras cosas, los motivos que le habían llevado a escribir GNU. Para él, desarrollar software libre obedecía, ante todo, al imperativo moral de mantener “la solidaridad con otros usuarios”.

Otra obra destacada de Stallman fue la licencia GNU General Public License, más conocida como GNU GPL o, simplemente, GPL, cuya primera versión fue publicada en 1989. Dos años después apareció la segunda versión de esta licencia, la GPLv2, bajo la que se iba a distribuir software tan importante en la historia de la informática como el kernel Linux.

Fragmento del Manifiesto GNU (Stallman, 1993)

Considero que la Regla de Oro me exige que si me gusta un programa lo debo compartir con otras personas a quienes también les guste. Los vendedores de software quieren dividir a los usuarios y dominarlos para llevarlos a aceptar no compartir su software con los demás. Me rehúso a romper la solidaridad con otros usuarios de esta manera. Mi conciencia me impide firmar un acuerdo de confidencialidad o un acuerdo de licencia de software. Durante años trabajé en el Laboratorio de Inteligencia Artificial oponiéndome a estas tendencias y otras descortesías, pero al final fueron demasiado lejos: no podía permanecer en una institución donde tales cosas se hicieran en mi nombre en contra de mi voluntad.

Para poder continuar a utilizar las computadoras sin deshonor, he decidido agrupar un conjunto suficiente de software libre para poder vivir sin usar ningún software que no sea libre. He renunciado al Laboratorio de Inteligencia Artificial para evitar que el MIT pueda usar alguna excusa legal que me impida distribuir software de GNU.

2.2. Los primeros sistemas operativos libres (1991-1998)

En 1991, Linus Torvalds, en aquel tiempo estudiante de informática en la Universidad de Helsinki, liberó la versión 0.02 del kernel Linux. El kernel o núcleo es un elemento fundamental del sistema operativo, cuya función consiste en asignar los recursos del hardware a los diferentes programas. Linux era la pieza que le faltaba a GNU para convertirse en un sistema operativo libre completo y en 1992 vio la luz la primera distribución GNU/Linux: HJ Lu, a la que pronto siguieron MCC Interim Linux, TAMU, SLS (Softlanding Linux System) e Yggdrasil.

En 1993, Ian Murdock anunció el desarrollo de la distribución Debian, que contó con el patrocinio de la FSF desde noviembre de 1994 a noviembre de 1995. La primera versión estable se lanzó el 17 de junio de 1996 bajo el nombre de Debian Buzz. El proyecto Debian continúa activo a día de hoy y es famoso por la gran estabilidad de su sistema, hasta tal punto que se emplea en entornos críticos como la Estación Espacial Internacional de la NASA. Decenas de distribuciones se derivan de Debian, entre ellas Ubuntu, que ha popularizado el uso de sistemas GNU/Linux en ordenadores de uso personal.

Mientras que el desarrollo de Debian corre a cargo de una comunidad de usuarios—regida por el Contrato Social de Debian, las Directrices de software libre de Debian y la Constitución de Debian—, otras distribuciones GNU/Linux tienen detrás empresas que apuestan por el software libre y de código abierto. La distribución comercial más emblemática es Red Hat, empaquetada y comercializada por la empresa del mismo nombre. Red Hat Inc. fue fundada en 1993 para comercializar productos y servicios basados en software libre, siendo la distribución Red Hat (hoy denominada Red Hat Enterprise Linux) el más famoso de todos. Esta compañía también está detrás del desarrollo de la distribución Fedora, el sistema de empaquetado RPM o el servidor de aplicaciones Java EE JBoss.

A finales de la década de 1990, los sistemas operativos GNU/Linux, que en lenguaje común conocemos simplemente como Linux a pesar de que GNU es un componente esencial de los mismos, ya contaban con millones de usuarios, cientos de desarrolladores y un mercado en expansión. Multinacionales como Oracle Corporation, que en 1998 comenzó a ofrecer soporte comercial para Oracle Database en equipos GNU/Linux, prestaban cada vez más atención al software libre. El éxito de estos sistemas era tal que el software libre quedaría asociado a Linux en el imaginario colectivo.

Nadie imaginaba que el sistema operativo libre que un estudiante de informática finlandés anunció en el grupo de noticias comp.os.minix una noche de verano de 1991 acabaría alcanzado estas dimensiones. El propio Linus Torvalds (1999, p. 101) confesó años después que estaba sorprendido por este éxito. “Me gustaría decir que sabía que esto iba a ocurrir, que todo era parte de mi plan para dominar el mundo, pero [...] vi venir mucho más la transición de un usuario de Linux a cien que la de cien usuarios a un millón”, contó.

Las distribuciones GNU/Linux no fueron el único sistema operativo libre desarrollado en la década de 1990.

En 1974 la Universidad de Berkeley obtuvo una copia de Unix de Bell Labs, propiedad de AT&T, y comenzó a desarrollar su propia versión del sistema, conocida como BSD (Berkeley Software Distribution). Las licencias de AT&T eran demasiado caras para las empresas que querían adquirir esa tecnología, por lo que BSD fue recibiendo cada vez más atención y en Berkeley decidieron trabajar en una distribución completamente libre del código de AT&T. En junio de 1989, Berkeley tenía lista la primera versión libre de BSD, conocida como Net/1.

En 1991 lanzó Net/2, pero tuvo que detener la distribución debido a que AT&T interpuso una demanda contra Berkeley Software Design y la propia Universidad. El juicio se resolvió favorablemente para Berkeley en 1994, pero para entonces GNU/Linux ya había llenado el vacío de un sistema operativo completo libre. Se especula que sin los problemas legales entre AT&T y Berkeley, el kernel de Linus Torvalds probablemente hubiera pasado desapercibido y BSD hoy ocuparía el lugar de GNU/Linux como principal alternativa a los sistemas operativos propietarios.

Módulo 1. La revolución del software libre y de código abierto

Sea como fuere, la sentencia favorable de 1994 relanzó el desarrollo de distribuciones libres de BSD, entre las cuales destacan FreeBSD, OpenBSD y NetBSD. A diferencia de la GPLv2, la licencia BSD permite distribuir bajo otras licencias las versiones derivadas del programa original y empresas como Microsoft incorporaron código de FreeBSD a su software privativo.



redhat®



debian

Logos de RedHat y Debian.

Anuncio del kernel Linux a la comunidad (Torvalds, 1991)

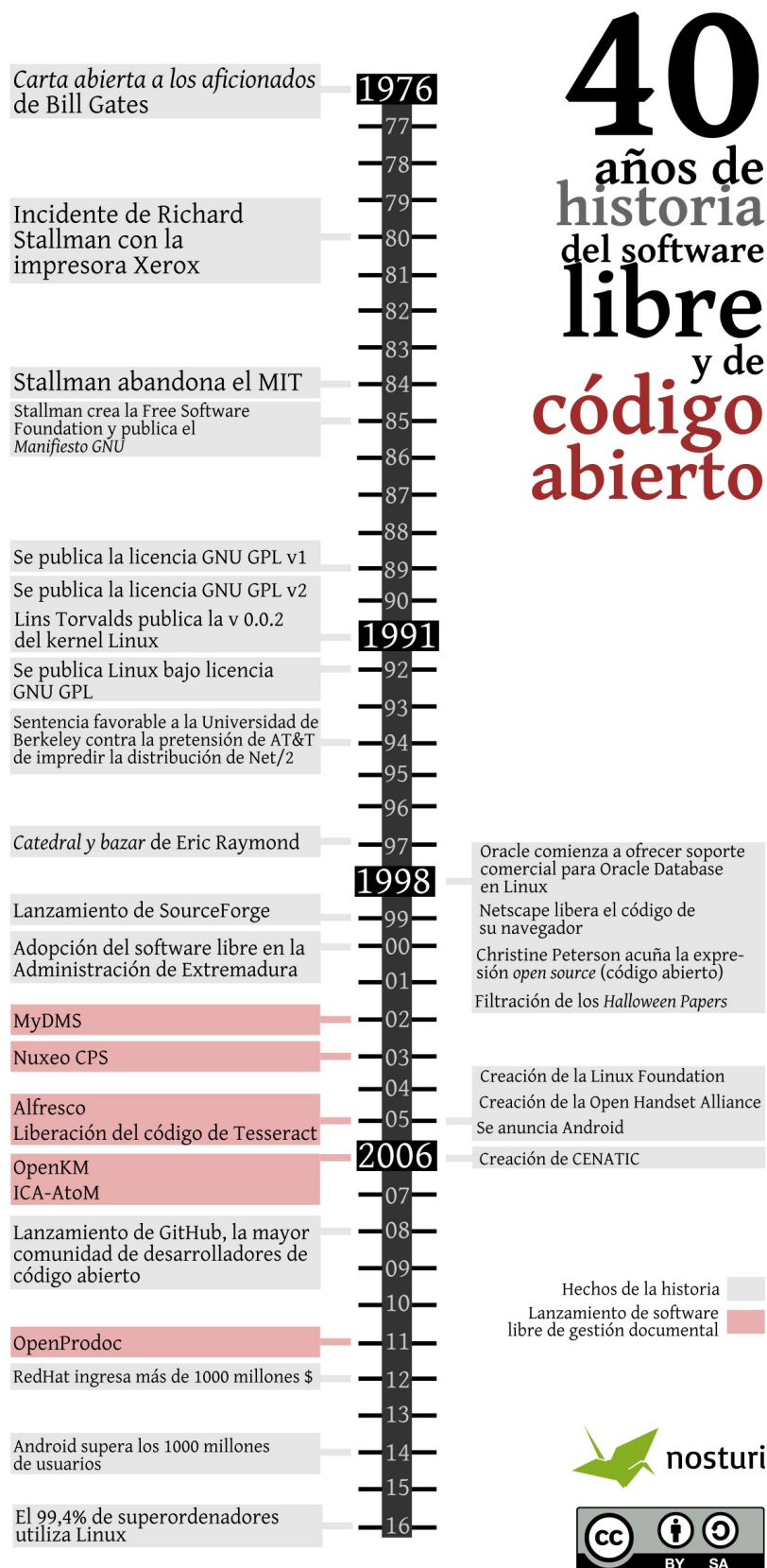
Hola a todos los que estáis ahí fuera usando minix -

Estoy haciendo un sistema operativo (libre), (solamente por afición, no será grande ni profesional como el gnu) para clones 386(486) AT. Este ha estado gestándose desde abril, y está comenzando a estar listo. Me gustaría recibir cualquier comentario sobre las cosas que gustan/disgustan en minix, ya que mi SO (Sistema Operativo) se le parece un poco (la misma disposición física del sistema de ficheros, debido a motivos prácticos, entre otras cosas).

Actualmente he portado bash(1.08) y gcc(1.40), y las cosas parecen funcionar. Esto implica que conseguiré algo práctico dentro de unos meses, y me gustaría saber qué características quiere la mayoría de la gente. Cualquier sugerencia es bienvenida, pero no prometeré que las implementaré :-)

Linus Benedict Torvalds (torvalds@kruuna.helsinki.fi)

PD. Sí – está libre de cualquier código de minix, y tiene un sistema de ficheros multi-hilo. NO es portable (usa el cambio de tareas del 386 etc), y probablemente nunca soporte otra cosa que no sean los discos duros AT, porque es todo lo que tengo :-).



40 años de historia del software libre y de código abierto.

Fuente: Elaboración propia.

2.3. Consolidación del software libre y de código abierto (1998-actualidad)

Lo más revolucionario de Linux no fue el código en sí, sino que puso en valor la capacidad de la comunidad de usuarios para producir buen software. Bill Gates había afirmado en 1976 que compartir el software frenaba la innovación, pero los primeros sistemas operativos libres demostraron que era posible superar los productos de compañías como Microsoft sin restringir la libertad de los usuarios gracias al trabajo colaborativo.

Eric Raymond (1998) teorizó sobre el modo de trabajo de las comunidades de software libre en *La Catedral y el Bazar*. Mientras que GNU y BSD eran desarrollados por grupos pequeños en base a un modelo centralizado —el modelo catedral—, Linux “fue programado de forma más bien eventual por un gran número de voluntarios coordinados a través de Internet. La calidad se mantenía, no con estándares rígidos o autocracia, sino con la estrategia, ingenuamente sencilla, de publicar cada semana y obtener la respuesta de cientos de usuarios en cuestión de días, creando un tipo de selección darwiniana rápida sobre las mutaciones presentadas por los desarrolladores” —el modelo bazar.

La fortaleza del modelo bazar reside en el dicho popular de que “cuatro ojos ven más que dos”. En el mundo del software libre, esta cualidad del desarrollo colaborativo se conoce como Ley de Linus, originalmente formulada por Raymond (Raymond, 1998) en los siguientes términos: “dada una base suficiente de desarrolladores asistentes y beta-testers, casi cualquier problema puede ser caracterizado rápidamente, y su solución ser obvia al menos para alguien”.

El 22 de enero de 1998, Netscape anunció que iba a liberar el código de Netscape Navigator, lo que supuso el punto de partida del proyecto Mozilla, detrás de programas tan populares como el navegador Firefox o el cliente de correo electrónico Thunderbird. Eric Hahn, entonces vicepresidente ejecutivo y CTO de Netscape, reveló a Eric Raymond que la lectura de *La Catedral y el Bazar* fue inspiradora para ellos y los condujo a tomar esa decisión (Raymond, 1998).

El software libre seducía. El soporte comercial para Oracle Database en GNU/Linux y la liberación del código de Netscape Navigator mostraban que los vientos estaban cambiando y Microsoft no permaneció ajena a esta nueva realidad. Gracias a una filtración de documentación confidencial que tuvo lugar en 1998, conocida como *Halloween Documents* o *Halloween Papers*, sabemos que la compañía de Bill Gates percibía el software libre como una amenaza significativa para su negocio y que, a diferencia de lo que declaraba públicamente, era consciente de que los programas de código abierto sí podían competir con el software privativo. Más importante incluso, en esos papeles reconocía que practicaba el FUD (siglas en inglés de Fear, Uncertainty and Doubt) y el Embrace and Extend, dos estrategias comerciales que Microsoft siempre ha desmentido utilizar contra sus competidores (*The Halloween papers*).

Si 1976 representó un punto de inflexión hacia el software privativo y el inicio del movimiento de software libre como forma de resistencia contra la privatización del conocimiento, 1998 marcó el comienzo de la consolidación del software libre y de código abierto.

La denominación de software de código abierto o, simplemente, “open source”, nació de un esfuerzo por parte de un grupo de integrantes del movimiento de software libre de reforzar el acercamiento de las empresas al software libre. Entre ellos estaba Eric Raymond, de quien ya hemos hablado, si bien el término fue idea de la especialista en nanotecnología Christine Peterson (Bretthauer, 2001, p. 17).

Durante la década del 2000, cada vez más empresas se animaron a desarrollar soluciones de código abierto. Las principales soluciones libres que conocemos en el ámbito de la gestión documental, como Alfresco, Nuxeo, OpenKM y AtoM, estaban listas a la altura de 2006. Uno de los acontecimientos más importantes de aquella década fue la creación en 2007 de la Open Handset Alliance, una alianza comercial que se dedica a desarrollar estándares abiertos para dispositivos móviles, entre ellos Android, un sistema operativo de código abierto basado en Linux que en 2014 superaría los 1.000 millones de usuarios en todo el mundo.

Módulo 1. La revolución del software libre y de código abierto

Si miramos España, a lo largo de la década del 2000 las administraciones públicas comenzaron a interesarse por el software libre y comprender sus ventajas con respecto a las aplicaciones privativas, en parte gracias a la campaña de Hispalinux para promover el uso del software libre en la administración pública.

Roberto Rodríguez Echeverría (1997, p. 30) ha estudiado en detalle la penetración del software libre a nivel de la Unión Europea y de las diferentes administraciones españolas. Cuenta cómo Extremadura fue una de las comunidades autónomas pioneras, con la puesta en marcha en 2002 del proyecto gnuLinEx, que considera una “referencia obligada para otras iniciativas posteriores llevadas a cabo tanto a nivel nacional como internacional”. Cabe destacar también la constitución en 2006 del Centro Nacional de Referencia de Aplicación de las Tecnologías de Información y Comunicación (CENATIC), un organismo gubernamental para impulsar el conocimiento y el uso del “software de fuentes abiertas”, que no es otra cosa que el software libre.

Aunque la adopción del software libre en las administraciones públicas españolas es desigual, los datos manejados por el Observatorio Nacional del Software de Fuentes Abiertas, dependiente del CENATIC (2011), revelan que en 2011 9 de cada 10 organismos públicos tenían soluciones de software libre en sus servidores y que el 40% del software instalado en estos servidores era libre.

Una de las áreas donde más se está impulsando el software libre es en las universidades. Anualmente se realiza un ranking para medir el uso y la difusión de las soluciones no privativas en las universidades hispanoamericanas y españolas y las diez primeras universidades son españolas, encabezadas por la Universidad Miguel Hernández, la Universidad de Granada y la Universidad de La Laguna.

También las empresas españolas están abrazando las soluciones de software libre y de código abierto. Aunque los datos varían dependiendo del estudio que consultemos—del 75% de las empresas de más de 10 empleados, según un estudio del CENATIC de 2011, al 52% de las empresas, según un estudio publicado en 2017 por SofDoit y la Asociación de Técnicos de Informáticas—, vemos que la idea de que el software libre no vale para la empresa es otro falso mito. Más de 500 empresas forman parte de la Federación Nacional de Empresas de Software Libre (ASOLIF), y las ferias de referencia del software libre y de código abierto en España, OpenExpo y Librecon, crecen cada año.

Aunque aún queda mucho por hacer, el éxito de Android marca el final de una época en la que hemos visto derrumbarse muchos de los prejuicios sobre el software libre y consolidarse los modelos de negocio open source. Nos muestra también uno de los retos que tendrá que afrontar el software libre y de código abierto en los próximos años. El movimiento del software libre nació para defender la libertad de los usuarios, pero hoy tenemos aplicaciones y sistemas como Android que, a pesar de ser de código abierto, son utilizados por grandes empresas para monitorizar nuestra actividad. ¿Puede considerarse libre un programa que nos priva de nuestro derecho a la privacidad?

Canonical, la empresa detrás del desarrollo de Ubuntu, se vio forzada en 2016 a suprimir algunas funcionalidades de su sistema operativo después de ser duramente criticada por la comunidad y organizaciones como la FSF y la Electronic Frontier Foundation por guardar datos sobre las búsquedas que los usuarios realizaban desde la interfaz de Unity para mostrarles publicidad de Amazon a medida (Sneddon, 2016). Este caso nos enseña que una de las ventajas de la libertad de estudiar y modificar el software, base tanto del movimiento del software libre como del movimiento open source, es que permite a la comunidad detectar código malicioso en los programas y tomar medidas para proteger la privacidad de los usuarios.

Con todo, la mayoría de los usuarios de software libre—que ni sabe de programación ni está implicada en la comunidad—está expuesta a prácticas como las de Google y otras empresas que adoptan las ventajas competitivas del software libre y de código abierto, pero no la filosofía que tiene detrás. ¿Tiene razón Stallman (2004, p. 241) cuando critica al movimiento open source por citar únicamente los beneficios prácticos de este tipo de software y negar que compartir el código sea una cuestión de principios?

3. “Software libre” y “open source”: ¿son lo mismo?

El término “open source” cumplirá veinte años en 2018 y parte de la comunidad todavía discute si el software libre es lo mismo que el software de código abierto y cuál es la denominación más adecuada. La primera cuestión es muy fácil de responder: “software libre”, “open source”, “software de código abierto” y “software libre y de código abierto” se refieren exactamente a lo mismo. Si un software reúne las cuatro libertades, su código fuente es abierto; y, a la inversa, si el código fuente de un programa es abierto, el software reúne las cuatro libertades.



Diferencias entre el software libre y el software de código abierto.

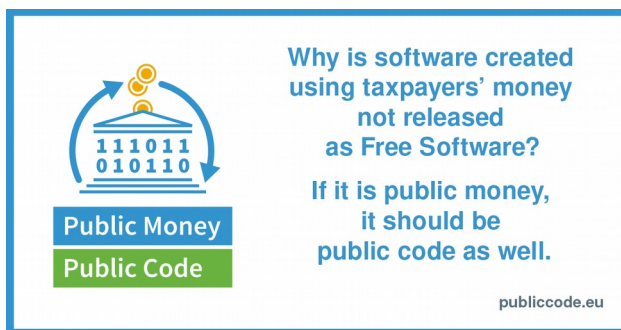
Fuente: CTIN (2016).

La denominación de “software de código abierto” fue adoptada a finales de los noventa por integrantes de la comunidad que pensaban que ayudaría a derribar los prejuicios que muchas empresas tenían al respecto del software libre. En lugar de hacer hincapié en la obligación moral de compartir el software, como hacía Stallman (recordemos sus palabras en el Manifiesto GNU), su discurso ponía en valor los beneficios prácticos de liberar el código fuente. Una parte de la comunidad aceptó con entusiasmo esta nueva denominación y otra, encabezada por la FSF, la rechazó y continuó utilizando la expresión “software libre”.

Módulo 1. La revolución del software libre y de código abierto

A día de hoy sigue existiendo una rama dentro del software libre y de código abierto más activista, que lo promueve por principios y trabaja también temas como la privacidad, el DRM (del que hablaremos en el último módulo del curso) o la promoción del software libre en las administraciones públicas (*Public Money, Public Code*), y una rama más pragmática, que libera el código porque da mejores resultados. Ambos enfoques son complementarios (este tipo de software respeta la libertad de los usuarios y es más ventajoso) y, de hecho, el propio Stallman (2004, p. 242) reconoce que “el movimiento open source ha ayudado considerablemente a nuestra comunidad y trabajamos juntos en proyectos prácticos”.

Cada uno puede utilizar la denominación con la que se sienta más cómodo o identificado. Muchos empleamos indistintamente estas denominaciones porque pensamos que el debate terminológico está superado y que resultan más determinantes cuestiones como que la licencia asegure que las versiones derivadas sigan siendo libres, el código respete la privacidad de los usuarios y el fabricante no utilice el software libre únicamente como reclamo comercial para vender una versión privativa de pago. Esto nos lleva directamente al siguiente punto de este tema: los modelos de negocio con software libre y de código abierto.



Dos campañas recientes de la Free Software Foundation.
Fuente: publiccode.eu y defectivebydesign.org.

4. Modelos de negocio con software libre y de código abierto

Muchas veces no se entiende bien por qué a un fabricante de software le sale rentable compartir el código que tantas horas de trabajo le ha costado desarrollar y se tiende a pensar que quizá el programa no es lo suficientemente bueno para comercializarse.

Circula por internet una lista de 16 modelos de negocio open source elaborada por Groen y Maduro (2012). Ésta nos puede dar una idea de cuáles son las fórmulas más habituales para generar dinero gracias al software libre o financiar el desarrollo de una nueva aplicación no privativa, que podemos clasificar en tres grandes categorías: venta de servicios, modelo open core y desarrollo colaborativo.

4.1. Ventas de servicios

La venta de servicios adicionales es la estrategia más extendida en el sector y consiste, como su propio nombre indica, en vender al usuario servicios extra como instalación y configuración de la herramienta, consultoría, soporte técnico, formación, desarrollos a medida, etc. Estos servicios van dirigidos tanto a organizaciones pequeñas, que no cuentan con un equipo de IT propio para realizar estas tareas, como para organizaciones más grandes, que necesitan expertos en la herramienta para integrarla con el resto de sus aplicaciones y necesitan un soporte técnico dedicado.

Una forma cada vez más popular de hacer rentable un proyecto de software libre es ofrecer al cliente la posibilidad de utilizarlo en la nube a cambio de una suscripción. En este modelo, también conocido como SaaS (Software as a Service), el proveedor está cobrando la instalación y configuración del programa, el almacenamiento de

Módulo 1. La revolución del software libre y de código abierto

los datos, el hardware, la administración del servidor, la actualización del software, además de otros servicios que se acuerden entre el cliente y el proveedor. Es una fórmula atractiva para organizaciones que quieren probar el software antes de decidirse a desplegarlo en sus propios servidores, con el coste que eso conlleva, y para organizaciones que no cuentan con recursos para administrar un servidor, garantizar la seguridad de su información, etc.

4.2. Modelo open core

Otra forma de hacer rentable una aplicación de código abierto es ofrecer una versión premium, profesional o enterprise con funcionalidades adicionales a la versión community. Este contenido de pago puede consistir en add-ons, acceso anticipado a actualizaciones, forks que mejoran la experiencia de usuario, etc. En este modelo de negocio, el núcleo del programa es de código abierto, pero las capas que se le agregan encima no lo son.

Este modelo es común en fabricantes que están realizando una transición del software privativo al software libre y aún no confían plenamente en las posibilidades de un modelo de negocio 100% open source. Estas empresas suelen ofrecer una versión de libre descarga con contenido recortado como “gancho” para que los clientes prueben el programa y paguen por pasar a la versión completa, por lo que muchos consideran que no podemos hablar con propiedad de que sigan un modelo open source. Un caso típico en el mercado de los ECM es Alfresco.

Aunque compartamos esta crítica, hay que distinguir entre los fabricantes que utilizan el código abierto como estrategia de marketing y aquellos que financian el desarrollo de nuevo código ofreciendo acceso anticipado al mismo mediante una suscripción de pago, pero lo liberan pasado un tiempo.

4.3. Desarrollo colaborativo

Puede ocurrir que varias empresas tengan la misma necesidad y colaboren entre ellas para desarrollar una tecnología que permita resolverla. También es frecuente que una empresa usuaria de una determinada aplicación de código abierto contribuya a que sea mejor, ya sea financiando su desarrollo o dedicando tiempo de sus propios desarrolladores a mejorar el código. O que una empresa haya desarrollado un programa y decida compartir el código en comunidades como GitHub para beneficiarse de las mejoras que realizan otros desarrolladores.

Ejemplos no faltan de cualquiera de estas situaciones. Cada vez más organizaciones abrazan este modelo porque se han percatado de que desarrollar software de manera colaborativa es más rentable que hacerlo de manera competitiva. Esto no significa que estas empresas hayan dejado de competir entre ellas; sin embargo, han entendido que no deben competir por tener el mejor código, sino por ser quienes mejor lo sepan aprovechar para dar valor a sus clientes.

Resumen

No es la gratuidad lo que define el software libre, sino la libertad de ejecutar, estudiar, distribuir y mejorar el programa (las llamadas cuatro libertades del software libre).

Es posible dividir la historia del software libre en tres grandes etapas:

1. Orígenes del movimiento del software libre (1976-1991).
2. Los primeros sistemas operativos libres (1991-1998).
3. Consolidación del software libre y de código abierto (1998-actualidad).

Existe una rama dentro del software libre y de código abierto más activista, que se identifica con el término “software libre”, y una rama más pragmática, que se identifica con el término “código abierto”.

Los modelos de negocio basados en software libre y de código abierto se clasifican en tres categorías:

1. Venta de servicios.

Módulo 1. La revolución del software libre y de código abierto

2. Modelo open core.
3. Desarrollo colaborativo.

Bibliografía

- BALTER, B. "Open source license usage on GitHub.com". En: *GitHub* [en línea]. 9 marzo 2015 [consulta 05-05-2017]. Disponible en: <https://github.com/blog/1964-open-source-license-usage-on-github-com>.
- BRETTTHAUER, D. "Open source software: A history". En: *Published Works* [en línea]. 2001, n. 7 [consulta 05-05-2017]. Disponible en: http://digitalcommons.uconn.edu/libr_pubs/7/.
- CENATIC. "El software libre en los organismos públicos del ámbito estatal: informe de resultados de la encuesta sobre el software de fuentes abiertas en la administración pública (ESFA-AGE) 2011". En: *CENATIC* [en línea]. 4 enero 2012 [consulta 05-05-2017]. Disponible en: <http://www.cenatic.es/dossier/panel-informe-age2011>.
- CTIN. "Free Software Vs Open Source". En: *Facebook* [en línea]. 13 julio 2016 [consulta 05-05-2017]. Disponible en: <https://www.facebook.com/ctinmexico/photos/a.112767675549967.20561.112489032244498/613369235489806/?type=3&theater>.
- DAFFARA, C. "Open source license selection in relation to business models". En: *Technology Innovation Management Review* [en línea]. Febrero 2011 [consulta 05-05-2017]. Disponible en: <https://timreview.ca/article/416>.
- GATES, B. "An open letter to hobbyists". En: *Computer Notes* [en línea]. Febrero 1976, v. 1, n. 9, p. 3 [consulta 05-05-2017]. Disponible en: https://web.archive.org/web/20120323162213/http://startup.nmnaturalhistory.org/gallery/notesViewer.php?ii=76_2&p=3.
- GROEN, P., MADURO, R. A. "Open source business models – A more in depth view". En: *Open Health News* [blog]. 23 diciembre 2012 [consulta 05-05-2017]. Disponible en: <http://www.openhealthnews.com/articles/2012/open-source-business-models-more-depth-view>.
- Public Money, Public Code* [web]. [Consulta 11-09-2017]. Disponible en: <https://publiccode.eu/>.
- RAYMOND, E. 1998. *La catedral y el bazar* [en línea]. 1998 [consulta 05-05-2017]. Disponible en: <http://biblioweb.sindominio.net/telematica/catedral.html>.
- RODRÍGUEZ Echeverría, R. *Penetración del software libre en la Administración Extremeña* [en línea]. Noviembre 2007 [consulta 05-05-2017]. Disponible en: https://observatorio.iti.upv.es/media/managed_files/2009/02/04/12penetracionflossaappextremadura.pdf.
- SNEDDON, J. "Ubuntu 'spyware' will be disabled in Ubuntu 16.04 LTS". En: *OMG! Ubuntu!* [blog]. 8 enero 2016 [consulta 05-05-2017]. Disponible en: <http://www.omgubuntu.co.uk/2016/01/ubuntu-online-search-feature-disabled-16-04>.
- STALLMAN, R. "El manifiesto de GNU". En: *El sistema operativo GNU y el movimiento del software libre* [web]. 1985 [consulta 05-05-2017]. Disponible en: <https://www.gnu.org/gnu/manifesto.es.html>.
- *Software libre para una sociedad libre* [en línea]. Madrid: Traficantes de Sueños, 2004 [consulta 05-05-2017]. 317 pp. ISBN: 84-933555-18. Disponible en: https://www.gnu.org/philosophy/fsfs/free_software.es.pdf.
- The Halloween Papers* [web]. [Consulta 05-05-2017]. Disponible en: <http://www.catb.org/ěsr/halloween/>.
- TORVALDS, L. "What would you like to see most in minix?". En: *comp.os.minix* [en línea]. 25 agosto 1991 [consulta 05-05-2017]. Disponible en: <https://groups.google.com/forum/#!original/comp.os.minix/dlNtH7RRrGA/SwRavCzVE7g>.
- "The Linux Edge". En: *Open Sources: Voices from the open source revolution*. O'Reilly Media, 1999. 280 pp. ISBN: 1-56592-582-3.

Módulo 1. La revolución del software libre y de código abierto

ZIZEK, S. *En defensa de las causas perdidas*. Madrid: Ediciones Akal, 2016. 479 pp. ISBN: 978-84-460-2957-1.