

2. Un poco de historia

"When I started working at the MIT Artificial Intelligence Lab in 1971, I became part of a software-sharing community that had existed for many years. Sharing of software was not limited to our particular community; it is as old as computers, just as sharing of recipes is as old as cooking. But we did it more than most. [...] We did not call our software *free software*, because that term did not yet exist; but that is what it was. Whenever people from another university or a company wanted to port and use a program, we gladly let them. If you saw someone using an unfamiliar and interesting program, you could always ask to see the source code, so that you could read it, change it, or cannibalize parts of it to make a new program."

"Cuando comencé a trabajar en el Laboratorio de Inteligencia Artificial del MIT en 1971, pasé a formar parte de una comunidad de software compartido que existía desde hacía muchos años. Compartir código no era algo específico de nuestra comunidad: es algo tan antiguo como los ordenadores, como compartir recetas es tan viejo como cocinar. Pero nosotros lo hacíamos más que la mayoría. [...] No llamábamos a nuestro software *software libre* porque ese término aún no existía, pero eso es lo que era. Cuando alguien de otra universidad o de una empresa quería portar y usar un programa, nosotros le dejábamos hacerlo con gusto. Si veías a alguien utilizando un programa raro e interesante, siempre podías pedirle ver el código fuente, para poder leerlo, cambiarlo o canibalizar partes para hacer un programa nuevo."

Richard Stallman, "The GNU Project" (publicado originalmente en el libro *Open sources*) [208]

Aunque todas las historias relacionadas con la informática son forzosamente breves, la del software libre es una de las más largas entre ellas. De hecho, podría decirse que en sus comienzos, prácticamente todo el software desarrollado cumplía con las definiciones de *software libre*, aunque el concepto ni siquiera existía aún. Más tarde la situación cambió por completo, y el software privativo dominó la escena, prácticamente en exclusiva, durante bastante tiempo. Fue durante esta época cuando se sentaron las bases del software libre como lo entendemos hoy en día, y cuando, poco a poco, empezaron a aparecer programas libres. Con el tiempo, estos comienzos se han convertido en una tendencia que ha ido creciendo y madurando hasta llegar a la situación actual, en la que el software libre es una posibilidad que hay que considerar en casi todos los ámbitos.

Esta historia es bastante desconocida, hasta tal punto que para gran parte de los profesionales informáticos el software privativo es el software "en su estado natural". Sin embargo, la situación es más bien la contraria, y las semillas del cambio que se empezó a entrever en la primera década de 2000 habían sido plantadas ya a principios de los años ochenta.

Bibliografía

No hay muchas historias exhaustivas sobre el software libre, y las que hay son artículos normalmente limitados al objeto de su estudio. En cualquier caso, el lector interesado puede completar lo expuesto en este capítulo consultando "Open Source Initiative. History of the OSI" [146] (<http://www.opensource.org/docs/history.php>), que pone el énfasis en el impacto del software libre en el mundo empresarial entre los años 1998 y 1999; "A brief history of free/open source software movement" [190], de Chris Rasch, que cubre la historia del software libre hasta el año 2000, o "The origins and future of open source software" (1999) [177], de Nathan Newman, que se centra en gran medida en la promo-

ción indirecta que el Gobierno de EE.UU. ha hecho del software libre o sistemas similares durante las décadas de 1970 y 1980.

2.1. El software libre antes del software libre

El software libre como concepto no apareció hasta principios de la década de 1980. Sin embargo, su historia puede trazarse desde bastantes años antes.

2.1.1. Y en el principio fue libre

Durante los años sesenta el panorama de la informática estaba dominado por los grandes ordenadores, instalados fundamentalmente en empresas y centros gubernamentales. IBM era el principal fabricante, con gran diferencia sobre sus competidores. En esta época, cuando se adquiría un ordenador (el hardware), el software venía como un acompañante. Mientras se pagase el contrato de mantenimiento, se tenía acceso al catálogo de software que ofrecía el fabricante. Además, no era común la idea de que los programas fueran algo "separado" desde un punto de vista comercial.

En esta época el software se distribuía habitualmente junto con su código fuente (en muchos casos sólo como código fuente), y en general, sin restricciones prácticas. Los grupos de usuarios como SHARE (usuarios de sistemas IBM) o DECUS (usuarios de DEC) participaban de estos intercambios, y hasta cierto punto, los organizaban. La apartado "Algorithms" de la revista *Communications of the ACM* era otro buen ejemplo de foro de intercambio. Podría decirse que durante estos primeros años de la informática el software era libre, al menos en el sentido de que los que tenían acceso a él podían disponer habitualmente del código fuente, estaban acostumbrados a compartirlo, a modificarlo y a compartir también estas modificaciones.

El 30 de junio de 1969 IBM anunció que, a partir de 1970, iba a vender parte de su software por separado (Burton Grad, 2002) [131]. Ello supuso que sus clientes ya no podían obtener, incluido en el precio del hardware, los programas que necesitaban. El software se comenzó a percibir como algo con valor intrínseco y, como consecuencia, se hizo cada vez más habitual restringir escrupulosamente el acceso a los programas y se limitaron, en la medida de lo posible (tanto técnica como legalmente), las posibilidades de los usuarios para compartir, modificar o estudiar el software. Dicho de otro modo, se pasó a la situación que sigue siendo habitual en el mundo del software a principios del siglo XXI.

Bibliografía

El lector interesado en esta época de transición puede leer, por ejemplo, "How the ICP Directory began" [226] (1998), donde Larry Welke comenta cómo nació uno de los primeros catálogos de software no ligados a un fabricante, y cómo en este proceso descubrió que las empresas estaban dispuestas a pagar por programas no hechos por el fabricante de sus ordenadores.

A mediados de la década de 1970 era ya absolutamente habitual, en cualquier ámbito informático, encontrarse con software privativo. Ello supuso un gran cambio cultural entre los profesionales que trabajaban con software y fue el origen del florecimiento de un gran número de empresas dedicadas al nuevo negocio. Faltaba aún casi una década para que empezase a aparecer, de forma organizada y como reacción a esta situación, lo que hoy conocemos como *software libre*.

2.1.2. Década de los setenta y principios de la década de los ochenta

Incluso cuando la tendencia abrumadoramente mayoritaria era explorar el modelo de software privativo, había iniciativas que mostraban algunas características de lo que luego se consideraría software libre. De hecho, alguna de ellas llegó a producir software libre tal como lo definimos hoy en día. Entre ellas cabe destacar SPICE, TeX y Unix, cuyo caso es mucho más complejo.

SPICE (Simulation Program with Integrated Circuit Emphasis) es un programa desarrollado en la Universidad de California, en Berkeley, para simular las características eléctricas de un circuito integrado. Fue desarrollado y puesto en el dominio público por su autor, Donald O. Pederson, en 1973. SPICE fue originalmente una herramienta docente, y como tal se extendió rápidamente a muchas universidades de todo el mundo. En ellas fue usado por muchos estudiantes de la por aquel entonces incipiente disciplina del diseño de circuitos integrados. Puesto que estaba en el dominio público, SPICE podía redistribuirse, modificarse, estudiarse... Incluso se podía adaptar a unas necesidades concretas y vender esa versión como producto privativo (lo que han hecho durante su historia docenas de veces una gran cantidad de empresas). Con estas características, SPICE tenía todas las papeletas para convertirse en el estándar de la industria, con sus diferentes versiones. Y efectivamente, eso fue lo que ocurrió. Probablemente éste fue el primer programa con características de software libre que durante un tiempo copó un mercado, el de los simuladores de circuitos integrados, y sin duda pudo hacerlo precisamente por tener estas características (además de sus innegables cualidades técnicas).

Bibliografía

Se puede consultar más información sobre la historia de SPICE en "The life of SPICE", trabajo presentado en el 1996 Bipolar Circuits and Technology Meeting, Minneapolis, MN, EE.UU., en septiembre de 1996 [175].

La página web de SPICE puede encontrarse en <http://bwrc.eecs.berkeley.edu/Classes/Ic-Book/SPICE/>.

Donald Knuth comenzó a desarrollar TeX durante un año sabático, en 1978. TeX es un sistema de tipografía electrónica muy utilizado para la producción de documentos de calidad. Desde el principio, Knuth utilizó una licencia que hoy sería considerada como de software libre. Cuando el sistema se consideró

razonablemente estable, en 1985, mantuvo esa licencia. En esa época, TeX era uno de los sistemas más grandes y más conocidos que se podía considerar software libre.

Bibliografía

Algunos hitos de la historia de TeX pueden consultarse en línea en <http://www.math.utah.edu/software/plot79/tex/history.html> [39]. Para saber más detalles, también es útil el artículo correspondiente de Wikipedia, <http://www.wikipedia.org/wiki/TeX> [233].

2.1.3. Desarrollo temprano de Unix

Unix, uno de los primeros sistemas operativos portables, fue creado originalmente por Thompson y Ritchie (entre otros) en los Bell Labs de AT&T. Su desarrollo ha continuado desde su nacimiento, hacia 1972, dando lugar a innumerables variantes comercializadas por (literalmente) decenas de empresas.

Durante los años 1973 y 1974, Unix llegó a muchas universidades y centros de investigación de todo el mundo, con una licencia que permitía su uso para fines académicos. Aunque había ciertas restricciones que impedían su libre distribución, entre las organizaciones que disponían de la licencia el funcionamiento fue muy similar al que se vio más tarde en muchas comunidades de software libre. Los que tenían acceso al código fuente de Unix tuvieron un sistema que podían estudiar, mejorar y ampliar. Alrededor de él apareció una comunidad de desarrolladores que pronto empezó a girar en torno al CSRG de la Universidad de California, en Berkeley. Esta comunidad desarrolló su propia cultura, que como veremos más adelante, fue muy importante en la historia del software libre. Unix fue, hasta cierto punto, un ensayo temprano de lo que se vio con GNU y Linux varios años más tarde. Estaba confinado a una comunidad mucho más pequeña y era necesaria la licencia de AT&T, pero en otros aspectos su desarrollo fue similar (en un mundo mucho menos comunicado).

Modos de desarrollo propios del software libre

En *Netizens. On the history and impact of Usenet and the Internet* (IEEE Computer Society Press, 1997 [139], página 139) pueden leerse unas líneas que se podrían referir a muchos proyectos libres: "Las contribuciones al valor de Unix durante su desarrollo temprano fueron muchas, gracias al hecho de que el código fuente estaba disponible. Podía ser examinado, mejorado y personalizado".

En la página 142 de la misma obra se dice lo siguiente: "Los pioneros como Henry Spencer están de acuerdo en lo importante que fue para los que pertenecían a la comunidad Unix tener el código fuente. él resalta cómo la disposición de los códigos fuente hacía posible la identificación y la reparación de los errores que descubrían. [...] Incluso en los últimos 1970 y primeros 1980, prácticamente cada sitio Unix tenía fuentes completas".

Aún más explícito es el texto de Marc Rochkind "Interview with dick haight" (*Unix Review*, mayo 1986) [198]: "ésta era una de las grandes cosas sobre Unix en los primeros días: la gente realmente compartía con los demás. [...] No sólo aprendíamos mucho aquellos días del material compartido, sino que nunca teníamos que preocuparnos sobre cómo funcionaban realmente las cosas porque siempre podíamos ir a leer el código fuente".

Con el tiempo, Unix fue también un ejemplo temprano de los problemas que podían presentar los sistemas privativos que a primera vista "tenían alguna característica del software libre". A finales de la década de 1970, y sobre todo durante la de 1980, AT&T cambió su política, y el acceso a nuevas versiones de Unix se convirtió en algo difícil y caro. La filosofía de los primeros años, que había hecho tan popular a Unix entre los desarrolladores, cambió radicalmente hasta tal punto que en 1991 AT&T puso una demanda a la Universidad de Berkeley por publicar el código de Unix BSD que el CSRG de Berkeley había creado. Pero ésa es otra historia, que retomaremos más adelante.

2.2. El comienzo: BSD, GNU

Todos los casos descritos en el apartado anterior fueron iniciativas individuales o no cumplían estrictamente los requisitos del software libre. Hasta principios de la década de 1980 no aparecieron, de forma organizada y consciente, los primeros proyectos para la creación de sistemas compuestos de software libre. En esa época se empezaron también a fijar (lo que probablemente es más importante) los fundamentos éticos, legales e incluso económicos, que se han ido desarrollando y completando hasta hoy en día. Y como el nuevo fenómeno necesitaba un nombre, durante esos años se acuñó también el propio término *software libre*.

2.2.1. Richard Stallman, GNU, FSF: nace el movimiento del software libre

A principios de 1984, Richard Stallman, en aquella época empleado en el AI Lab del MIT, abandonó su trabajo para comenzar el proyecto GNU. Stallman se consideraba un *hacker* de los que gozaban compartiendo sus inquietudes tecnológicas y su código. Veía con desagrado cómo su negativa a firmar acuerdos de exclusividad y de no compartición le estaban convirtiendo en un extraño en su propio mundo, y cómo el uso de software privativo en su entorno le dejaba impotente ante situaciones que antes podía solventar fácilmente.

Su idea al abandonar el MIT era construir un sistema de software completo, de propósito general, pero totalmente libre ("The GNU Project", DiBona *et al.*) [208]. El sistema (y el proyecto que se encargaría de hacerlo realidad) se llamó GNU (acrónimo recursivo, "GNU's not Unix"). Aunque desde el principio el proyecto GNU incluyó en su sistema software ya disponible (como TeX o, más adelante, el sistema X Window), había mucho que construir. Richard Stallman comenzó por escribir un compilador de C (GCC) y un editor (Emacs), ambos aún en uso (y muy populares).

Desde el principio del proyecto GNU, Richard Stallman se preocupó por las libertades que tendrían los usuarios de su software. Estaba interesado en que no sólo los que recibieran los programas directamente del proyecto GNU, sino cualquiera que lo recibiera después de cualquier número de redistribuciones y (quizás) modificaciones, siguiera pudiendo disfrutar de los mismos de-

rechos (modificación, redistribución, etc.). Para ello escribió la licencia GPL, probablemente la primera licencia de software diseñada específicamente para garantizar que un programa fuera libre en este sentido. Al mecanismo genérico que utilizan las licencias de tipo GPL para conseguir estas garantías, Richard Stallman lo llamó *copyleft*, que sigue siendo el nombre de una gran familia de licencias de software libre (Free Software Foundation, GNU General Public License, versión 2, junio 1991) [118].

Richard Stallman también fundó la Free Software Foundation (FSF) para conseguir fondos, que dedica al desarrollo y la protección del software libre, y sentó sus fundamentos éticos con "The GNU Manifesto" (Free Software Foundation, 1985) [117] y "Why software should not have owners" (Richard Stallman, 1998) [207].

Desde el punto de vista técnico, el proyecto GNU fue concebido como un trabajo muy estructurado y con metas muy claras. El método habitual estaba basado en grupos relativamente pequeños de personas (normalmente voluntarios) que desarrollaban alguna de las herramientas que luego encajarían perfectamente en el rompecabezas completo (el sistema GNU). La modularidad de Unix, en la que se inspiraba el desarrollo de este proyecto, coincidía totalmente con esta idea. El método de trabajo generalmente implicaba el uso de Internet, pero ante su escasa implantación en aquellos días, la Free Software Foundation también vendía cintas en las que grababa las aplicaciones, de manera que fue probablemente una de las primeras organizaciones que se benefició económicamente (aunque de manera bastante limitada) de la creación de software libre.

A principios de la década de 1990, unos seis años después de su nacimiento, el proyecto GNU estaba muy cerca de tener un sistema completo similar a Unix. Aun así, hasta ese momento aún no había producido una de las piezas fundamentales: el núcleo del sistema (también llamado *kernel*, la parte del sistema operativo que se relaciona con el hardware, lo abstrae, y permite que las aplicaciones compartan recursos y, en el fondo, funcionen). Sin embargo, el software de GNU era muy popular entre los usuarios de las distintas variantes de Unix, por aquel entonces el sistema operativo más usado en las empresas. Además, el proyecto GNU había conseguido ser relativamente conocido entre los profesionales informáticos, y muy especialmente entre los que trabajaban en universidades. En esa época, sus productos ya gozaban de una merecida reputación de estabilidad y calidad.

2.2.2. El CSRG de Berkeley

El CSRG (Computer Science Research Group) de la Universidad de California, en Berkeley, fue desde 1973 uno de los centros donde más desarrollo relacionado con Unix se hizo durante los años 1979 y 1980. No sólo se portaron aplicaciones y se construyeron otras nuevas para su funcionamiento sobre Unix, sino que se llevaron a cabo importantes mejoras en el núcleo y se le añadió

mucha funcionalidad. Por ejemplo, durante la década de los ochenta, varios contratos de DARPA (dependiente del Ministerio de Defensa de EE.UU.) financiaron la implementación de TCP/IP que ha sido considerada hasta nuestros días como la referencia de los protocolos que hacen funcionar Internet (vinculando, de paso, el desarrollo de Internet y la expansión de las estaciones de trabajo con Unix). Muchas empresas utilizaron como base de sus versiones de Unix los desarrollos del CSRG dando lugar a sistemas muy conocidos en la época, como SunOS (Sun Microsystems) o Ultrix (Digital Equipment). De esta manera, Berkeley se convirtió en una de las dos fuentes fundamentales de Unix, junto con la "oficial", AT&T.

Para poder utilizar todo el código que producía el CSRG (y el de los colaboradores de la comunidad Unix que ellos de alguna manera coordinaban), hacía falta la licencia de Unix de AT&T, que cada vez era más difícil (y más cara) de conseguir, sobre todo si se quería el acceso al código fuente del sistema. Tratando de evitar en parte este problema, en junio de 1989 el CSRG liberó la parte de Unix relacionada con TCP/IP (la implementación de los protocolos en el núcleo y las utilidades), que no incluía código de AT&T. Fue la llamada *Networking Release 1* (Net-1). La licencia con que se liberó fue la famosa *licencia BSD*, que salvo ciertos problemas con sus cláusulas sobre obligaciones de anuncio, ha sido considerada siempre un ejemplo de licencia libre minimalista (que además de permitir la redistribución libre, permitía también su incorporación a productos privativos). Además, el CSRG probó un novedoso método de financiación (que ya estaba probando con éxito la FSF): vendía cintas con su distribución por 1.000 dólares. Aunque cualquiera podía a su vez redistribuir el contenido de las cintas sin ningún problema (ya que la licencia lo permitía), el CSRG vendió cintas a miles de organizaciones, con lo que consiguió fondos para seguir desarrollando.

Viendo el éxito de la distribución de Net-1, Keith Bostic propuso reescribir todo el código que aún quedaba del Unix original de AT&T. A pesar del escepticismo de algunos miembros del CSRG, realizó un anuncio público en el que pedía ayuda para realizar esta tarea, y poco a poco las utilidades (reescritas a partir de especificaciones) comenzaron a llegar a Berkeley. Mientras tanto, se realizó el mismo proceso con el núcleo, de manera que se reescribió de forma independiente la mayor parte del código que no habían realizado Berkeley ni colaboradores voluntarios. En junio de 1991, después de conseguir el permiso de la administración de la Universidad de Berkeley, se distribuyó la *Networking Release 2* (Net-2), con casi todo el código del núcleo y todas las utilidades de un sistema Unix completo. De nuevo el conjunto se distribuyó bajo la licencia BSD y se vendieron miles de cintas al precio de 1.000 dólares la unidad.

Sólo seis meses después de la liberación de Net-2, Bill Jolitz escribió el código que faltaba en el núcleo para que funcionase sobre arquitectura i386, liberando 386BSD, que fue distribuida por Internet. A partir de este código surgieron más tarde, en sucesión, todos los sistemas de la familia *BSD: primero apareció NetBSD, como una recopilación de los parches que se habían ido aportan-

do en la Red para mejorar 386BSD; más adelante apareció FreeBSD, como un intento de soportar fundamentalmente la arquitectura i386; varios años más tarde se formó el proyecto OpenBSD, con énfasis en la seguridad. Y también hubo una distribución propietaria basada en Net-2 (aunque era ciertamente original, ya que ofrecía a sus clientes todo el código fuente como parte de la distribución básica), que realizó de forma independiente la desaparecida empresa BSDI (Berkeley Software Design Inc.).

En parte como reacción a la distribución hecha por BSDI, Unix System Laboratories (USL), subsidiaria de AT&T que tenía los derechos de la licencia de Unix, puso una demanda judicial, primero a BSDI y luego a la Universidad de California. En ella los acusaba de distribuir su propiedad intelectual sin permiso. Después de varias maniobras judiciales (que incluyeron una contrademanda de la Universidad de California contra USL), Novell compró los derechos de Unix a USL, y en enero de 1994 llegó a un acuerdo extrajudicial con la Universidad de California. Como resultado de este acuerdo, el CSRG distribuyó la versión 4.4BSD-Lite, que pronto fue utilizada por todos los proyectos de la familia *BSD. Poco después (tras liberar aún la versión 4.4BSD-Lite Release 2), el CSRG desapareció. En ese momento hubo quien temió que fuera el fin de los sistemas *BSD, pero el tiempo ha demostrado que siguen vivos y coleando, con una nueva forma de gestión más típica de proyectos libres. Aún en la década de 2000 los proyectos que gestionan la familia *BSD son de los más antiguos y consolidados en el mundo del software libre.

Bibliografía

La historia de Unix BSD es muy ilustrativa de una forma peculiar de desarrollar software durante los años setenta y ochenta. Quien esté interesado en ella puede disfrutar de la lectura de "Twenty years of Berkeley Unix" (Marshall Kirk McKusick, 1999) [170], en la que se puede seguir su evolución desde la cinta que llevó Bob Fabry a Berkeley con la idea de hacer funcionar en un PDP-11 una de las primeras versiones del código de Thompson y Ritchie (comprada conjuntamente por los departamentos de informática, estadística y matemáticas), hasta las demandas judiciales de AT&T y las últimas liberaciones de código que dieron lugar a la familia de sistemas operativos libres *BSD.

2.2.3. Los comienzos de Internet

Casi desde su nacimiento, a principios de la década de 1970, Internet tuvo mucha relación con el software libre. Por un lado, desde sus comienzos, la comunidad de desarrolladores que la construyeron tuvo claros varios principios que luego se harían clásicos en el mundo del software libre; por ejemplo, la importancia de que los usuarios puedan ayudar a depurar errores o la compartición de código. La importancia de BSD Unix en su desarrollo (al proporcionar durante los años ochenta la implementación más popular de los protocolos TCP/IP) hizo que muchas costumbres y formas de funcionamiento pasaran fácilmente de una comunidad, la de desarrolladores alrededor del CSRG, a otra, la de los que estaban construyendo lo que entonces era NSFNet y luego sería Internet, y viceversa. Muchas de las aplicaciones básicas en el desarrollo

de Internet, como Sendmail (servidor de correo) o BIND (implementación del servicio de nombres) fueron libres y, en gran medida, fruto de esta colaboración entre comunidades.

Por último, a finales de los años ochenta y en la década de los noventa, la comunidad del software libre fue una de las primeras que exploró hasta el fondo las nuevas posibilidades que permitía Internet para la colaboración entre grupos geográficamente dispersos. Esta exploración hizo posible, en gran medida, la propia existencia de la comunidad BSD, la FSF o el desarrollo de GNU/Linux.

Uno de los aspectos más interesantes del desarrollo de Internet, desde el punto de vista del software libre, fue la gestión completamente abierta de sus documentos y sus normas. Aunque hoy pueda parecer algo normal (pues es lo habitual, por ejemplo, en el IETF o en el World Wide Web Consortium), en su época la libre disposición de todas las especificaciones y documentos de diseño, incluidas las normas que definen los protocolos, fue algo revolucionario y fundamental para su desarrollo. En *Netizens. On the history and impact of Usenet and the Internet* [139] (página 106) podemos leer:

"Este proceso abierto promovía y llevaba al intercambio de información. El desarrollo técnico tiene éxito sólo cuando se permite que la información fluya libre y fácilmente entre las partes interesadas. La promoción de la participación es el principio fundamental que hizo posible el desarrollo de la Red."

Podemos observar cómo este párrafo podría ser suscrito, casi con toda seguridad, por cualquier desarrollador al referirse al proyecto de software libre en el que colabora.

En otra cita, en "The evolution of packet switching" [195] (página 267) podemos leer:

"Como ARPANET era un proyecto público que conectaba muchas de las principales universidades e instituciones de investigación, los detalles de implementación y rendimiento se publicaban ampliamente."

Obviamente, esto es lo que suele ocurrir en los proyectos de software libre, donde toda la información relacionada con el proyecto (y no sólo la implementación) suele ser pública.

En este ambiente, y antes de que Internet, ya bien entrados los años noventa, se convirtiese sobre todo en un negocio, la comunidad de usuarios y su relación con los desarrolladores era clave. En aquella época muchas organizaciones aprendieron a confiar no en una única empresa proveedora del servicio de comunicación de datos, sino en una compleja combinación de empresas de servicios, fabricantes de equipos, desarrolladores profesionales y voluntarios, etc. Las mejores implementaciones de muchos programas no eran las que venían con el sistema operativo que se compraba con el hardware, sino implementaciones libres que rápidamente las sustituían. Los desarrollos más inno-

vadores eran fruto no de grandes planes de investigación en empresas, sino de estudiantes o profesionales que probaban ideas y recogían la realimentación que les enviaban cientos de usuarios de sus programas libres.

Como ya se ha dicho, Internet también proporcionó al software libre las herramientas básicas para colaborar a distancia. El correo electrónico, los grupos de noticias, los servicios de FTP anónimo (que fueron los primeros almacenes masivos de software libre) y, más tarde, los sistemas de desarrollo integrados basados en web han sido fundamentales (e imprescindibles) para el desarrollo de la comunidad del software libre tal como la conocemos y, en particular, para el funcionamiento de la inmensa mayoría de los proyectos de software libre. Desde el principio, proyectos como GNU o BSD hicieron un uso masivo e intenso de todos estos mecanismos, desarrollando, a la vez que las usaban, nuevas herramientas y sistemas que a su vez mejoraban Internet.

Bibliografía

El lector interesado en una historia de la evolución de Internet, escrita por varios de sus protagonistas, puede consultar "A brief history of the Internet" (comunicación de la ACM, 1997) [166].

2.2.4. Otros proyectos

Durante la década de 1980 vieron la luz otros importantes proyectos libres. Entre ellos destaca, por su importancia y proyección futura, el X Window (sistema de ventanas para sistemas de tipo Unix), desarrollado en el MIT, que fue uno de los primeros ejemplos de financiación a gran escala de proyectos libres con recursos de un consorcio de empresas. También merece la pena mencionar Ghostscript, un sistema de gestión de documentos PostScript desarrollado por una empresa, Aladdin Software, que fue uno de los primeros casos de búsqueda de un modelo de negocio basado en la producción de software libre.

A finales de los años ochenta hay ya en marcha toda una constelación de pequeños (y no tan pequeños) proyectos libres. Todos ellos, junto con los grandes proyectos mencionados hasta aquí, sentaron las bases de los primeros sistemas libres completos, que aparecieron a principios de la década de 1990.

2.3. Todo en marcha

Hacia 1990, gran parte de los componentes de un sistema informático completo estaban ya listos como software libre. Por un lado, el proyecto GNU y las distribuciones BSD habían completado la mayor parte de las aplicaciones que componen un sistema operativo. Por otro, proyectos como X Window o el propio GNU habían construido desde entornos de ventanas hasta compiladores, que muchas veces estaban entre los mejores de su género (por ejemplo, muchos administradores de sistemas SunOS o Ultrix sustituían para sus usuarios las aplicaciones propietarias de su sistema por las versiones libres de GNU o de BSD). Para tener un sistema completo construido sólo con software libre faltaba únicamente un componente: el núcleo. Dos esfuerzos separados e independientes vinieron a rellenar este hueco: 386BSD y Linux.

2.3.1. En busca de un núcleo

A finales de los ochenta y principios de los noventa, el proyecto GNU contaba con una gama básica de utilidades y herramientas que permitía tener el sistema operativo al completo. Ya entonces, muchas aplicaciones libres, entre las que fue especialmente interesante el caso de X Window, eran las mejores en su campo (utilidades Unix, compiladores...). Sin embargo, para completar el rompecabezas faltaba una pieza esencial: el núcleo del sistema operativo. El proyecto GNU estaba buscando esa pieza con un proyecto llamado Hurd, que pretendía construir un núcleo con tecnologías modernas.

2.3.2. La familia *BSD

Prácticamente en la misma época, la comunidad BSD estaba también en camino hacia un núcleo libre. En la distribución Net-2 sólo faltaban seis ficheros para tenerlo (el resto ya había sido construido por el CSRG o sus colaboradores). A principios de 1992, Bill Jolitz completó esos ficheros y distribuyó 386BSD, un sistema que funcionaba sobre arquitectura i386 y que con el tiempo daría lugar a los proyectos NetBSD, FreeBSD y OpenBSD. El desarrollo durante los meses siguientes fue rápido, y a finales de año ya era lo bastante estable como para ser usado en producción en entornos no críticos, que incluían, por ejemplo, un entorno de ventanas gracias al proyecto XFree (que había portado X Window a la arquitectura i386) o un compilador de gran calidad, GCC. Aunque había componentes que usaban otras licencias (como los procedentes del proyecto GNU, que usaban la GPL), la mayor parte del sistema se distribuía bajo la licencia BSD.

Bibliografía

Algunos de los episodios de esta época son ilustrativos de la potencia de los modelos de desarrollo de software libre. Es bien conocido el caso de Linus Torvalds, que desarrolló Linux mientras era estudiante de segundo curso de la Universidad de Helsinki. Pero no es el único caso de un estudiante que se abrió camino gracias a sus desarrollos libres. Por ejemplo, el alemán Thomas Roel portó X11R4 (una versión del sistema X Window) a un PC basado en un 386. Este desarrollo lo llevó a trabajar en Dell, y más adelante, a ser fundador de los proyectos X386 y XFree, fundamentales para que GNU/Linux y los *BSD tuvieran pronto un entorno de ventanas. Puede leerse más sobre la historia de XFree y el papel de Roel en ella en "The history of xFree86" (*Linux Magazine*, diciembre 1991) [135].

Luego vino la demanda de USL, que hizo que muchos usuarios potenciales temieran ser a su vez demandados si la Universidad de California perdía el juicio, o simplemente que el proyecto se parara. Quizás ésta fue la razón de que más adelante la base instalada de GNU/Linux fuera mucho mayor que la de todos los *BSD combinados. Pero eso es algo difícil de asegurar.

2.3.3. GNU/Linux entra en escena

En julio de 1991 Linus Torvalds (estudiante finés de veintiún años) pone el primer mensaje donde menciona su (por entonces) proyecto de hacer un sistema libre similar a Minix. En septiembre libera la primerísima versión (0.01), y cada pocas semanas aparecen nuevas versiones. En marzo de 1994 apareció

la versión 1.0, la primera denominada *estable*, pero el núcleo que había construido Linus era utilizable desde hacía bastantes meses. Durante este período, literalmente cientos de desarrolladores se vuelcan sobre Linux, integrando a su alrededor todo el software de GNU, XFree y muchos otros programas libres. A diferencia de los *BSD, el núcleo de Linux y gran parte de los componentes que se integran alrededor de él se distribuyen con la licencia GPL.

Bibliografía

La historia de Linux es probablemente una de las más interesantes (y conocidas) en el mundo del software libre. Se pueden encontrar muchos enlaces a información sobre ella en las páginas del décimo aniversario de su anuncio, aunque probablemente la más interesante es "History of Linux", de Ragib Hasan [138]. Como curiosidad, puede consultarse el hilo en el que Linus Torvalds anunciaba que estaba empezando a crear lo que luego fue Linux (en el grupo de noticias comp.os.minix) en <http://groups.google.com/groups?th=d161e94858c4c0b9>. Allí explica cómo lleva trabajando en su núcleo desde abril y cómo ya ha portado algunas herramientas del proyecto GNU sobre él (concretamente, menciona Bash y GCC).

Entre los muchos desarrollos aparecidos alrededor de Linux, uno de los más interesantes es el concepto de *distribución*¹. Las primeras distribuciones aparecieron pronto, en 1992 (MCC Interim Linux, de la Universidad de Manchester; TAMU, de Texas A&M, y la más conocida, SLS, que más tarde dio lugar a Slackware, que aún se distribuye en la década de 2000), y han supuesto la entrada de la competencia en el mundo del empaquetamiento de sistemas alrededor de Linux. Cada distribución trata de ofrecer un GNU/Linux listo para usar, y partiendo todas del mismo software, han de competir en mejoras que su base de usuarios considere importantes. Además de proporcionar paquetes precompilados y listos para usar, las distribuciones suelen ofrecer sus propias herramientas para gestionar la selección, la instalación, la sustitución y la desinstalación de estos paquetes, así como la instalación inicial en un ordenador, y la gestión y la administración del sistema operativo.

⁽¹⁾ Este concepto se explica en detalle en la entrada correspondiente de Wikipedia, www.wikipedia.org/wiki/Linux_distribution

Con el tiempo, unas distribuciones han ido sucediéndose a otras como las más populares. Entre todas ellas, cabe destacar las siguientes:

- 1) Debian, desarrollada por una comunidad de desarrolladores voluntarios.
- 2) Red Hat Linux, que primero desarrolló internamente la empresa Red Hat, pero que más adelante adoptó un modelo más comunitario, dando lugar a Fedora Core.
- 3) Suse, que dio lugar a OpenSUSE, en una evolución similar a la de Red Hat.
- 4) Mandriva (sucesora de Mandrake Linux y de Conectiva).
- 5) Ubuntu, derivada de Debian y producida a partir de ella por la empresa Canonical.

2.4. Tiempos de maduración

A mediados de la década de 2000, GNU/Linux, OpenOffice.org o Firefox tienen una presencia relativamente habitual en los medios de comunicación. La inmensa mayoría de empresas utiliza software libre al menos para algunos de sus procesos informáticos. Es difícil ser un estudiante de informática y no utilizar software libre en grandes cantidades. El software libre ha dejado de ser una nota a pie de página en la historia de la informática para convertirse en algo muy importante para el sector. Las empresas de informática, las del sector secundario (las que utilizan intensivamente software, aunque su actividad principal es otra) y las administraciones públicas empiezan a considerarlo como algo estratégico. Y está llegando, poco a poco pero con fuerza, a los usuarios domésticos. En líneas generales, se entra en una época de maduración.

Y en el fondo, empieza a surgir una pregunta muy importante, que de alguna forma resume lo que está ocurriendo: "¿estamos ante un nuevo modelo de industria software?". Aún podría ocurrir, quizás, que el software libre no llegara a ser más que una moda pasajera que con el tiempo sólo fuera recordada con nostalgia. Pero también podría ser (y ello parece cada vez más plausible) un nuevo modelo que está aquí para quedarse, y tal vez para cambiar radicalmente una de las industrias más jóvenes, pero también de las más influyentes.

2.4.1. Finales de la década de los noventa

A mediados de la década de los noventa, el software libre ofrecía ya entornos completos (distribuciones de GNU/Linux, sistemas *BSD...) que permitían el trabajo diario de mucha gente, sobre todo de desarrolladores de software. Aún había muchas asignaturas pendientes (la principal, disponer de mejores interfaces gráficas de usuario en una época en la que Windows 95 era considerado el estándar), pero ya había unos cuantos miles de personas en todo el mundo que sólo usaban software libre en su trabajo diario. Los anuncios de nuevos proyectos se sucedían y el software libre comenzaba su largo camino hacia las empresas, los medios de comunicación y, en general, el conocimiento público.

De esta época es también el despegue de Internet como red para todos, en muchos casos de la mano de programas libres (sobre todo en su infraestructura). La llegada del web a los hogares de millones de usuarios finales consolida esta situación, al menos en lo que se refiere a servidores: los servidores web (HTTP) más populares siempre han sido libres (primero el servidor del NCSA, luego Apache).

Quizás el comienzo del camino del software libre hasta su puesta de largo en la sociedad pueda situarse en el célebre ensayo de Eric Raymond, "La catedral y el bazar" (Eric S. Raymond, 2001) [192]. Aunque mucho de lo expuesto en él era ya bien conocido por la comunidad de desarrolladores de software libre, reunirlo en un artículo y darle una gran difusión lo convirtió en una influyente

herramienta de promoción del concepto de *software libre* como mecanismo de desarrollo alternativo al usado por la industria del software tradicional. Otro artículo muy importante en esta época fue "Setting up shop. The Business of open-source software" [141], de Frank Hecker, que por primera vez expuso los modelos de negocio posibles en torno al software libre y que fue escrito para influir en la decisión sobre la liberación del código del navegador de Netscape.

Si el artículo de Raymond supuso una gran herramienta de difusión de algunas de las características fundamentales del software libre, la liberación del código del navegador de Netscape fue el primer caso en que una empresa relativamente grande, de un sector muy innovador (la entonces naciente industria del web), tomaba la decisión de liberar como software libre uno de sus productos. En aquella época, Netscape Navigator estaba perdiendo la batalla de los navegadores web frente al producto de Microsoft (Internet Explorer), en parte por las tácticas de Microsoft de combinarlo con su sistema operativo. Para muchos, Netscape hizo lo único que podía hacer: tratar de cambiar las reglas para poder competir con un gigante. Y de este cambio de reglas (tratar de competir con un modelo de software libre) nació el proyecto Mozilla. Este proyecto, no sin problemas, ha llevado varios años después a un navegador que, si bien no ha recuperado la enorme cuota de mercado que tuvo en su día Netscape Navigator, parece que técnicamente es al menos tan bueno como sus competidores privativos.

En cualquier caso, y con independencia de su éxito posterior, el anuncio de Netscape de liberar el código de su navegador supuso un fuerte impacto en la industria del software. Muchas empresas comenzaron a considerar el software libre como digno de consideración.

También los mercados financieros se empezaron a ocupar del software libre. En plena euforia de las puntocom, varias empresas de software libre se convierten en objetivo de inversores. Quizás el caso más conocido es el de Red Hat, una de las primeras empresas que reconocieron que la venta de CD con sistemas GNU/Linux listos para usar podía ser un modelo de negocio. Red Hat comenzó distribuyendo su Red Hat Linux, haciendo gran énfasis (al menos para lo habitual en la época) en la facilidad de manejo y mantenimiento del sistema por personas sin conocimientos específicos de informática. Con el tiempo, fue diversificando su negocio, manteniéndose en general en la órbita del software libre, y en septiembre de 1998 anunció que Intel y Netscape habían invertido en ella. "Si es bueno para Intel y Netscape, seguro que es bueno para nosotros", debieron de pensar muchos inversores. Cuando Red Hat salió a bolsa en el verano de 1999, la oferta pública de acciones fue suscrita completamente, y pronto el valor de cada título subió espectacularmente. Fue la primera vez que una empresa conseguía financiación del mercado de valores con un modelo basado en el software libre. Pero no fue la única: lo mismo hicieron más tarde otras como VA Linux o Andover.net (que más tarde fue adquirida por VA Linux).

Nota

Red Hat proporciona una lista de hitos históricos relacionados con su empresa en <http://fedora.redhat.com/about/history/>.

En esta época nacieron también muchas empresas basadas en modelos de negocio en torno al software libre. Sin salir a bolsa y sin lograr tan estupendas capitalizaciones, fueron sin embargo muy importantes para el desarrollo del software libre. Por ejemplo, aparecieron muchas empresas que empezaron distribuyendo sus propias versiones de GNU/Linux, como SuSE (Alemania), Conectiva (Brasil) o Mandrake (Francia), que más tarde se uniría a la anterior para formar Mandriva. Otras proporcionaban servicios a empresas que ya demandaban mantenimiento y adaptación de productos libres: LinuxCare (EE.UU.), Alcove (Francia), ID Pro (Alemania); y muchas más.

Por su parte, los gigantes del sector también empezaron a posicionarse ante el software libre. Algunas empresas, como IBM, lo incorporaron directamente en su estrategia. Otras, como Sun Microsystems, mantuvieron con él una curiosa relación, a veces de apoyo, a veces de indiferencia, a veces de enfrentamiento. La mayoría (como Apple, Oracle, HP, SGI, etc.) exploraron el modelo del software libre con diversas estrategias, que iban desde la liberación selectiva de software hasta el simple porte a GNU/Linux de sus productos. Entre estos dos extremos, hubo otras muchas líneas de acción, como la utilización más o menos intensa de software libre en sus productos (como fue el caso del Mac OS X) o la exploración de modelos de negocio basados en el mantenimiento de productos libres.

Desde el punto de vista técnico, lo más destacable de esta época fue probablemente la aparición de dos ambiciosos proyectos al objeto de conseguir llevar el software libre al entorno de escritorio (*desktop*) de los usuarios no muy versados en la informática: KDE y GNOME. Dicho de forma muy simplista, el objetivo final era que no hubiera que usar la línea de órdenes para interactuar con GNU/Linux o *BSD ni con los programas sobre esos entornos.

KDE fue anunciado en octubre de 1996. Utilizando las bibliotecas gráficas Qt (por aquel entonces un producto privativo de la empresa Trolltech, pero gratuito para su uso sobre GNU/Linux²), iniciaron la construcción de un conjunto de aplicaciones de escritorio que funcionasen de forma integrada y que tuviesen un aspecto uniforme. En julio de 1998 liberaron la versión 1.0 del K Desktop Environment, que fue pronto seguida de nuevas versiones cada vez más completas y maduras. Las distribuciones de GNU/Linux pronto incorporaron KDE como escritorio para sus usuarios (o al menos como uno de los entornos de escritorio que sus usuarios podían elegir).

⁽²⁾Más tarde, Qt pasó a ser distribuido bajo una licencia libre, la QPL (Qt Public License), no compatible con la GPL, lo que causaba algún problema, puesto que la mayor parte de KDE estaba distribuida bajo la GPL. Con el tiempo, finalmente Trolltech decidió distribuir Qt bajo la licencia GPL, con lo que estos problemas terminaron.

En gran medida como reacción a la dependencia que tenía KDE de la biblioteca propietaria Qt, en agosto de 1997 se anunció el proyecto GNOME (Miguel de Icaza, "The story of the GNOME Project") [101], con fines y características muy similares a los de KDE, pero con el objetivo explícito de que todos sus

componentes fueran libres. En marzo de 1999 se liberó GNOME 1.0, que con el tiempo también se iría mejorando y estabilizando. A partir de ese momento, la mayor parte de las distribuciones de sistemas operativos libres (y muchos derivados de Unix privativos) ofrecieron como opción el escritorio de GNOME o el de KDE y las aplicaciones de ambos entornos.

Simultáneamente, los principales proyectos de software libre que ya estaban en marcha continúan con buena salud y surgen nuevos proyectos cada día. En varios nichos de mercado, se observa cómo la mejor solución (reconocida por casi todo el mundo) es software libre. Por ejemplo, Apache ha mantenido casi desde su aparición en abril de 1995 la mayor cuota de mercado entre los servidores web; XFree86, el proyecto libre que desarrolla X Window, es con diferencia la versión de X Window más popular (y por tanto, el sistema de ventanas para sistemas de tipo Unix más extendido); GCC es reconocido como el compilador de C más portable y uno de los de mejor calidad; GNAT, sistema de compilación para Ada 95, se hace con la mayor parte del mercado de compiladores Ada en pocos años; y así sucesivamente.

En 1998 se creó la Open Source Initiative (OSI), que decidió adoptar el término *open source software* ('software de fuente abierta') como una marca para introducir el software libre en el mundo comercial, tratando de evitar la ambigüedad que en inglés supone el término *free* (que significa tanto 'libre' como 'gratis'). Esta decisión supuso (y aún supone) uno de los debates más enconados del mundo del software libre, ya que la Free Software Foundation y otros consideraron que era mucho más apropiado hablar de *software libre* (Richard Stallman, "Why *free software* is better than *open source*", 1998) [206]. En cualquier caso, la OSI realizó una fructífera campaña de difusión de su nueva marca, que ha sido adoptada por muchos como la forma preferida de hablar del software libre, sobre todo en el mundo anglosajón. Para definir el software *open source*, la OSI utilizó una definición derivada de la que utiliza el proyecto Debian para definir el software libre ("Debian free software guidelines", http://www.debian.org/social_contract.html#guidelines) [104], que por otra parte refleja con bastante aproximación la idea de la FSF al respecto ("Free software definition", <http://www.gnu.org/philosophy/free-sw.html>) [120], por lo que desde el punto de vista práctico casi cualquier programa que es considerado software libre es también considerado *open source* y viceversa. Sin embargo, las comunidades de software libre y de software de fuente abierta (o al menos las personas que se identifican como parte de una o de otra) pueden ser profundamente diferentes.

2.4.2. Década de 2000

A principios de la década de 2000 el software libre es ya un serio competidor en el segmento de servidores y comienza a estar listo para el escritorio. Sistemas como GNOME, KDE, OpenOffice.org y Mozilla Firefox pueden ser utilizados por usuarios domésticos y son suficientes para las necesidades de muchas em-

presas, al menos en lo que a ofimática se refiere. Los sistemas libres (y sobre todo los basados en Linux) son fáciles de instalar, y la complejidad para mantenerlos y actualizarlos es comparable a la de otros sistemas privativos.

En estos momentos, cualquier empresa de la industria del software tiene una estrategia con respecto al software libre. La mayoría de las grandes multinacionales (IBM, HP, Sun, Novell, Apple, Oracle...) incorpora el software libre con mayor o menor decisión. En un extremo podríamos situar empresas como Oracle, que reaccionan simplemente portando sus productos a GNU/Linux. En el otro podría situarse IBM, que tiene la estrategia más decidida y ha realizado las mayores campañas de publicidad sobre GNU/Linux. Entre los líderes del mercado informático, sólo Microsoft se ha significado con una estrategia claramente contraria al software libre y en particular al software distribuido bajo licencia GPL.

En cuanto al mundo del software libre en sí mismo, a pesar de los debates que de vez en cuando sacuden a la comunidad, el crecimiento es enorme. De día en día hay más desarrolladores, más proyectos de software libre activos, más usuarios... Cada vez más el software libre está pasando de ser algo marginal a convertirse en un competidor que hay que tener en cuenta.

Ante este desarrollo, aparecen nuevas disciplinas que estudian específicamente el software libre, como la ingeniería del software libre. A partir de sus investigaciones comenzamos, poco a poco, a entender cómo funciona el software libre en sus diferentes aspectos: modelos de desarrollo, modelos de negocio, mecanismos de coordinación, gestión de proyectos libres, motivación de desarrolladores, etc.

En estos años empiezan también a verse los primeros efectos de la *deslocalización* que permite el desarrollo de software libre: países considerados "periféricos" participan en el mundo del software libre de forma muy activa. Por ejemplo, es significativo el número de desarrolladores mexicanos o españoles (ambos países con poca tradición de industria software) en proyectos como GNOME (Lancashire, "Code, culture and cash: the fading altruism of open source development", 2001) [164]. Y es aún más interesante el papel de Brasil, que está teniendo una gran cantidad de desarrolladores y expertos en tecnologías de software libre y un decidido apoyo por parte de las administraciones públicas. Mención aparte merece el caso de gnuLinEx, muy significativo de cómo una región con poca tradición de desarrollo de software puede tratar de cambiar la situación con una estrategia agresiva de implantación de software libre.

Desde el punto de vista de la toma de decisiones a la hora de implantar soluciones software, hay que destacar el hecho de que hay ciertos mercados (como los servicios de Internet o la ofimática) en los que el software libre se ha convertido en una opción natural cuya no consideración es difícil de justificar cuando se está estudiando qué tipo de sistema utilizar.

En el lado negativo, estos años han visto cómo el entorno legal en el que se mueve el software libre está cambiando rápidamente en todo el mundo. Por una parte, las patentes de software (patentes de programación) son consideradas cada vez en más países. Por otro, las nuevas leyes de protección de derechos de autor dificultan o hacen imposible el desarrollo de aplicaciones libres en algunos ámbitos, el más conocido de los cuales es el de los visores de DVD (debido al algoritmo CSS de ofuscación de imágenes que se utiliza en esa tecnología).

gnuLinEx

A principios de 2002 la Junta de Extremadura dio a conocer públicamente el proyecto gnuLinEx. La idea era simple: promover la creación de una distribución basada en GNU/Linux con el objetivo fundamental de utilizarla en los miles de ordenadores que se van a instalar en los centros educativos públicos de toda la región. Extremadura, situada en la parte occidental de España, frontera con Portugal, cuenta con aproximadamente un millón de habitantes y nunca se había destacado por sus iniciativas tecnológicas. De hecho, la región prácticamente carecía de industria de software.

En este contexto, gnuLinEx ha supuesto una aportación muy interesante en el panorama del software libre a escala mundial. Mucho más allá de ser una nueva distribución de GNU/Linux basada en Debian (lo que no deja de ser algo relativamente anecdótico), y más allá de su enorme impacto en los medios de comunicación (es la primera vez que Extremadura ha sido portada de *The Washington Post* y una de las primeras que lo ha sido un producto de software libre), lo extraordinario es la (al menos aparentemente) sólida apuesta de una administración pública por el software libre. La Junta de Extremadura decidió probar un modelo diferente en cuanto al software usado para la enseñanza y más adelante a todo el uso de la informática dentro de sus competencias. Esto la ha convertido en la primera administración pública de un país desarrollado que toma decididamente este camino. En torno a la iniciativa de la Junta se ha producido también mucho movimiento, tanto dentro como fuera de Extremadura: hay academias que enseñan informática con gnuLinEx; se han escrito libros para apoyar esta enseñanza; se venden ordenadores con gnuLinEx preinstalado. En general, se intenta crear alrededor de esta experiencia todo un tejido docente y empresarial que le proporcione soporte. Y la experiencia se ha exportado. A principios del siglo XXI varias comunidades autónomas en España han apostado (de una u otra forma) por el software libre para la enseñanza, y en general, su importancia para las administraciones públicas es ampliamente reconocida.

Knoppix

Desde finales de los años noventa hay distribuciones de GNU/Linux que se instalan fácilmente, pero probablemente Knoppix, cuya primera versión apareció en 2002, ha llevado este concepto a su máxima expresión. Se trata de un

CD que arranca prácticamente en cualquier PC, convirtiéndolo (sin tener siquiera que formatear el disco, ya que permite su uso "en vivo") en una máquina GNU/Linux completamente funcional, con una selección de las herramientas más habituales. Knoppix une una buena detección automática de hardware con una buena selección de programas y un funcionamiento "en vivo". Permite, por ejemplo, una experiencia rápida y directa de qué puede suponer trabajar con GNU/Linux. Y está dando lugar a toda una familia de distribuciones del mismo tipo, especializadas para necesidades específicas de un perfil de usuarios.

OpenOffice.org

En 1999 Sun Microsystems compró una empresa alemana llamada Stardivision, cuyo producto estrella era StarOffice, un juego de herramientas ofimático similar en funcionalidad a Office, el juego de herramientas de Microsoft. Un año más tarde, Sun distribuyó gran parte del código de StarOffice bajo una licencia libre (la GPL), dando lugar al proyecto OpenOffice.org. Este proyecto liberó la versión 1.0 de OpenOffice.org en mayo de 2002. OpenOffice.org se ha convertido en un juego de aplicaciones ofimáticas de calidad y funcionalidad similar a la de cualquier otro producto ofimático y, lo que es más importante, interopera muy bien con los formatos de datos de Microsoft Office. Estas características han hecho de ella la aplicación de referencia del software libre en el mundo de la ofimática.

La importancia de OpenOffice.org, desde el punto de vista de extensión del software libre a un gran número de usuarios, es enorme. Por fin ya es posible cambiar, prácticamente sin traumas, de los entornos privativos habituales en ofimática (sin duda la aplicación estrella en el mundo empresarial) a entornos completamente libres (por ejemplo, GNU/Linux más GNOME y/o KDE más OpenOffice.org). Además, la transición puede hacerse de forma muy suave: como OpenOffice.org funciona también sobre Microsoft Windows, no es preciso cambiar de sistema operativo para experimentar en profundidad el uso de software libre.

Mozilla, Firefox y los demás

Prácticamente desde su aparición en 1994 hasta 1996, Netscape Navigator fue el líder indiscutible del mercado de navegadores web, con cuotas de hasta el 80%. La situación empezó a cambiar cuando Microsoft incluyó Internet Explorer en su Windows 95, lo que supuso que poco a poco Netscape Navigator fuera perdiendo cuota de mercado. A principios de 1998 Netscape anunció que iba a distribuir gran parte del código de su navegador como software libre, cosa que efectivamente hizo en marzo del mismo año, lanzando el proyecto Mozilla. Durante bastante tiempo dicho proyecto estuvo rodeado de incerti-

dumbre, e incluso de pesimismo (por ejemplo cuando su líder, Jamie Zawinski, lo abandonó), dado que pasaba el tiempo y no aparecía ningún producto como resultado de su lanzamiento.

En enero de 2000, el proyecto liberó Mozilla M13, que fue considerada la primera versión razonablemente estable. Pero sólo en mayo de 2002 se publicó finalmente la versión 1.0, la primera oficialmente estable, más de cuatro años después de la liberación del primer código de Netscape Navigator.

Por fin Mozilla era una realidad, aunque quizás demasiado tarde, si tenemos en cuenta las cuotas de mercado que tuvo Internet Explorer durante 2002 ó 2003 (en los que fue líder indiscutible relegando a Mozilla y a otros a una posición completamente marginal). Pero el proyecto Mozilla, a pesar de tardar tanto, dio sus frutos; no sólo los esperados (el navegador Mozilla), sino muchos otros que podrían considerarse "colaterales", como por ejemplo Firefox, otro navegador basado en el mismo motor de HTML, que con el tiempo se ha convertido en el producto principal, y que desde su aparición en 2005 está consiguiendo erosionar poco a poco las cuotas de mercado de otros navegadores.

El proyecto Mozilla ha ayudado a completar un gran hueco en el mundo del software libre. Antes de la aparición de Konqueror (el navegador del proyecto KDE), no había muchos navegadores libres con interfaz gráfica. A partir de la publicación de Mozilla, han ido apareciendo gran cantidad de proyectos basados en él que han producido una buena cantidad de navegadores. Por otro lado, la combinación de Mozilla Firefox y OpenOffice.org permite usar software libre para las tareas más cotidianas, incluso en un entorno Microsoft Windows (ambos funcionan no sólo sobre GNU/Linux, *BSD y otros sistemas de tipo Unix, sino que también lo hacen sobre Windows). Esto permite, por primera vez en la historia del software libre, que la transición de software privativo a software libre en entornos de oficina sea simple: se puede empezar utilizando estas dos aplicaciones sobre Windows, sin cambiar de sistema operativo (en el caso de los que lo usan habitualmente), y con el tiempo eliminar la única pieza no libre y pasar a GNU/Linux o a FreeBSD.

El caso SCO

A principios de 2003 la corporación SCO (anteriormente Caldera Systems y Caldera International) interpuso una demanda contra IBM por supuesta infracción de sus derechos de propiedad intelectual. Aunque la demanda era compleja, estaba centrada en la acusación de que IBM había contribuido al núcleo de Linux con código que era de SCO. En mayo de 2007 el asunto todavía no estaba resuelto y se había complicado aún más con más demandas (de IBM y Red Hat contra SCO, de SCO contra AutoZone y DaimlerChrysler, dos grandes usuarios informáticos) y con campañas de SCO en las que amenazaba con demandar a grandes empresas que usan Linux, etc.

Bibliografía

En "Netscape Navigator", de Brian Wilson, [234], puede consultarse una reseña detallada de las principales versiones de Netscape Navigator y Mozilla, así como de sus principales características.

Aunque el ganador de esta gran batalla legal aún no se conoce, el asunto ha puesto de relieve ciertos aspectos legales del software libre. En particular, muchas empresas se han planteado los problemas a los que se pueden enfrentar al usar Linux y otros programas libres y qué garantías tienen de que al hacerlo no están infringiendo derechos de propiedad intelectual o industrial de terceros.

De alguna manera, este caso y algunos otros (como los relacionados con la validez de la GPL que se han resuelto en Alemania en 2005) pueden interpretarse también como un síntoma de la madurez del software libre. Ya ha dejado de ser un elemento extraño al mundo empresarial y ha entrado a formar parte de muchas de sus actividades (incluidas las que tienen que ver con estrategias legales).

Ubuntu, Canonical, Fedora y Red Hat

Aunque Canonical (la empresa que produce y comercializa Ubuntu) podría considerarse casi como una recién llegada al negocio de las distribuciones GNU/Linux, sus actividades merecen que le dediquemos atención. En relativamente poco tiempo, Ubuntu se ha establecido como una de las distribuciones más conocidas y utilizadas, con fama de buena calidad y mucha simplicidad de instalación y uso. Ubuntu también se caracteriza por tener mucho más cuidado en incluir fundamentalmente software libre que la mayoría de las demás distribuciones producidas por empresas.

Sin embargo, la característica probablemente fundamental de Ubuntu (y de la estrategia de Canonical) ha sido basarse en Debian, una distribución creada y mantenida por voluntarios. De hecho, Ubuntu no ha sido el primer caso de distribución basada en Debian (otro caso bien conocido es gnuLinEx), pero quizás sí que ha sido, entre todos ellos, el que más recursos ha recibido. Por ejemplo, Canonical ha contratado a una gran cantidad de expertos en Debian (muchos de los cuales participan en el proyecto) y ha seguido una estrategia que parece buscar la colaboración con el proyecto voluntario. De alguna manera, Canonical ha tratado de completar lo que considera que falta en Debian para tener una aceptación por el usuario medio.

Red Hat, por su parte, ha seguido un camino diferente para llegar a una situación bastante similar. Partiendo de una distribución realizada completamente con sus propios recursos, decidió colaborar con Fedora, un grupo de voluntarios que ya estaba trabajando con distribuciones basadas en Red Hat, para producir Fedora Core, su distribución "popular". Red Hat mantiene su versión para empresas, pero esta colaboración con voluntarios es, en el fondo, similar a la que ha dado lugar a Ubuntu.

Quizás todos estos movimientos no son más que el fruto de la feroz competencia que tiene lugar en el mercado de distribuciones GNU/Linux y de una tendencia de más calado: la colaboración de empresas con voluntarios (con *la comunidad*) para producir software libre.

Las distribuciones particularizadas

Desde que Linux entró en escena, muchos grupos y empresas han creado su propia distribución basada en él. Pero durante estos años, el fenómeno se ha extendido a muchas organizaciones y empresas que quieren tener una distribución particularizada para sus propias necesidades. El abaratamiento del proceso de particularización de una distribución y la amplia disposición del conocimiento técnico para hacerlo han permitido la expansión de esta actividad, que se ha convertido incluso en un nicho de negocio para algunas empresas.

Quizás uno de los casos más conocidos de distribuciones particularizadas es el de las distribuciones autonómicas en España. La Junta de Extremadura comenzó con su gnuLinEx una tendencia que han seguido muchas otras comunidades autónomas. El proceso es tan común que varias de ellas convocan de forma regular concursos públicos para la creación y el mantenimiento de las nuevas versiones de sus distribuciones.

La creación de distribuciones particularizadas hace real una tendencia de la que se hablaba desde hacía tiempo en el mundo del software libre: la adaptación de los programas a las necesidades específicas de los usuarios, sin que haga falta que los productores originales tengan necesariamente que realizar esta adaptación.

Bibliografía

Algunas de las distribuciones autonómicas más conocidas de GNU/Linux:

- gnuLinEx: <http://linex.org> (Extremadura)
- Guadalinux: <http://guadalinux.org> (Andalucía)
- Lliurex: <http://lliurex.net> (Comunidad Valenciana)
- Augustux: <http://www.zaralinux.org/proy/augustux/> (Aragón)
- MAX: http://www.educa.madrid.org/web/madrid_linux/ (Madrid)
- MoLinux: <http://molinux.info> (Castilla-La Mancha)

Colaboración de empresas con empresas y de voluntarios con empresas

Desde prácticamente el principio del software libre ha habido empresas que colaboraban con voluntarios en el desarrollo de aplicaciones. Sin embargo, durante estos años en los que parece que se está llegando a la madurez, cada vez son más las empresas que usan el software libre como parte de su estrategia para colaborar con otras empresas, cuando eso les resulta interesante. Dos

de los casos más significativos, organizados específicamente con este fin, son ObjectWeb (alianza nacida en Francia que con el tiempo pasó a ser claramente internacional) y Morfeo (en España). En ambos casos, un grupo de empresas se ha puesto de acuerdo para desarrollar un conjunto de sistemas libres que les resultan interesantes y que deciden distribuir como software libre.

En otros casos, las empresas buscan activamente bien colaborar en proyectos libres promovidos por voluntarios, o bien tratar de que sean los voluntarios los que vayan a colaborar a sus propios proyectos libres. Ejemplos de la primera situación son la GNOME Foundation o el ya mencionado Ubuntu con respecto a Debian. Entre los segundos, se puede destacar el caso de Sun y OpenOffice.org y OpenSolaris, o el de Red Hat con Fedora Core.

Extensión a otros ámbitos

El software libre ha demostrado que, en el campo de la producción de programas, es posible otra forma de hacer las cosas. Se ha visto en la práctica cómo otorgando las libertades de distribución, modificación y uso se puede conseguir la sostenibilidad, bien mediante trabajo voluntario, bien incluso mediante una generación de negocio que permita la supervivencia de las empresas.

Con el tiempo, esta misma idea se está trasladando a otros campos de producción de obra intelectual. Las licencias Creative Commons han permitido simplificar el proceso de liberación en campos como la literatura, la música o el vídeo. Wikipedia está mostrando que en un área tan particular como la producción de enciclopedias se puede recorrer un camino muy interesante. Y cada vez hay más autores literarios, grupos musicales e incluso productoras de películas interesados en modelos libres de producción y distribución.

En todos estos dominios queda mucho camino por recorrer, y en casi todos ellos la práctica aún no ha demostrado completamente que es posible la creación sostenible con modelos libres. Pero no se puede negar que la experimentación al respecto está entrando en estado de ebullición.

El software libre como objeto de estudio

Aunque algunos trabajos, como el conocido "La catedral y el bazar" empezaron a abrir el camino del estudio del software libre como tal, no fue hasta el año 2001 y siguientes cuando la comunidad académica comenzó a considerar el software libre como un objeto digno de estudio. Con el tiempo, la gran disponibilidad de datos (casi todo en el mundo del software libre es público y está disponible en almacenes de información públicos) y las novedades que se observan en él han ido centrando la atención de muchos grupos. A mediados de la década de 2000 hay ya varios congresos internacionales que se dedican

específicamente al software libre, las revistas más prestigiosas le dedican con cierta regularidad monográficos, y las agencias que financian la investigación están abriendo líneas orientadas específicamente a él.

2.5. El futuro: ¿una carrera de obstáculos?

Sin duda, es difícil predecir el futuro. Y desde luego, no es algo a lo que nos vayamos a dedicar aquí. Por lo tanto, más que tratar de explicar cómo será el futuro del software libre, intentaremos mostrar los problemas que previsiblemente tendrá que afrontar (y de hecho lleva ya tiempo afrontando). De cómo sea el mundo del software libre capaz de superar estos obstáculos dependerá, indudablemente, su situación dentro de unos años.

- Técnica FUD (*fear, uncertainty, doubt*, o en español, 'miedo, desconocimiento, duda'). Se trata de una técnica bastante habitual en el mundo de las tecnologías de la información y que ha sido utilizada por los competidores de productos de software libre para tratar de desacreditarlos, con mayor o menor razón y con éxito variable. En líneas generales, el software libre, quizás debido a su complejidad y a diversos métodos de penetración en las empresas, ha resultado bastante inmune a estas técnicas.
- Disolución. Muchas empresas están probando los límites del software libre como modelo, y en particular tratan de ofrecer a sus clientes modelos que presentan algunas características similares al software libre. El principal problema que puede presentar este tipo de modelos es la confusión que genera en los clientes y los desarrolladores, que tienen que estudiar con mucho detalle la letra pequeña para darse cuenta de que lo que se les está ofreciendo no tiene las ventajas que para ellos supone el software libre. El caso más conocido de modelos de este tipo es el programa Shared Source, de Microsoft.
- Desconocimiento. En muchos casos los usuarios llegan al software libre simplemente porque creen que es gratis; o porque consideran que está "de moda". Si no profundizan más y no estudian con cierto detenimiento las ventajas que les puede ofrecer el software libre como modelo, corren el riesgo de no aprovecharse de ellas. En muchos casos, las suposiciones de partida en el mundo del software libre son tan diferentes de las habituales en el mundo del software privativo que es indispensable un mínimo análisis para comprender que lo que en un caso es habitual en el otro puede ser imposible, y viceversa. El desconocimiento, por lo tanto, no puede sino generar insatisfacciones y pérdida de oportunidades en cualquier persona u organización que se aproxime al software libre.
- Impedimentos legales. Sin duda éste es el principal problema con el que se va a encontrar el software libre en los próximos años. Aunque el entorno legal en el que se desarrolló el software libre durante la década de 1980 y la primera mitad de la de 1990 no era ideal, al menos dejaba suficien-

te espacio como para que creciese en libertad. Desde entonces, la extensión del ámbito de la patentabilidad al software (que se ha producido en muchos países desarrollados) y las nuevas legislaciones sobre derechos de autor (que limitan la libertad de creación del desarrollador de software) suponen cada vez barreras más altas a la entrada del software libre en segmentos importantes de aplicaciones.

2.6. Resumen

En este capítulo se presenta la historia del software libre. Los años sesenta fueron una etapa dominada por los grandes ordenadores e IBM en la que el software se distribuía junto al hardware, y habitualmente con el código fuente. En la década de los setenta se comenzó a vender el software por separado, y rápidamente la distribución privativa, que no incluye el código fuente y que no otorga permiso de modificación o redistribución, se convirtió casi en la única opción.

En la década de 1970 comenzó el desarrollo del sistema operativo Unix en los Bell Labs de AT&T, que dio lugar más adelante a Unix BSD. Su evolución, paralela al nacimiento de Internet, sirvió de campo de pruebas para nuevas formas de desarrollo en colaboración que fueron luego habituales en el mundo del software libre.

En 1984 Richard Stallman comenzó a trabajar en el proyecto GNU, fundó la Free Software Foundation (FSF), escribió la licencia GPL y, en general, sentó los fundamentos del software libre tal como ha sido conocido más tarde.

En la década de 1990 Internet fue madurando y proporcionando a las comunidades de software libre nuevos canales de comunicación y distribución. En 1991, Linus Torvalds comenzó a desarrollar un núcleo libre (Linux) que permitió completar el sistema GNU, que contaba ya con casi todas las piezas para convertirse en un sistema completo similar a Unix: compilador de C (GCC), editor (Emacs), sistema de ventanas (X Window), etc. Nacieron así los sistemas operativos GNU/Linux, que fructificaron en multitud de distribuciones, como Red Hat Linux y Debian GNU/Linux. A finales de la década de 1990 estos sistemas se completaban con dos entornos de escritorio: KDE y GNOME.

En la década de 2000 el software libre llega a liderar algunos sectores (como el de servidores web, dominado por Apache), y aparecen nuevas herramientas que cubren gran cantidad de necesidades informáticas.

Ved también

El lector interesado podrá encontrar en el apéndice B una lista de algunas de las fechas más relevantes de la historia del software libre.