

CONFERENCIA DE LAS NACIONES UNIDAS SOBRE COMERCIO Y DESARROLLO
Ginebra

**INFORME SOBRE COMERCIO ELECTRÓNICO
Y DESARROLLO, 2003**

Capítulo 4

**SOFTWARE LIBRE Y DE CÓDIGO FUENTE ABIERTO:
CONSECUENCIAS PARA LA POLÍTICA EN
MATERIA DE TIC Y EL DESARROLLO**



NACIONES UNIDAS
Nueva York et Ginebra, 2003

Capítulo 4

SOFTWARE LIBRE Y DE CÓDIGO FUENTE ABIERTO: CONSECUENCIAS PARA LA POLÍTICA EN MATERIA DE TIC Y EL DESARROLLO

Los programas libres (software libre) y de código fuente abierto (FOSS) ponen en tela de juicio nuestras ideas preconcebidas sobre cómo se producen y se distribuyen los programas informáticos. La industria del software genera actualmente ingresos anuales por encima de los 300.000 millones de dólares de los EE.UU. Los FOSS son programas que han hecho público su código fuente y que permiten a los usuarios cambiar el código fuente y redistribuir los programas que de él se derivan, y quizá incluso los motivan a hacerlo. La liberación del código fuente facilita una amplia colaboración en la producción de programas, una mejor adaptación de los programas a otros sistemas operativos (*porting*)¹ con programas producidos por programadores independientes, y la adaptación de los programas para satisfacer diferentes necesidades comerciales, reglamentarias, culturales y lingüísticas. Lo que es más importante, en particular para los países en desarrollo, los FOSS permiten a los expertos y líderes en tecnología de la información (TI) de hoy y de mañana adquirir cualificaciones y acrecentar rápidamente sus conocimientos.

Su opuesto tecnológico, los programas de fuente cerrada o protegidos por derechos de propiedad, pueden no respaldar el proceso de desarrollo de las TIC, de la misma manera, porque requieren una inversión inicial considerable en derechos de licencia para la instalación y las mejoras, no siempre son adaptables a las circunstancias locales, y su uso exclusivo o incluso dominante puede no promover adecuadamente el desarrollo local de los conocimientos especializados y las cualificaciones que hacen falta para abrazar toda la economía de la información. Si bien los programas patentados tienen su lugar y función, los gobiernos han de examinar su posición normativa sobre los FOSS en el contexto de su programa general y de sus ambiciones de superar

la brecha digital y utilizar las TIC para un mayor y mejor comercio y desarrollo.

A. Introducción

El equipo (hardware) que hace posibles la informática y las comunicaciones modernas ha evolucionado a un ritmo extraordinario en los últimos decenios y es probable que ese proceso continúe. La "ley de Moore", que es en realidad la observación de una pauta, afirma que la capacidad de los microprocesadores se duplica, mientras que su precio disminuye en casi la mitad, cada 18 meses². Esto ha creado una ecología del procesamiento de datos en que el hardware es mucho más perfeccionado y fidedigno que los programas - las instrucciones que los seres humanos crean para él.

Pero no hay ley de Moore para el software. Mientras que el precio de la capacidad de procesamiento baja rápidamente, los programas que pueden hacer uso de esa capacidad se hacen cada vez más complicados, a veces más caros y menos fiables, y casi siempre más difíciles de configurar y de mantener. Sin embargo, son los programas los que dictan las reglas fundamentales del procesamiento de la información, y por tanto de una economía de la información y una sociedad de la información. Una enorme capacidad de procesamiento conectada con un ancho de banda cada vez mayor es la infraestructura básica. Los programas determinan cómo se manipula la información, por dónde circula, hacia quién y por qué razones.

Los países en desarrollo tienen que definir sus estrategias en materia de TIC y relacionarlas con el proceso de desarrollo. La política relativa a la utilización de software se ha convertido de pronto en una cuestión importante porque una nueva opción ha pasado a ser viable: la de los FOSS, y,

con ella, la promesa del desarrollo posibilitado por la información. Sin embargo, Einstein comentó en una ocasión que "a veces se paga el precio más alto por las cosas que se consiguen gratis", y esta idea se aplica al debate sobre los FOSS. Los países y las personas que se beneficien de los FOSS serán los que se esfuercen por formular sus políticas de manera bien informada y contribuyan a su vez a la base de conocimientos de los FOSS.

En el presente capítulo se explica cómo se crean -no en un sentido técnico, sino en el sentido organizativo- los FOSS y, por comparación, los programas amparados por derechos de propiedad, y por qué eso importa, a las economías desarrolladas y, en particular, a las en desarrollo. Se sugiere que el proceso de los FOSS produce mejores programas, que podrían igualar las mejoras sin fin del equipo físico. Como cualquier producto, los programas son simplemente el resultado de un proceso de producción que combina de una manera particular esfuerzo humano, insumos y algún tipo de capital.

Cuadro 4.1

Las diez principales empresas de software, ordenadas por ingresos y capitalización bursátil

	Ingresos anuales	Capitalización bursátil
	(En millones de dólares de los EE. UU.)	
1. Microsoft	31.375	260.000
2. Oracle	9.487	63.400
3. SAP	7.700	32.300
4. Computer Associates	3.083	12.400
5. VERITAS	1.531	10.100
6. Electronic Arts	2.489	9.300
7. Intuit	1.373	9.000
8. Adobe Systems	1.194	8.000
9. Symantec	1.328	6.600
10. PeopleSoft	1.949	4.700
11. Competencia con las diez ³ principal	8.445	28.582
Total	69.954	444.400

Fuente: Estimaciones de la UNCTAD basadas en datos de Yahoo Finance (<http://finance.yahoo.com>) y Financial Times Market Data and Tools (<http://www.ft.com>).

La manera "normal" de organizar la producción de software ha sido muy semejante a la manera normal de crear un bien industrial

complejo: una división formal del trabajo que utiliza conocimientos de dominio privado, protegidos por derechos de propiedad intelectual (DPI) restrictivos y encuadrados en una jerarquía empresarial, que guía y gobierna el proceso. Aplicando este enfoque, la actual industria del software ha crecido hasta convertirse en un coloso que, según se estima, genera más de 300.000 millones de dólares de ganancias al año.

Según el analista de la industria IDC, el sector de los paquetes de software ⁴ vale por sí solo casi 200.000 millones de dólares, mientras que el Grupo Gartner sitúa el mercado de los programas para empresas en casi 80.000 millones de dólares ⁵. El cuadro 4.1 da algunos datos sobre los mayores productores de programas patentados. Estas cifras deben tomarse con cautela, ya que no todos los ingresos obtenidos proceden de la venta de licencias de esos programas. En efecto, la consultoría y la adaptación de programas para clientes también constituyen una actividad importante. Además de las empresas enumeradas, pesos pesados de la TI como IBM, Sun Microsystems y EMC, así como los mayores productores de equipo de computadoras personales (PC), generan asimismo unos ingresos considerables a partir de los servicios de software para empresas ⁶.

Pero ésta no es la única manera de organizar la producción de programas. En los últimos años, se ha difundido otra manera de crear programas, el proceso de la fuente abierta, justo cuando los productos de este proceso, como GNU/Linux ganan cuota de mercado. En realidad, la fuente abierta no es un proceso nuevo. Es, empero, fundamentalmente diferente de la primera opción, y el éxito de los proyectos de FOSS demuestra que es posible crear, mantener, desarrollar y extender software complejos en un entorno exento de derechos de propiedad, en el que muchos programadores trabajan de manera altamente paralela, relativamente no estructurada, a menudo sin remuneración monetaria directa o inmediata.

El presente capítulo sienta y desarrolla la premisa de que el proceso de fuente abierta es un modo viable de producir programas que representa una opción real para las empresas y los gobiernos que toman decisiones respecto de las TIC, en particular en los países en desarrollo. Se propone elucidar el fenómeno mismo de los FOSS y aclarar alguna de las cuestiones que interesan cuando se trata de elegir entre los programas de fuente abierta y los programas patentados. Presenta alguno de los

parámetros y variables que pueden influir en esta elección, junto con ejemplos prácticos de las posibilidades y consecuencias de la adopción de la fuente libre, utilizando ejemplos de países industrializados y poniendo de relieve iniciativas de países en desarrollo. Finalmente, el capítulo ofrece un marco para entender las consecuencias normativas asociados con los FOSS, centrándose en las opciones que el sector público ha de tener en cuenta y en las razones que pueden influir en sus decisiones.

B. El proceso y el desafío

¿Qué es el FOSS y en qué se diferencia del software patentado que se vende bajo los regímenes convencionales de propiedad intelectual? Puede ser útil una simple analogía con cualquier popular refresco de cola⁷. Un fabricante vende botellas de soda de cola a los consumidores. Algunos de ellos tal vez lean la lista de ingredientes que figura sobre la botella, pero esa lista de ingredientes será sorprendentemente genérica. El fabricante suele tener una "fórmula" patentada que no revela. La fórmula es el conocimiento que indica cómo combinar los ingredientes en proporciones particulares, quizá con cierta mezcla aromatizante "secreta", para obtener un producto con valor comercial. No obstante, el líquido burbujeante no puede someterse a "análisis reconstructivo" de sus partes constituyentes. Se puede comprar el refresco de cola y beberlo, pero no *entenderlo* como para reproducirlo o mejorarlo, y para distribuir el producto copiado o mejorado al resto del mundo. A fin de garantizar que ninguna entidad redescubra, reconstruya o (por medios más ilícitos) adquiera la fórmula de la cola, la fórmula también es objeto de protección de la propiedad intelectual.

La economía básica de los derechos de propiedad intelectual es la razón por la que la producción se organiza de esta manera. El problema central de la propiedad intelectual gira entorno a crear incentivos para la innovación. Patentes, derechos de autor, planes de licencias y otros medios de restringir el conocimiento dan respaldo jurídico al concepto de que se crean rentas económicas y los innovadores pueden y deben quedarse con una parte de esas rentas como incentivo para innovar. Sin la protección de la propiedad intelectual, si se descubriese una fórmula "nueva y mejorada", la persona que inventase la nueva fórmula no tendría ningún

derecho económico defendible a una participación en los beneficios que pudieran obtenerse vendiendo bebidas elaboradas aplicando la innovación. Esa persona no tendría, de partida, ningún incentivo financiero para innovar, por lo que el sistema se paralizaría y la cola mejorada nunca se produciría. Aunque el productor inicial ciertamente apoya la protección de la propiedad intelectual y se beneficia de ella en la mayor medida posible, es consciente de que la seguridad de su fórmula y, en consecuencia, de su empresa, radica en su protección física, y en los costos de entrada, así como en los de fabricación y distribución, para los posibles competidores. Así pues, el fabricante complica la receta, divide la fórmula de modo que ciertas personas conozcan sólo partes de ella y nadie la conozca por entero, utiliza una buena caja de caudales y se esfuerza por lograr una posición de mercado monopolística.

La producción de los software amparados por patentes se organiza típicamente bajo un régimen análogo, con un argumento paralelo añadido. Al comprar un programa, las personas o las empresas compran de hecho una licencia que les da derecho a usarlo. No son dueños del programa en el sentido de que puedan hacer lo que quieran con él. La licencia les permite utilizar en una computadora el software patentado, pero sólo en condiciones muy concretas: no pueden reproducirlo, modificarlo, ni mejorarlo, y no pueden redistribuir entre terceros su propia versión del programa. Los derechos de autor, las licencias, las patentes y otras estructuras jurídicas proporcionan una capa de protección jurídica de este régimen, pero hay un mecanismo aún más fundamental que impide a los titulares de licencias hacer cualquiera de estas cosas. Así como el productor de la bebida de cola no publica su fórmula, un productor de un software patentado no da a conocer el código fuente de los programas.

Los programadores *escriben el código fuente* utilizando un lenguaje de programación. Las computadoras *ejecutan* los programas en formato de *código binario*.

El código de fuente es una lista de instrucciones que forman la "receta" para una aplicación informática concreta, como un procesador de textos o una hoja de cálculo. Los ingenieros de programas escriben el código fuente utilizando un lenguaje de programación particular (como C++ o Fortran) que los expertos pueden leer y entender, así como arreglar y modificar. Para los que son expertos, el código fuente parece una

combinación de lenguaje ininteligible y de expresiones matemáticas y lógicas.

Antes de que el software pueda utilizarse en una computadora, es preciso "compilarlo". La *compilación* es el proceso de traducción del código fuente a un código binario, que consiste básicamente en series de unos y ceros, después de lo cual se guarda como archivo aparte. A partir de ese momento el archivo compilado puede ejecutarse en una computadora, y se le llama *archivo binario ejecutable* o *binario*. La mayoría de los software comerciales patentados sólo se distribuyen como archivos binarios ejecutables, que el ser humano no puede "leer" o entender. La falta de acceso al código fuente restringe la capacidad del usuario de modificar los programas. Tampoco es posible generalmente reconstruir a la inversa el código de fuente a partir del código binario. Así pues, vender solamente los archivos binarios ejecutables es una manera muy eficaz en que los productores de software patentado controlan lo que los usuarios pueden o no pueden hacer con los programas que compran.

El código de fuente patentado es la piedra de toque del régimen de propiedad intelectual usual para los programas informáticos. Es una razón importante de por qué la industria del software puede generar ingresos y ganancias considerables (véase el cuadro 4.1). A su vez, estas empresas distribuyen parte de sus beneficios a los programadores que escriben los códigos y, de este modo, los incentivan a innovar.

El proceso de fuente abierta invierte esta lógica. La esencia de los FOSS es que el código fuente es "libre". Esto es, junto con los archivos binarios ejecutables que son los que realmente se ejecutan en la computadora, se facilita el código fuente⁸ para quienes quieran examinarlo, utilizarlo o modificarlo. "Libre" en este contexto significa la libertad de ejecutar el programa con cualquier fin, estudiar cómo funciona y adaptarlo a las propias necesidades, redistribuir copias a otros y mejorar el programa y compartir las mejoras con la comunidad de manera que todos se beneficien (FSF 1996). No quiere necesariamente decir que el precio sea igual a cero, porque los FOSS pueden comercializarse en los mercados igual que cualquier otro artefacto. Los programadores explican a menudo esta aparente incongruencia con los siguientes ejemplos: cuando oiga hablar en inglés de "*free software*", piense en el "*free*" de "*free speech*" (libertad de palabra), no en el de

"*free lunch*" (almuerzo gratis); o en "software libre", no en "software gratis"⁹.

Por ejemplo, las populares distribuciones FOSS GNU/Linux se venden en CD-ROM a precios que van desde algunos dólares (por los discos únicamente) hasta más de 100 dólares (por paquetes que contienen manuales y acceso a ayuda en línea por un determinado período). A menudo, los archivos binarios ejecutables pueden obtenerse sin pagar, pero para ello los usuarios deben descargarlos de Internet y grabar sus propios CD-ROM de instalación. Este método también tiene claramente un costo, que comprende el acceso a Internet (preferiblemente por banda ancha), así como una impresora, un grabador de CD y CD vírgenes. Si una empresa de TIC puede o no ganar dinero con los FOSS es una cuestión importante que se analiza en la sección E del presente capítulo.

Crear programas complejos es tarea rigurosa y difícil por la complejidad técnica y humana que entraña, tanto en la fase de concepción abstracta y como en la ejecución. La gente emplea los programas en una matriz tecnológica y cultural extraordinariamente diversa que cambia casi continuamente. Por ejemplo, si un ingeniero de automóviles ha de prever toda la gama de condiciones en que las personas tratarán de conducir un coche, el ingeniero de programas se enfrenta con una tarea más dura aún porque gran parte del entorno tecnológico en que se usará un determinado software todavía no ha sido ni siquiera producido o distribuido en el momento en que se escribe el programa. Aparte de los adelantos en el hardware, los cambios en el sistema operativo y en los entornos para el trabajo en red influirán en cómo utilizaremos los programas diseñado hoy. Las autopistas y los puentes simplemente no cambian tan rápidamente, y no son configurables por los usuarios de la manera en que lo son los programas.

El problema de la producción de software lleva inevitablemente a una división del trabajo. Las cuestiones primarias son: ¿qué tipo de división del trabajo? ¿Cómo se ha de organizar? Poner la cantidad adecuada de personas en los puestos correctos es también importante, pero es en realidad un problema secundario.

La respuesta usual a esta pregunta es la organización jerárquica de estilo fordista¹⁰. La división clara entre diseño/arquitectura y ejecución,

la segmentación de las tareas en subsistemas que luego han de encajar perfectamente entre sí y las jerarquías de mando con control desde arriba son características familiares de la organización industrial. Una autoridad asigna tareas, vigila el rendimiento y retribuye de acuerdo con indicadores de ejecución mensurables. Controlar el código fuente se transforma en un medio de controlar la división del trabajo.

El proceso del código fuente abierto enfoca este problema de una dirección diferente. Una vez que el código fuente se ha divulgado, la configuración y la gestión del trabajo pasan a ser un proyecto de los trabajadores mismos. *Los elementos clave de un procedimiento de código fuente abierto ideal son la participación voluntaria y la elección voluntaria de tareas.* Cualquiera puede unirse a un proyecto de este tipo y cualquiera puede dejarlo en cualquier momento. No hay ninguna división del trabajo organizada o impuesta. Tal vez el concepto básico de la división del trabajo no encaje bien en el proceso del código fuente abierto. El trabajo se *distribuye*, y no podría ser de otra manera tratándose de proyectos que en todo momento pueden contar con un centenar o incluso con miles de programadores. Pero no está realmente *dividido*, en el sentido industrial de la palabra. En el análisis de las estructuras de Apache y GNU/Linux que se hace en la sección F del presente capítulo se reflexiona sobre cómo las motivaciones individuales pueden traducirse en acciones colectivas ¹¹.

C. Una historia de la producción de programas

El concepto de software "libre" no es nuevo. En los decenios de 1960 y 1970, las computadoras centrales que existían en los departamentos universitarios de ciencia informática y en las instalaciones empresariales eran principalmente herramientas para la investigación. La idea del libre acceso al código fuente de los programas instalados parecía natural y a menudo se daba por descontado. El entorno de FOSS era necesario para mejorar la compatibilidad entre los diferentes sistemas informáticos para los que había que modificar los programas a fin de que funcionaran con los diferentes equipos físicos -un proceso muchas veces largo y caro. La incompatibilidad estaba en pugna con la ética científica de compartir y acumular conocimientos, así como el problema práctico de tener que reescribir grandes cantidades

de textos codificados para distintos tipos de computadora.

En los Estados Unidos, los Laboratorios Bell de AT&T señalaron el camino al centrar su esfuerzo en el desarrollo del sistema operativo UNIX y un lenguaje asociado para desarrollar aplicaciones, llamado simplemente C, que podían funcionar en equipos diferentes y variados ¹². Conforme a los términos de su acuerdo de monopolio regulado con el Departamento de Justicia, AT&T no podía dedicarse a actividades informáticas comerciales y por lo tanto no podía vender UNIX con ganancia. Pareció, pues, casi natural dar a conocer el código fuente a las universidades y a otros que, a juicio de los ingenieros de los Laboratorios Bell, podían ayudar a perfeccionar los programas hallando fallos lógicos y arreglando el código fuente ¹³. De este modo, el software UNIX, normalmente protegido por derechos de autor, se fue distribuyendo gratuitamente en la mayoría de los casos junto al código de fuente.

Incentivos concretos estimulaban este tratamiento despreocupado e informal de los derechos de autor. El comportamiento tenía sentido para el titular de los derechos, porque el software no se consideraba en esos momentos una fuente de beneficios por sí mismo, sino principalmente un anzuelo para alentar a la gente a comprar equipo físico. Mejora el software y venderás más computadoras -así se pensaba. También era razonable que un programador innovador diese gratuitamente ideas al propietario del software. Si todas o muchas de estas novedades se incorporaban en las futuras versiones del programa, los departamentos de informática no tendrían que preocuparse de integrar las mejoras poco a poco, sino que podrían simplemente esperar la siguiente versión oficial.

La lógica del software libre comenzó a desmoronarse a finales del decenio de 1960. En 1969, el Departamento de Justicia de los Estados Unidos entabló una causa civil antitrust masiva contra IBM, empujándola a separar sus "soluciones" y comenzando a cobrar aparte el software ¹⁴. IBM empezó luego a enviar sus unidades centrales con sistemas operativos que no distribuían el código fuente. Por cierto, los administradores tenían que firmar acuerdos de no divulgación simplemente para obtener una copia ejecutable. Esta decisión representa el nacimiento de la industria moderna del software comercial. Microsoft se fundó en julio de 1975 como una empresa que, a todos los fines y

efectos, sencillamente escribía y vendía programas. La llegada del PC a principios del decenio de 1980 y su rápida y amplia distribución a los escritorios del mundo empresarial reforzaron esta tendencia. El software que antes se intercambiaba libremente entre los programadores pasó a ser un producto extraordinariamente valioso y lucrativo en sí mismo. El desarrollo de una industria y un modelo empresarial aparte para el software tuvo una gran repercusión en la profesión de programador. Muchos de los mejores programadores de los Estados Unidos y de otras partes fueron contratados para puestos lucrativos en empresas de software derivadas.

Ante estas novedades, en 1984, el investigador del Instituto de Tecnología de Massachusetts Richard Stallman fundó un proyecto para reactivar las actividades de FOSS creando un completo conjunto de herramientas de programación FOSS¹⁵. Esta iniciativa llevó al

establecimiento de la Fundación para el Software Libre (FSF). La FSF utiliza exclusivamente la expresión "software libre" para denotar programas que el usuario puede ejecutar, copiar, distribuir, estudiar, cambiar y mejorar mediante el acceso al código fuente. La FSF considera los derechos de autor como un medio de aprisionar la información y de crear un acceso desigual, por lo común entre ricos y pobres. Para reemplazar los derechos de autor tradicionales, la FSF ha elaborado un acuerdo de derechos de autor normalizado, la Licencia Pública General (LPG) GNU, llamada a menudo "copyleft"¹⁶. La LPG está concebida para disuadir a los programadores de "cerrar" el código fuente de un programa FOSS e impedir que se introduzca en un entorno de desarrollo comercial con derechos de propiedad¹⁷. En el recuadro 4.1 se analiza con más detalle la principal postura de la FSF. La sección 3 del presente capítulo trata de los detalles jurídicos de la LPG (bajo el epígrafe "Derechos de propiedad intelectual").

Recuadro 4.1

La Fundación para el Software Libre y la Licencia Pública General

La idea central de la Licencia Pública General (LPG) es impedir que el código fuente de programas abiertos/libres, desarrollados cooperativamente sea "encerrado" o transformado en código de programas patentados, restrictivamente protegidos por derechos de autor. La LPG declara que se permite a los usuarios usar el programa, copiarlo, modificarlo mediante su código fuente y distribuir a terceros versiones modificadas del mismo. Lo que no pueden hacer es añadir restricciones de su propia cosecha. Esta es la "cláusula viral" de la LPG. Obliga a cualquiera que publique programas derivados que incorporen códigos sujetos a la LPG a utilizar también la LPG en la nueva publicación. La Fundación para el Software Libre declara: "Debe usted hacer que toda obra que distribuya o publique, y que, en todo o en parte, contenga el Programa [cualquier programa cubierto por esta licencia], o se derive de él, o cualquier parte de la misma, sea licenciada como un todo y sin ningún cargo a todos los terceros conforme a las condiciones de la presente licencia" (FSF 1991)¹⁸.

Stallman y la FSF crearon alguna de las partes más ampliamente utilizadas del software UNIX, incluidos el editor de texto Emacs¹⁹, el compilador GCC²⁰ y el depurador GDB²¹. Como estos populares programas fueron adaptados para funcionar en casi todas las versiones de UNIX, su disponibilidad y eficiencia ayudaron a cimentar el papel de UNIX como sistema operativo de elección para los defensores del "software libre" y las principales instituciones académicas y de investigación. Pero el buen éxito de la FSF resultó ser, en cierto sentido, autolimitante en razón de la naturaleza viral de la LPG. Su posición contra los programas patentados chocó contra la opinión utilitaria de muchos programadores, que deseaban emplear partes de los códigos protegidos junto con los códigos libres cuando tenía sentido hacerlo, simplemente porque eran técnicamente buenos. La LPG no permite este tipo de flexibilidad y plantea a veces difíciles apremios a los programadores que buscan soluciones prácticas a los problemas.

El proceso de los FOSS utiliza en amplia medida las herramientas de comunicaciones para la modificación, la innovación y la evolución respecto del código, con colaboración entre localidades diversas y alejadas. Mientras que la ARPANET era apenas suficiente, la rápida difusión de Internet en los primeros años noventa aceleró la actividad de los FOSS. El desarrollo del software del sistema operativo de PC GNU/Linux comenzó durante este período a partir de raíces honorablemente modestas.

A fines de 1990, Linus Torvalds, un estudiante de informática de 21 años de la Universidad de Helsinki, empezó a crear el núcleo de un sistema operativo de tipo UNIX en el PC de su casa. En otoño de 1991, Torvalds dio a conocer el código fuente para el núcleo de su nuevo sistema operativo llamado GNU/Linux a un grupo de noticias de Internet, junto con una nota en la que pedía observaciones y colaboradores. La respuesta fue inesperadamente buena. Para fines del año, casi 100 personas de todo el mundo se habían sumado al grupo de noticias GNU/Linux; muchas de ellas eran desarrolladores en activo que repararon fallos lógicos, mejoraron el código y aportaron nuevas funciones. Durante 1992 y 1993, la comunidad de desarrolladores aumentó gradualmente. Esto sucedía en momentos en que en la comunidad más amplia de productores de software se iba imponiendo la idea de que la era de los sistemas operativos basados en el UNIX estaba tocando a su fin, en vista de la posición cada vez más dominante de Microsoft (Raymond 2000). En 1994, Torvalds hizo pública la versión oficial 1.0 de GNU/Linux.

Mientras diversas versiones patentadas de UNIX perdieron cuota de mercado a mediados de los años noventa, GNU Linux comenzó a aumentar la suya a finales de ese decenio y es ahora el único competidor creíble de Microsoft en el mercado de sistemas operativos de PC. El crecimiento de GNU/Linux obedeció a varias causas. Muchos en la comunidad de las TIC consideraban irritante la manera en que las compañías de software patentados sacaban provecho de su propiedad intelectual (código fuente). Otros afirmaban que la calidad técnica de esos programas se resentía del estilo empresarial del proceso de desarrollo. Se decía que, por más poderosas que llegaran a ser las empresas de software patentados, simplemente no podrían emplear bastantes ensayadores, diseñadores y desarrolladores para depurar sus programas. Al mismo tiempo, las empresas de software patentados promovían sólo una

interacción limitada entre usuarios avanzados y programadores para reparar y mejorar sus programas.

Mientras GNU/Linux crecía, la naturaleza viral de la LPG, así como el rigor de la postura de la FSF, dieron lugar, a mediados del decenio de 1990, a una institución alternativa en pro del software "libre", la Open Source Initiative (OSI) (Iniciativa fuente abierta). La OSI nació en febrero de 1998 durante una reunión de varios influyentes expertos en TI convocada en respuesta a la decisión de Netscape de hacer público el código fuente de su navegador. La decisión de Netscape se consideró como una indicación del camino a seguir en la promoción del desarrollo de los FOSS, en particular de cara a la comunidad de empresas y sociedades. En vez de incluir el mensaje prescrito sobre los derechos de autor o de la LPG, la OSI pide a las entidades que distribuyen FOSS que cumplan con la definición de fuente abierta (OSD) en su declaración sobre los derechos de autor²². Mientras la LPG *exige* que toda redistribución de programas de LPG se expida sólo bajo la LPG (para impedir el "cierre" del código), la OSD *permite* la redistribución en las mismas condiciones, pero no la exige. Algunas licencias comprendidas en la OSD autorizan al programador a modificar el software y a publicar la versión modificada en otras condiciones, que abarcan la de patentarla. El recuadro 4.2 ofrece un panorama de la OSD.

Haciendo hincapié en la competitividad económica, la OSI orientó su mensaje directamente a la corriente dominante del mundo empresarial²³. El argumento fue que el proceso del código fuente abierto potenciaba la fiabilidad, reducía los costos y mejoraba las características. Más importante aún, las empresas o los gobiernos que usaran FOSS podrían evitar verse obligados a utilizar programas producidos por un monopolista dominante. Los usuarios del código fuente abierto ganarían autonomía al controlar sus sistemas de información, que eran, cada vez más, el activo básico de casi toda empresa. La OSI dirigió inicialmente su mensaje a los directores generales de las mayores empresas transnacionales, insistiendo en las diversas maneras en que las mismas empresas de TI podían generar beneficios económicos aun liberando el código fuente. Por ejemplo, un mejor software permitiría a los fabricantes de equipo vender más computadoras y otros dispositivos. Los servicios que fabricasen paquetes de soluciones de código fuente abierto, y luego los optimizasen y mantuviesen y para empresas o para la administración pública, serían extremadamente valiosos.

Recuadro 4.2

Definición de fuente abierta

La definición de fuente abierta (OSD) mantiene la siguiente postura:

- El código fuente debe distribuirse con el software o ponerse a disposición de otra manera por no más que el costo de distribución.
- Cualquiera podrá redistribuir gratis el software, sin pagar regalías o derechos de licencia al autor.
- Cualquiera podrá modificar el software o derivar de él otro programa y luego distribuir el software modificado bajo las mismas condiciones.

La OSI elimina el efecto viral de la LPG. Fuente abierta no significa sólo acceso al código fuente. La OSI "aprueba" las licencias existentes como conformes con la OSD. (Un recuento reciente encontró 21 de ellas, incluidas la LPG pero también licencias de pesos pesados empresariales como IBM, Nokia e Intel.) La OSI se propone aplicar un criterio pragmático al desarrollo de software técnicamente complejo y descarta la ideología de la FSF. No todos comparten este objetivo o lo consideran un cambio progresivo. No obstante, vale la pena recordar que la idea central que inspiró a la OSI fue muy diferente. Uno de sus fundadores, Eric Raymond, explicaba:

"Nos parecía claro, retrospectivamente, que la expresión "software libre" había hecho un tremendo daño a lo largo de los años. Esto se debía en parte a la conocida ambigüedad del inglés entre "*free*" como "libre" y "*free*" como "gratis". Pero en su mayor parte se debía a algo peor -la fuerte asociación de la expresión "*free software*" con la hostilidad contra los derechos de propiedad intelectual, [con] el comunismo y [con] otras ideas poco propicias a ser gratas a un gestor de servicios de información"(1999a).

La respuesta del mundo empresarial fue inmediata. En enero de 1998 Netscape anunció que liberaría el código fuente de su navegador web como código fuente abierto. En verano, Oracle e Informix, dos de los mayores vendedores independientes de software para aplicaciones empresariales y bases de datos, anunciaron que adaptarían sus programas a GNU/Linux. En los meses siguientes, otros vendedores independientes de software de primera línea, como Sybase y la empresa alemana SAP, hicieron anuncios similares. En el primer semestre de 1999, IBM empezó a centrarse en GNU/Linux como sistema operativo para sus servidores (Berinato 1999, 2000). IBM también se convirtió en uno de los principales promotores de las supercomputadoras de Beowulf (CNET de 2000). Los grandes vendedores estadounidenses de hardware (Compaq, Dell, Hewlett Packard, Silicon Graphics), así como los fabricantes de chips Intel y AMD, se han comprometido seriamente con GNU/Linux. En los últimos años noventa empezaron a funcionar empresas comerciales que prestan servicios auxiliares y apoyo a GNU/Linux, tales como Red Hat Software en los Estados Unidos, SuSe en Alemania y MandrakeSoft en Francia. Apache siguió aumentando su predominio en el mercado de servidores de Internet, al tiempo

que los usuarios de ésta aumentaban a un ritmo exponencial. En octubre de 2000, Sun Microsystems liberó el código fuente de StarOffice, un conjunto de software de uso cotidiano en las oficinas, para establecer OpenOffice.org. En el recuadro 4.3 se describen estas y otras aplicaciones importantes o muy difundidas de FOSS.

Microsoft empezó a ver en el proceso de fuente abierta en general y en GNU/Linux en particular una amenaza importante y real a la presencia en el mercado de sus sistemas operativos Windows, sus servidores y tal vez incluso sus PC de escritorio²⁴. Un memorando interno de alto nivel de Microsoft, escrito en 1998, se filtró el 31 de octubre y pasó a ser conocido como el "Memo de Halloween". Al parecer describía a los FOSS como una amenaza directa a corto plazo para sus ingresos y su posición dominante en algunos mercados. También planteaba una cuestión estratégica a largo plazo, ya que, según el memorando, "el paralelismo intrínseco y el intercambio libre de ideas en los OSS [FOSS] ofrece ventajas que no se dan en nuestro actual modelo de licencias [de patentes en propiedad]"²⁵.

La breve historia del software, desde sus raíces de fuente abierta hasta los modelos

patentados y, ahora, el viaje de vuelta a la fuente abierta, ha tenido lugar sobre todo en los Estados Unidos. Lancashire (2001) corrobora ese hecho y da algunos datos geográficos para los desarrolladores participantes. En cierto sentido, este hecho se explica por sí mismo, ya que la mayoría de los desarrolladores FOSS estarán radicados en los países con las empresas de software más avanzadas. Si volvemos al cuadro 4.1, vemos que de las diez primeras empresas mundiales de software y de sus diez principales competidores, sólo tres no tienen su sede en los Estados Unidos y únicamente hay una en un país en desarrollo²⁶. No obstante, la situación parece estar cambiando rápidamente y, a juzgar por los resultados de la encuesta de la que se habló en la sección H de este capítulo, las actividades de FOSS en los países en desarrollo podrían adquirir cada vez más visibilidad en un próximo futuro.

A finales de los años noventa el procedimiento de FOSS había demostrado su viabilidad como medio para crear paquetes complejos de software que pudieron competir con éxito con los productos patentados y en un creciente número de sectores del mercado de TI, desde aplicaciones para procesar datos incorporados en terminales pequeños hasta supercomputadoras interconectadas. Compañías tan diversas como la distribuidora Red Hat de GNU/Linux y el gigante tradicional de la TI, IBM, han aprendido a obtener buenos beneficios brindando servicios con diversas clases de FOSS. Ahora está claro que hay al menos dos modelos distintos para organizar la producción de software. Los dos parecen sostenerse. En la actualidad, las administraciones públicas, las empresas y casi todas las personas que usan software pueden elegir, y tendrán que hacerlo, entre productos generados por uno u otro procedimientos de creación de software.

D. ¿Son mejores los FOSS?

La cuestión de fondo que el procedimiento de fuente abierta tiene que resolver es la de lograr una calidad igual o superior a la conseguida por las empresas de programas patentados. Tal cosa puede hacerse de cuatro modos.

1. Sin perjuicio de que todo software presenta errores de programación y problemas de estabilidad, los FOSS pueden contar con más

programadores que examinen críticamente los problemas y propongan soluciones que cualquier empresa de software patentado. En otras palabras, "con suficientes ojos, todos los errores se acaban resolviendo" (Raymond 2000).

2. Como los FOSS no se ven obstaculizados por la dinámica empresarial y comercial de maximizar el ingreso de las ventas por licencia, los desarrolladores pueden libremente dar solución a los errores, introducir correcciones y ofrecer versiones nuevas con mayor frecuencia.
3. La instalación de un software patentado después de la adquisición de una licencia de uso está a menudo vinculada a aceptar modalidades y condiciones que exoneran de toda responsabilidad por daños y perjuicios como consecuencia del uso, más allá de la sustitución del disco duro donde se haya instalado el software, lo que claramente no es una garantía de calidad con la que sea difícil competir.
4. La disponibilidad de un código fuente es por sí misma un importante elemento de calidad del producto. Imaginemos una compañía de transporte que adquiere una flota de vehículos que le llegan sin el mecanismo para abrir el capó del motor; esos mecanismos serían inútiles porque la compañía se ha comprometido por contrato con el fabricante a que de ningún modo intentará arreglar o inspeccionar los motores de los vehículos. Éstos, al igual que el software análogo, serán obviamente de calidad inferior.

Sin embargo, el procedimiento del FOSS, no está exento de equivocaciones o fallos. Un posible problema es la fragmentación y ramificación de los proyectos: un equipo de trabajo puede tropezar con dificultades técnicas e incluso con problemas de personalidad. La fragmentación o ramificación significa que los recursos de desarrollo existentes se dividen entre los equipos principales y los discrepantes, teniendo quizá los usuarios que enfrentarse a decisiones no deseadas y a cuestiones de compatibilidad. Otro problema citado es que resulta difícil para los usuarios predecir claramente a dónde puede estar conduciendo el desarrollo en términos de versiones futuras, de funcionalidad o de hardware de apoyo. Por último, los desarrolladores y los directores de los equipos de los proyectos pueden simplemente perder interés o iniciar otra vía que ya no sea pertinente para el

proyecto de software. No obstante, tales problemas no se dan exclusivamente en los FOSS. El software de propiedad tiene también una buena parte de normas y compatibilidades que difieren. A menudo, hay buen software producido por compañías que no logran resultados financieros comparables, lo que obliga a los consumidores a cambiar a productos de empresas mejor gestionadas. El software de apoyo para hardware nuevo en el mundo del software patentado está muchas veces condicionado a la "opción" forzada de renovar y volver a pagar las licencias.

Ningún software es inequívocamente "mejor" que todos los demás. Como cualquier instrumento, el software presenta ciertos rasgos de mayor o menor facilidad de uso y de más o menos fiabilidad, flexibilidad, robustez y costo. No hay un único equilibrio óptimo entre esas características, y mucho dependerá de las necesidades que tenga cada usuario. A igualdad en todos los demás aspectos, sin embargo, el software con menos errores graves y un costo total de propiedad inferior suele ser preferible por simples motivos económicos. Con todo, incluso esos criterios son difíciles de medir. Una prueba bastante utilizada de robustez es el tiempo medio desde la última reiniciación. En el cuadro 4.2 se indican los servidores web con el mayor tiempo desde la última reiniciación medido durante la semana del 18 de agosto de 2003, junto con el sistema operativo y el software servidor que usan. Resulta notable que sólo 1 de los 20 servidores de Internet más robustos trabajen con software patentados.

Como los errores del software aparecen en entornos diversos, no existe un medio preciso o fiable de estimar su alcance o gravedad en un programa concreto. Mayor importancia tiene la rapidez con que un error, una vez detectado, puede resolverse. Un estudio reciente comparó la solución de errores de tres parejas de FOSS y programas patentados de fuente cerrada: dos servidores web, dos sistemas operativos y dos paquetes de interfaces gráficas de usuarios. Se encontró algún respaldo para la hipótesis de que los errores comunicados respecto del software de fuente abierta se resuelven más aprisa que las peticiones a los servicios de fuente cerrada, después de controlar de algún modo la prioridad y gravedad de cada notificación²⁷. Esta conclusión

debe tomarse con cautela, ya que puede ocurrir que los errores se descubran también a un ritmo diferente, y presenten rasgos de complejidad distintos según los tipos de software. Este resultado se compadece con la expectativa de que los usuarios se sentirán más motivados para solucionar los problemas que se les presenten si se les da la posibilidad de hacerlo mediante el acceso al código fuente.

Los cálculos de los costos totales de posesión (CTP) intentan captar en su totalidad el costo de desplegar, mantener y usar un sistema durante toda su vida útil. Los estudios de CTP para paquetes de FOSS han dado resultados controvertidos, en parte porque la estructura de costos de las mejoras y el mantenimiento es algo opaca en comparación con los precios de los productos patentados. En la adquisición, las soluciones de fuente abierta a menudo cuestan menos, según el tipo de adaptación al cliente y de los servicios adicionales que elija comprar una organización. El despliegue requiere frecuentemente una capacitación, que a veces es igual o más costosa con el FOSS que con las soluciones patentadas. En el uso y mantenimiento, que es cuando el grueso de los CTP se materializan, los FOSS pueden presentar ventajas apreciables. En esos aspectos, los gastos dependerán en definitiva de los costos laborales locales, lo que en muchos países en desarrollo puede favorecer el recurso a los FOSS. La disponibilidad del código fuente permite utilizar los conocimientos propios para resolver los errores o cambiar las configuraciones, como también contratar apoyo externo en un mercado competitivo en el que cualquiera puede entrar. Lo que parece claro es que los FOSS pueden ayudar a que una empresa o institución pública no quede encerrada en un círculo vicioso de mejoras del hardware y el software y de cambios en los formatos de los datos que requieren nuevas inversiones en licencias y un reciclaje apreciable, y que pueden dar lugar a largos tiempos de inactividad.

A la postre, los mercados de software decidirán qué procedimiento es el mejor, siempre que la piratería, las prácticas anticompetitivas y los monopolios puedan ser frenados por los reguladores oficiales. El constante crecimiento de la cuota de mercado del sistema operativo GNU/Linux indica que muchas organizaciones

Cuadro 4.2

Servidores web con el mayor tiempo medio desde la última reiniciación

	Sitio	Tiempo medio desde la última reiniciación (días)*	Sistema operativo	Software servidor
1	<i>www.daiko-lab.co.jp</i>	1.569	FreeBSD	Apache/1.2.4
2	<i>www.rfj.ac.se</i>	1.389	BSD/OS	Apache/1.3.26 (Unix)
3	<i>amedas.wni.co.jp</i>	1.360	FreeBSD	Apache/1.3.26 (Unix)
4	<i>www.alfaoffset.se</i>	1.347	BSD/OS	Apache/1.3.26 (Unix)
5	<i>www.sisu.ac.se</i>	1.320	BSD/OS	Apache/1.3.26 (Unix)
6	<i>www.lobomar.se</i>	1.319	BSD/OS	Apache/1.3.26 (Unix)
7	<i>d1o20.telia.com</i>	1.309	BSD/OS	Apache/1.3.26 Ben-SSL/1.48 (Unix) PHP/3.0.18
8	<i>treefort.org</i>	1.298	FreeBSD	Apache/1.2.6
9	<i>www.treefort.org</i>	1.298	FreeBSD	Apache/1.2.6
10	<i>www.21stcenturycomputers.com</i>	1.283	BSD/OS	Apache/1.3.26 (Unix) mod_ssl/2.8.10 OpenSSL/0.9.6g
11	<i>www.wycomp.com</i>	1.282	BSD/OS	Apache/1.3.26 (Unix) mod_ssl/2.8.10 OpenSSL/0.9.6g
12	<i>www.dir.telia.com</i>	1.272	BSD/OS	Apache/1.3.26 (Unix)
13	<i>www.21net.ne.jp</i>	1.155	FreeBSD	Apache/1.3.9 (Unix)
14	<i>www.helmarparts.com</i>	1.149	BSD/OS	Apache/1.3.23 (Unix)
15	<i>www.lan.ne.jp</i>	1.113	FreeBSD	Apache/1.2.6
16	<i>dbtech.net</i>	1.028	BSD/OS	Apache/1.3.27 (Unix)
17	<i>www.icard.com.hk</i>	1.023	BSD/OS	Apache/1.3.26 (Unix)
18	<i>www.alasearch.com</i>	1.015	BSD/OS	Apache/1.3.27 (Unix)
19	<i>www.murrayfin.com</i>	1.000	BSD/OS	Apache
20	<i>www.ehokenstore.com</i>	999	BSD/OS	Oracle_Web_Listener/4.0.8.1.0EnterpriseEdition

* Se trata del tiempo medio transcurrido desde la última reiniciación de la computadora frontal (*front-end*) o de las computadoras que acogen un sitio.

Fuente: Netcraft. <http://uptime.netcraft.com/up/today/top.avg.html>. El acceso se hizo el 28 de agosto de 2003.

piensan que los procedimientos de fuente abierta producirán con el tiempo mejores soluciones para sus necesidades de TI. En general, el software patentado no está quitando cuota de mercado a los programas de fuente abierta cuando existen soluciones de FOSS. La prueba final de la calidad está en las cifras, y en la sección siguiente se describe el desempeño de los FOSS en diversas tareas de las TIC.

E. Los FOSS en los mercados

Los FOSS son muy comunes, pero los usuarios no expertos quizá no estén muy familiarizados con ellos porque todavía no se usan mucho en los PC de escritorio en forma de

sistemas operativos o de aplicaciones de software para oficina, tales como procesadores de textos u hojas de cálculo. Las estimaciones corrientes dan a Windows de Microsoft una cuota de mercado de poco más del 90%, mientras que el resto se divide entre Apple Macintosh y los sistemas basados en GNU/Linux. Informes recientes indican que hasta el 15,5% de las empresas están pensando en cambiar a GNU/Linux²⁸. El que esto ocurra es otro cantar. No obstante, puesto que muchos usuarios llevan a sus hogares los hábitos de TI del lugar del trabajo (juntamente con software pirateado), la penetración de Linux en las computadoras de escritorio de empresas y administraciones públicas puede generar un crecimiento adicional en el mercado de las computadoras usadas en los

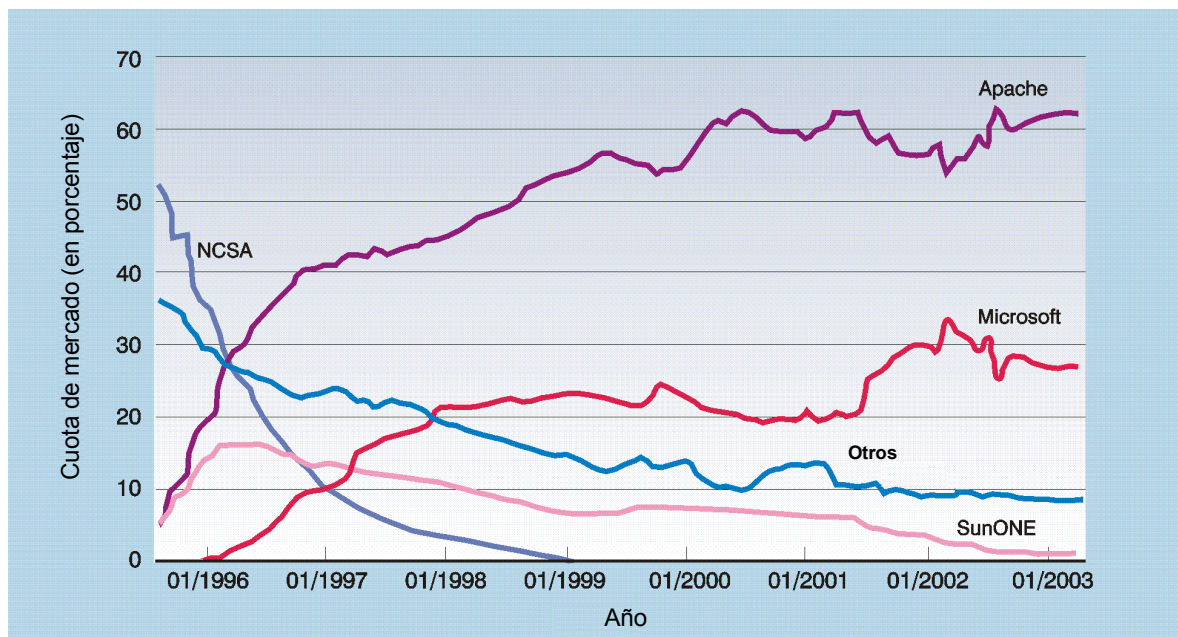
hogares. La cuestión de medir esa penetración se complica por el hecho de que si bien muchas instalaciones de GNU/Linux se descargan de Internet, no está claro si llegan a instalarse y, en caso afirmativo, dónde, es decir, sobre instalaciones patentadas o de FOSS ya existentes o en computadoras nuevas.

Incluso así, muchos usuarios pueden no tener conciencia de que utilizan regularmente programas y formatos de datos de FOSS al navegar en Internet y usar el correo electrónico, las dos aplicaciones más comunes de la tecnología informática en el hogar, que no serían posibles sin los FOSS. En esta subsección se explica por qué los FOSS se imponen cada vez más.

El crecimiento y, en algunos casos, la prevalencia de los FOSS en sectores importantes de la TI son notables²⁹. El servidor web de fuente abierta Apache, que envía páginas web a la computadora de quien accede a un sitio, ha dominado su sector del mercado desde 1996 y ahora tiene por lo menos dos veces más cuota de mercado que su competidor más próximo. Una encuesta publicada en junio de 2003 sobre la cuota de mercado de los servidores activos de Internet arroja cifras similares, con un 65,3% para Apache³⁰. En el gráfico 4.1 se indican las cuotas de mercado de los software servidores de la Web desde 1996 hasta abril de 2003.

Gráfico 4.1

Cuotas de mercado del software servidor de la Web



Fuente: Netcraft <http://www.netcraft.com>.

GNU/Linux es conocido desde hace tiempo como sistema operativo³¹ para computadoras que actúan de servidores de Internet. Encuestas recientes muestran que GNU/Linux es el sistema operativo del 29,6% de los servidores, mientras las diversas versiones de Windows lo son del 49,6%, correspondiendo a la versión patentada de UNIX (Solaris) de Sun el 7,1% y a varios derivados BSD (que son, como GNU/Linux, de fuente abierta) el 6,1%³².

En los últimos años GNU/Linux ha penetrado cada vez más en los extremos altos y bajos del mercado empresarial de sistemas operativos. Casi el 40% de las grandes compañías estadounidenses y el 65% de las japonesas usan de alguna forma GNU/Linux, que podría tener ahora hasta un 15% del amplio mercado de servidores (*Business Week* 2003). Un estudio de octubre de 2002 determinó que el 59% de los desarrolladores de software encuestados a nivel internacional esperaban escribir aplicaciones de GNU/Linux en

algún momento del año siguiente³³. La encuesta FLOSS patrocinada por la UE (Berlecon/III 2002) arrojó que el 43,7% de las compañías alemanas y el 31,5% de las británicas usaban FOSS. Resulta notable que, según varios estudios, los proveedores de servicios de Internet, las grandes y pequeñas empresas, las entidades financieras, los minoristas

y el sector público creen que el uso de GNU/Linux aumentará rápidamente tanto en sus propias organizaciones como en el mercado en su conjunto en los próximos años³⁴. En el recuadro 4.3 se hace un examen más detallado de los FOSS importantes que están disponibles y que se usan hoy día.

Recuadro 4.3

Ejemplos de software libre y de código fuente abierto

- El software de código fuente abierto se utiliza muchas veces en entornos de importancia fundamental. Muchas aplicaciones industriales estándar son de hecho programas de fuente abierta. A continuación figura una lista de algunos programas de fuente abierta importantes, además de GNU/Linux y Apache, ya descritos en este capítulo.
- La familia BSD/OS/FreeBSD/NetBSD/OpenBSD³⁵ está constituida por sistemas operativos basados en UNIX, de código fuente libre/abierto, similares a GNU/Linux. Desarrollado en la Universidad de California en Berkeley en los años setenta, el BSD se considera uno de los sistemas operativos más seguros y estables y está presente en un alto porcentaje de servidores de Internet. La base del sistema operativo Macintosh de Apple, Darwin, se basa en FreeBSD y se ha mantenido en el ámbito de los sistemas de fuente abierta. (En el cuadro 4.2 figuran pormenores sobre las actividades de fuente abierta de Apple).
- GNU fue el predecesor de GNU/Linux. Es una versión libre de los instrumentos UNIX creada por Richard Stallman en 1984. GNU es la sigla de "GNU no es UNIX".
- Sendmail es un programa de código fuente libre/abierto utilizado para dirigir aproximadamente el 40% del correo electrónico que discurre por Internet.
- Perl (Practical Extraction and Report Language) es un lenguaje descriptivo disponible libremente para UNIX, MS/DOS, Macintosh, OS/2 y GNU/Linux, entre otros. Tiene potentes funciones de manipulación de textos y se usa mucho para programar formularios electrónicos en la Web, y en general para producir interfaces entre sistemas, bases de datos y usuarios que intercambian datos en Internet.
- BIND (Berkeley Internet Name Domain) es un programa de código fuente libre/abierto que permite introducir los nombres de dominio en Internet como nombres en forma de texto en lugar de como direcciones IP, o series de números, facilitando a los usuarios el acceso a los sitios de Internet.
- El Proyecto Beowulf es un método para conectar computadoras y formar una de alto desempeño (conglomerado Beowulf) cuyo rendimiento se aproxima al de una "supercomputadora". Como un conglomerado Beowulf puede desarrollarse con computadoras corrientes que utilizan FOSS, es posible establecer y hacer funcionar una "supercomputadora" con un conglomerado Beowulf a un costo inferior al de otros sistemas con capacidad de procesamiento similar.
- OpenOffice.org es un conjunto de software que permite una automatización básica en trabajos administrativos o de oficina. Como producto del StarOffice de Sun Microsystems, se aplica a todos los principales sistemas operativos, incluido MS Windows, ya que su funcionalidad de plataforma transversal se basa en formatos de archivos estándar XML abiertos.
- GNOME y KDE son sistemas GUI de escritorio que operan sobre GNU/Linux y UNIX, proporcionando cálculos de fácil uso para los usuarios de la comunidad de no programadores que utilizan una fuente abierta.
- MySQL es un servidor de bases de datos conectadas.
- El Gimp es un programa de gráficos ampliamente distribuido con GNU/Linux. (También existe una versión para el sistema operativo Windows). A veces se denomina "free Photoshop".

Amazon, E*TRADE, Reuters y Merrill Lynch son ejemplos de compañías multinacionales que han adoptado recientemente el software de los servidores web GNU/Linux y Apache para sus sistemas informáticos posteriores (*back-end*). En buena parte de los organismos y departamentos del Gobierno de los Estados Unidos, como el Departamento de Defensa, el Departamento de Energía y la Agencia Nacional de Seguridad, se trabaja con FOSS. Administraciones públicas nacionales, regionales y municipales, en países como China, Alemania y el Perú, están estudiando el uso de FOSS en aplicaciones de gestión informatizada y, en algunos casos, los están utilizando. En el examen de la sección H del presente capítulo figura una descripción más detallada de la participación de los países en desarrollo en los FOSS.

F. La razón de ser de los FOSS

Si bien está claro que hay razones convincentes para querer *utilizar* los FOSS, otra cuestión importante es por qué alguien querría *producirlos*, y cómo esa motivación se traduce en una producción coherente. Antes de examinar las motivaciones, merece la pena examinar algunas indicaciones e informaciones del mundo real.

1. Información de dominio público

Los usuarios rara vez se limitan a comprar licencias de software; también adquieren servicios conexos. Las organizaciones y empresas normalmente compran soluciones que entrañan una combinación de software, hardware y servicios. Los servicios que acompañan a los productos de software van desde consultoría, ejecución, apoyo y formación hasta administración de las aplicaciones. En realidad, incluso Microsoft ha concedido a lo que parece que, según revela una encuesta del Grupo Gartner, el costo de las licencias de software supone sólo el 8% del costo total de éste y el otro 92% corresponde a los gastos de instalación, mantenimiento, gestión y reparación³⁶. En lo que parece ser una estimación equivalente, Raymond (1999b) sostiene que una proporción muy pequeña del software, tal vez menos del 10%, se desarrolla para su venta al por menor como

producto preempaquetado. La gran mayoría consiste en códigos internos que están tan integrados en las actividades de las empresas y en los entornos de TI que reutilizar o copiar el código "tal cual" resulta difícil o inviable.

La conclusión es clara: La gran mayoría del dinero que se gana desarrollando software no corresponde a la venta de licencias de programas preempaquetados. La idea contraria se basa en el hecho de que el software patentado preempaquetado genera grandes ingresos; pero, ello ocurre porque unos pocos productores pueden aplicar precios de monopolio. Para una empresa que presta servicios de TI, los ingresos adicionales que reportan las comisiones que recibe por la reventa de licencias pueden ser tan marginales que no influyen apreciablemente a la hora de elegir un sistema patentado en lugar de una plataforma de FOSS para un determinado cliente. Lo que influirá en esa elección es con cuánta precisión pueden atender las demandas de los clientes y el grado de adaptación, la facilidad de mantenimiento y la robustez de la plataforma. Desde el punto de vista de los países en desarrollo, esta cuestión por sí sola es suficiente para disipar la preocupación de que el uso de plataformas de FOSS puede reducir las oportunidades empresariales.

Corroborar esta idea el hecho de que gran parte del sector de la TI está desarrollando actividades basadas en FOSS. La IBM es ahora uno de los principales campeones del software de fuente abierta, después de asignar públicamente en 2001 la suma de 1.000 millones de dólares a desarrollar la tecnología adecuada y a reconfigurar partes centrales de sus modelos empresariales para GNU/Linux y otros programas de fuente abierta. Ya en 2002, la IBM anunció que había obtenido ingresos de más de 1.000 millones de dólares con las ventas de software, hardware y servicios basados en Linux³⁷. Otros líderes de la tecnología, como Hewlett-Packard, Motorola, Dell, Oracle, Intel y Sun Microsystems, han hecho importantes inversiones en FOSS para sistemas operativos, sistemas integrados, supercomputadoras múltiples y software de aplicaciones empresariales. En el cuadro 4.3 figura un panorama más detallado de la participación de las principales empresas de TI en los FOSS.

Cuadro 4.3

Participación de los líderes del sector de la tecnología de la información en los FOSS

Compañía	Participación en los FOSS
IBM	IBM tiene una variedad de proyectos de fuente abierta, todos con licencias de fuente abierta aprobadas por la OSI. http://www-124.ibm.com/developerworks/oss/
Microsoft	Microsoft propone un modelo de "fuente compartida" como una alternativa a la fuente abierta. http://www.microsoft.com/licensing/sharedsource/ La tecnología Interix de Microsoft, ahora integrada en Servicios Windows para UNIX 3.0, proporciona un medio, con licencia LPG, para ejecutar aplicaciones de Windows y de UNIX en un mismo sistema. http://www.microsoft.com/windows/sfu/howtobuy/default.asp
Pricewaterhouse-Coopers	Los temas de los FOSS se examinan en el sitio desde la perspectiva de una consultoría. http://www.pwcglobal.com/Extweb/service.nsf/docid/30F66202E467710C85256B990072FC55
EDS	Ocasionalmente desempeña actividades de FOSS. Dynamator, un programa de mantenimiento de la página del servidor desarrollado por un programador EDS, es un FOSS. http://www.eds.com/about_eds/homepage/home_page_dynamator.shtml
Oracle	Oracle no tiene actividades de FOSS visibles pero sí productos de bases de datos adaptados a Linux. http://www.oracle.com/linux/
Hewlett-Packard	Hewlett-Packard tiene varios proyectos de FOSS. http://opensource.hp.com/
Accenture	El tema de los FOSS se examina en el sitio desde la perspectiva de una consultoría. http://www.accenture.com/xdoc/en/ideas/outlook/pov/open_source_pov_rev.pdf
SAP	mySAP Business Suite opera con Linux. http://www.sap.com/solutions/netweaver/linux/ SAP DB es una base de datos para empresas de fuente libre/abierta. http://www.sapdb.org
Computer Associates	Es un cofundador de Open Source Development Lab. http://www.osdl.org
Hitachi	Hitachi participa en proyectos de FOSS. http://oss.hitachi.co.jp/index-e.html
Sun Microsystems	Sun patrocina varios proyectos de FOSS, como OpenOffice.org y NetBeans. http://www.sunsource.net
Compuware	Compuware no tiene actividades de FOSS, pero el entorno de desarrollo enviado con su producto Optimal J se basa en el entorno de desarrollo integrado (IDE) de fuente abierta NetBeans. http://www.compuware.com/products/optimalj/1811_ENG_HTML.htm
BMC Software	BMC Software está cooperando con el Open Group para desarrollar un servicio de gestión de fuente abierta. http://www.bmc.com/corporate/nr2001/032701_2.html http://www.opengroup.org/
EMC	EMC no desempeña actividades de FOSS visibles, pero el desarrollo de FOSS forma parte de la descripción de funciones para los puestos actualmente vacantes. También ha adaptado ciertos productos a Linux. http://www.emc.com/technology/auto_advice.jsp
Cadence Design	Cadence apoya un intercambio abierto entre desarrolladores internos, desarrolladores comerciales y profesores universitarios. La biblioteca de clases y banco de pruebas Systems TestBuilder C++ está disponible con licencia de fuente abierta. http://www.testbuilder.net Cadence participa en la coalición OpenAccess para bases de datos de diseño electrónico estándar. http://www.cadence.com/feature/open_access.html y http://OpenEDA.org
Adobe	Adobe tiene actividades FOSS ocasionales, que se centran mayormente en conexiones Pitón para productos Adobe. http://opensource.adobe.com/
Silicon Graphics SGI	SGI apoya un gran número de proyectos de fuente abierta. http://oss.sgi.com/
Apple	Darwin es el núcleo del sistema operativo Mac OS X de Apple. Basado en FreeBSD, Darwin se mantiene en el ámbito de los dominios de fuente abierta con la Licencia de Fuente Pública Apple. Apoya varios otros proyectos de fuente abierta. http://developer.apple.com/darwin/

2. Motivaciones de la oferta

El software es un producto digital que puede copiarse un número infinito de veces sin costo alguno, sin que disminuya la calidad o la utilidad, lo que lo convierte en un bien indivisible en términos económicos. Liberar el código fuente hace también que el software no sea excluyente, y que adquiera, por ende, las características de un bien público³⁸. Sin embargo, los bienes públicos normalmente fomentan el consumo gratuito. ¿Por qué contribuiría la gente voluntariamente a un bien público que sin necesidad de ello podrían usar gratuitamente? Si todos adoptaran la misma actitud, el sistema se desmoronaría hasta el punto en que nadie aportaría contribuciones sustanciales y el bien nunca se produciría. ¿Por qué programadores de mucho talento prefieren dedicar una parte sustancial de su tiempo e inteligencia, recursos ambos escasos y valiosos, a un proyecto conjunto del que no sacarán ningún beneficio directo?

Se han desplegado muchos esfuerzos para explorar los motivos de los desarrolladores. Algunos estudios afirman que pueden explicarse con la teoría económica corriente. Un código fuente abierto de un programador suele estar muy asociado a su autor y estar bien reconocido, proporcionando un cierto nivel de gratificación del ego. Los directores de personal de las compañías comerciales examinan frecuentemente las contribuciones a proyectos de FOSS y la participación en ellos al evaluar los méritos. Las personas con autoridad reconocida en fuentes abiertas pueden acceder a financiación y captar la atención del capital empresarial. Sun y Netscape fueron creadas por personas que habían sido programadores de fuente abierta. Así, los incentivos profesionales pueden ocupar un lugar prominente entre los motivos que mueven a los programadores a contribuir. Estos hechos, denominados a menudo "incentivos indicativos", pueden aparecer cuando las aportaciones pueden juzgarse y recompensarse en uno o varios períodos futuros, incluso aunque de momento no haya ningún contrato (Lerner y Tirole 2000 y 2001; Holmström 1999).

Raymond (1999b) explica el procedimiento de fuente abierta como una economía de donaciones donde los programadores hacen contribuciones voluntarias como una reacción a la abundancia más bien que a la escasez, correspondiendo la abundancia al conocimiento y la información, así como a la anchura de banda y a la capacidad de procesamiento. Esto implica la existencia de situaciones con

ganadores-indiferentes (es decir, beneficio sin costo), ganadores-ganadores (beneficio mutuo) y ganadores-perdedores (beneficio con un costo, donde éste ha de reembolsarse financieramente).

Otro modo de racionalizar la existencia de FOSS es el llamado modelo de cocina (Ghosh 1998). Este modelo sugiere que los FOSS se presentan como un resultado directo de la estructura distribuida de Internet, donde los usuarios no desean pagar o cobrar por los bienes y servicios que se desenvuelven en la Red. No se trata de una economía de trueque, ya que no requiere transacciones bilaterales. Además, al descartarse la igualdad entre lo que no cuesta y lo que es valioso, cobra sentido el hecho de que millones de personas en Internet publiquen sobre asuntos que les interesan y contribuyan a las comunidades, incluidas aquellas que participan en software FOSS. Si bien no obtienen ningún dinero a cambio, su "remuneración" podría revestir la forma de contribuciones complementarias de terceros, o consistir en estima y atención, que suponen un logro valioso. A decir verdad, se ha sugerido que lo que hoy escasea es la atención, mientras que factores como información e incluso financiación se están volviendo más abundantes, pese a estar distribuidos desigualmente (Goldhaber 1997).

Otros estudios se han centrado más en la comunidad real de desarrolladores de FOSS. El desarrollador típico de GNU/Linux es una persona que se siente parte de una comunidad técnica, que está muy interesada en mejorar sus aptitudes de programación, en que se facilite su labor mediante un software mejor y en tener experiencias intelectuales y sociales gratas. Esa persona reconoce los costos de oportunidad de la programación de fuente abierta en términos del tiempo y el dinero invertidos, pero simplemente no parece valorar esos aspectos (particularmente en términos financieros) tanto como los profesionales propiamente dichos³⁹. El aprendizaje individual, la eficiencia en el trabajo y los motivos colectivos o "prosociales" son las razones principales de que los programadores decidan aportar tiempo y esfuerzos a proyectos de FOSS. En el recuadro 4.4 se describen dos encuestas recientes sobre los motivos de los desarrolladores.

3. De la motivación a la producción

El estudio de los motivos personales proporciona información interesante acerca de lo que los desarrolladores piensan sobre sus elecciones individuales. Sin embargo, los motivos personales no se traducen por sí mismos en una

Recuadro 4.4

¿Qué motiva a los desarrolladores de fuente abierta?

Una encuesta de 2001 del Grupo de Consultoría Boston aportó más ideas al segmentar las respuestas de los desarrolladores en cuatro grupos característicos⁴⁰. Alrededor de un tercio de los encuestados son "creyentes" que dicen estar muy motivados por la convicción de que los códigos fuente deberían ser abiertos. Una cuarta parte son "buscadores de satisfacciones" que contribuyen a los códigos sobre todo por razones de estímulo intelectual. Alrededor de un quinto son "profesionales" que trabajan con fuente abierta porque ello los ayuda en su trabajo. Otro quinto son "promotores" de conocimientos que destacan el aprendizaje y la experiencia que obtienen con la programación de fuente abierta. La encuesta también reveló que los programadores de fuente abierta parecen agruparse mayoritariamente (70,4%) en el grupo de edad de 22 a 37 años, con un 14% más joven o de más edad. Pocos son novicios: más de la mitad son programadores profesionales, administradores de sistemas o personas que trabajan en TI. (Sólo el 20% se identifica como estudiantes).

Un estudio de 2002 patrocinado por la Unión Europea (FLOSS) versó sobre unos 2.800 desarrolladores en línea⁴¹. Ese estudio muestra un grupo constituido predominantemente por varones y todos con menos de 40 años de edad. Alrededor de un tercio de los encuestados son licenciados universitarios, otro 28% ha cursado un máster y un 9% tiene un doctorado. La gran mayoría trabaja en el sector de la TI para compañías privadas o universidades, o por cuenta propia. Los estudiantes son el 17% y los desarrolladores en paro un 4%. Están muy distribuidos en muchos países, sin que predominen los Estados Unidos, y muestran una gran movilidad al desplazarse de un país a otro para trabajar en distintos lugares.

Todas estas encuestas debería examinarse con cautela, ya que las muestras de las que obtienen los datos pueden estar sesgadas por la distribución de la encuesta, por niveles de respuesta inadecuados y por otros tipos de sesgos en la selección que hacen difícil una interpretación precisa.

acción coordinada en gran escala. La organización de la comunidad ha recibido menos atención, aunque es igual de importante. En casi todos los proyectos de fuente abierta participa un pequeño número de desarrolladores. Esos proyectos suelen depender de una comunicación intensiva y de la capacidad de persuasión de quien dirija *de facto* el proyecto para coordinar la labor del grupo. En los proyectos mayores se han establecido estructuras de gestión más explícitas y formales.

Lo que distingue a esas estructuras es un cambio sutil en la autoridad para tomar decisiones y en las relaciones jerárquicas. La idea de que no existe jerarquía en la división del trabajo está en el corazón mismo de los procedimientos de fuente abierta. Con todo, puede haber una jerarquía en la toma de decisiones para comprobar e incorporar los resultados del trabajo distribuido. Sin embargo, la participación en esa jerarquía de adopción de decisiones sigue siendo voluntaria para todo desarrollador.

El sistema por el que se rige el proyecto de FOSS de Apache es un ejemplo. Iniciado con sólo ocho personas a principios de 1995, el Grupo Apache creció rápidamente hasta tener varias docenas de desarrolladores principales que

trabajaban en una asociación sin ataduras con centenares de otras personas que ocasionalmente aportaban ideas, códigos y documentación al proyecto. Al comienzo las decisiones se adoptaban mediante un consenso informal obtenido por correo electrónico. Ese sistema comenzó a fallar al aumentar el número de participantes, y dispararse la participación: los desarrolladores podían dedicarse a otra cosa durante toda una semana antes de volver a trabajar con Apache; pero, la marcha del proyecto en su conjunto no podía esperar a que las múltiples actividades de todos coincidieran.

La respuesta en la práctica fue un sistema de votación por correo electrónico basado en un consenso con un quórum mínimo⁴². En 1999 el Grupo Apache se constituyó oficialmente como entidad sin fines de lucro con el nombre de Fundación Apache de Software⁴³. Ahora sirve de cobertura orgánica y de gestión para toda una gama de proyectos de fuente abierta en Internet (incluidos el servidor Apache original, así como Jakarta, Perl, TCL y otros).

GNU/Linux, al ir creciendo, estableció una organización semiformal para decidir sobre el código. En esa comunidad existen estructuras con

papeles claramente diferenciados. Cuando el programa y la comunidad de desarrolladores crecieron, Torvalds delegó la responsabilidad de los subsistemas y componentes en otros desarrolladores, a los que se conoció como *tenientes*. Algunos de ellos delegaron a su vez la responsabilidad en propietarios de "área" cuyo trabajo tiene un ámbito más estrecho. El resultado orgánico se asemeja y funciona de modo muy parecido a una estructura jerárquica donde las decisiones discurren a lo largo de una pirámide bastante bien definida. La pirámide GNU/Linux funciona imperfectamente, pero está evolucionando, por tanteo, hacia unas dimensiones mejores.

G. Los FOSS y el desarrollo

La era digital ofrece claras oportunidades y riesgos reales a los países en desarrollo. Uno de los riesgos es quedar al margen de las tendencias del software que impulsan a una economía mundial cada vez más digital. La combinación de incrementos rápidos en la capacidad de procesamiento del hardware a precios que disminuyen y las externalidades positivas de Internet, a cuyo tenor el valor de la Web aumenta de manera desproporcionada cuando crece, sugieren que los mercados pueden agrandarse de modo intensivo y extraordinario *dentro* del mundo desarrollado, sin necesidad de tener que expandirse geográficamente hacia los países en desarrollo.

Conforme las economías desarrolladas crean de forma creciente sistemas de producción y compra mediante redes que dependen de infraestructuras avanzadas de TIC, los países que no están conectados en condiciones favorables, y las empresas de esos países, pueden verse muy desaventajados. Los organismos internacionales y las organizaciones no gubernamentales también están cada vez más informatizados y pueden interactuar mejor con los países e instituciones del mundo en desarrollo que están igualmente capacitados en TIC.

Esto implica que las decisiones que toman los gobiernos sobre compras, entornos estandarizados, adopción de TIC, inversiones en tecnología y formación son fundamentales. En los últimos cinco años gobiernos de todo el mundo han empezado a pensar en la legislación que requerirá el uso de los FOSS cuando ofrezcan una alternativa viable al software patentado. Este fenómeno ha sido especialmente pronunciado en el mundo en

desarrollo, donde los países, enfrentados con unos presupuestos de TI limitados, miran con esperanza las soluciones de FOSS. Además, los partidarios de estos últimos han defendido sus ventajas en relación con la creciente preocupación sobre la seguridad de las redes y al facilitar una rendición de cuentas y una transparencia en los datos públicos. Si hubiera alguna duda en cuanto a la funcionalidad de los formatos de datos o del software de procesamiento para actividades fundamentales del gobierno tales como la fiscalidad o las votaciones, siempre cabrá solicitar de expertos independientes que, sin restricciones, inspeccionen el código abierto y los formatos de datos. Los gobiernos también han considerado la posible contribución del despliegue de FOSS a una incipiente producción local de software y al fomento de la capacidad de los recursos humanos en TIC, así como sus efectos potenciales de difusión hacia otros sectores de la economía.

Los sectores públicos de los países en desarrollo han empezado a usar FOSS y a alentarlos en el sector privado por varios motivos fáciles de comprender. Esos motivos pueden agruparse en términos generales en tres categorías: un deseo de independencia, un afán de disfrutar de seguridad y autonomía, y una nueva aplicación de los derechos de propiedad intelectual. En esta sección se examina cada uno de esos factores o motivos por separado.

1. Hacia la sostenibilidad de las TIC

Los partidarios de los FOSS han señalado la dependencia tecnológica creada por el hecho de tener que recurrir a unos pocos grandes proveedores de software ubicados en otros países. El debate sobre ese particular se vio muy acelerado cuando el congresista peruano Edgar Villanueva Núñez, junto con el congresista Jacques Rodrich Ackerman, redactaron un proyecto de ley sobre "Software libre en la administración pública" fechado el 9 de abril de 2002. El proyecto de ley N° 1609, como se llamó, permitiría, si se aprobara, requerir a todos los organismos públicos que usaran exclusivamente software FOSS en sus sistemas y equipos informáticos. El caso del Perú se examina más adelante bajo el epígrafe sobre "Seguridad y autonomía" de esta sección y en el recuadro 4.6. Un número apreciable de gobiernos de países en desarrollo han empezado a explorar las posibilidades de los FOSS. En Sudáfrica, el Consejo de Funcionarios de Información ha citado los costos reducidos, la menor dependencia tecnológica, la

promoción del acceso universal a las TIC, la eliminación de las presiones de los vendedores de software patentados y la adecuación a lenguas y culturas locales como los beneficios principales que ofrece la adopción de software de fuente abierta en el marco de la estrategia de administración pública electrónica⁴⁴. En la India, el Departamento de Tecnología de la Información del Ministerio de Comunicación está fomentando el uso de GNU/Linux y de software de fuente abierta en las instituciones académicas, mientras el Estado de

Bengala Occidental está revisando su programa de FOSS⁴⁵. China está también examinando la cuestión y ha proporcionado apoyo estratégico a Red Flag, un distribuidor local de Linux⁴⁶. En el Estado brasileño de Pernambuco se aprobó en marzo de 2000 la primera ley en todo el mundo sobre el uso de software de fuente abierta⁴⁷. En la encuesta que se presenta en la sección H del presente capítulo figura una amplia lista de las medidas e iniciativas adoptadas en relación con los FOSS.

Recuadro 4.5

Los procedimientos de fuente abierta fuera del sector del software

Dos áreas notables donde está irrumpiendo el principio de la fuente abierta y libre son el mundo de la edición y la biología, en particular en la labor de investigación sobre el genoma humano.

Las editoriales que usan fuentes abiertas se suelen denominar editoriales de contenido abierto. Por este concepto se entiende el proceso de producción de contenido, juntamente con el contenido mismo, cuando se distribuye con arreglo a una licencia de contenido abierto. Ello quiere decir que el contenido está libremente disponible para su modificación, uso y redistribución, con ciertas restricciones encaminadas a evitar la amenaza de un cierre con derechos de autor (Keats 2003). Son varios los directorios y proyectos de contenido abierto que han surgido⁴⁸, inspirados en parte en la insatisfacción de los docentes por el costo creciente y la calidad decreciente de las nuevas ediciones de libros de texto⁴⁹. En el contexto del desarrollo, habida cuenta del costo del contenido así como de la financiación insuficiente de las escuelas y de la falta de expertos en muchos países, la elaboración en colaboración en un entorno y con procedimientos abiertos podría mejorar el acceso a contenidos de gran calidad y de interés en el plano local. El contenido abierto tiene muchas posibilidades de contribuir a un conocimiento común que influya positivamente en el desarrollo económico. Los gobiernos y el sistema de las Naciones Unidas podrían contribuir a un acervo mundial de conocimientos reutilizables si declarasen de contenido abierto muchos de sus documentos, publicaciones y demás textos producidos con las contribuciones de los países miembros o con fondos públicos.

El programa de FOSS que permitió al Proyecto público del genoma humano en el Instituto Sanger reunir el genoma, paralelamente al esfuerzo comercial de Celera, garantizó que los datos del genoma humano no dejaron de ser de dominio público⁵⁰. Jim Kent escribió el programa para evitar que los datos del genoma quedasen aprisionados en patentes comerciales. Esta situación demostró la necesidad de no limitarse a los códigos fuente abiertos; en la comunidad científica se tiene conciencia de la importancia de los datos y procedimientos abiertos, ya que la repetibilidad es la única garantía de validez científica⁵¹. Se ha afirmado, sin embargo, que sin un competidor público de fuente abierta, el genoma humano podría quedar en el dominio de los derechos de propiedad intelectual, a disposición sólo de quienes puedan pagar una suscripción a lo que muchos consideran un conocimiento común de la humanidad.

Otras organizaciones han imitado también el modelo del FOSS. Bioinformatics.org afirma en una declaración que se propone "promover la libertad y la apertura en la esfera de la bioinformática y espera reducir las barreras a la entrada y la participación en ese campo, ya que el acceso a recursos de vanguardia puede resultar prohibitivo para quienes trabajan individualmente, en grupos pequeños, en instituciones con pocos fondos o en países en desarrollo"⁵². En otro ejemplo, Alliance for Cellular Signaling construirá una célula virtual que permitirá a los científicos realizar experimentos completos en sus computadoras. Imitando el procedimiento FOSS, varios laboratorios actuarán como coordinadores centrales, y se espera que centenares de investigadores contribuyan por medio de Internet⁵³.

Los países están interesados no sólo en los posibles ahorros de costos a largo plazo de las soluciones de FOSS, sino también en saber con precisión dónde van a parar los gastos en TI. Los

gobiernos deberían minimizar su dependencia de un proveedor único. Los FOSS también ayudan a evitar una relación que resulta a la larga financieramente desventajosa con determinados

vendedores o productores de software patentados. Aunque todavía no se ha dicho la última palabra en el debate sobre los costos, el uso de software libre significa que la instalación, la formación, el apoyo y el mantenimiento pueden contratarse de manera flexible con toda una gama de proveedores locales que compiten en calidad y precio. Con el empleo de FOSS, más talento nacional puede participar en el desarrollo de software local. Esto permite que el gasto en TI, así como los expertos y los jóvenes talentos, se queden en el país y contribuyan a una incipiente producción local de software. Al mismo tiempo, se mejoran los recursos humanos y la base de conocimientos tecnológicos del país.

Los FOSS eliminan las pérdidas económicas en el plano nacional como consecuencia de la duplicación del trabajo, en particular si el desarrollo de esos programas se hace en una institución pública o académica. El intercambio de aplicaciones y sus códigos fuente entre ministerios, oficinas de la administración y escuelas y universidades puede convertirse en la política pública. Diversos efectos positivos de difusión a otros sectores tecnológicos y no tecnológicos son también posibles y se examinan en el recuadro 4.5.

Por último, promover los FOSS puede tener un efecto antimonopolista en el mercado y los sectores de TI de un país. Las externalidades de la Web en la industria del software, por las que el valor de un producto como un procesador de textos o un sistema operativo aumenta con el número de personas que lo usan, pueden traducirse en monopolios con productos de calidad inferior. La prevalencia de una determinada aplicación de software se ve como una cualidad dominante en sí misma, y esto puede motivar a los desarrolladores a adaptar programas nuevos o actualizaciones específicamente a ese software, con independencia de sus cualidades técnicas básicas. Los FOSS permiten a cualquiera ofrecer servicios de TI, reduciendo así las barreras a la entrada. Si bien un determinado programa de fuente abierta puede llegar a dominar su nicho de mercado, ninguna institución o empresa en particular puede usarlo para hacerse con una posición de monopolio en el mercado.

Cabe esperar ciertas reservas, sobre todo entre los usuarios más acostumbrados, si un gobierno decide prescindir de las soluciones patentadas existentes. No obstante, la facilidad de uso, derivada del hecho de estar familiarizados con esas soluciones, parecerá menos ventajosa a esos usuarios cuando tengan que comprarse licencias

nuevas para actualizaciones que a menudo, a su vez, requerirán cambios en el hardware.

2. Seguridad y autonomía

La seguridad de los datos públicos preocupa mucho a los gobiernos, sobre todo a raíz de los ataques recientes en todo el mundo con virus informáticos y de los temores crecientes al ciberterrorismo y el cibercrimen, así como a los programas espía (*spyware*)⁵⁴. Como mínimo, introducir diversidad en la base del código funcional del software reduce la posibilidad de fallos catastróficos causados por virus que atacan un monocultivo de software. Además, como los gobiernos no pueden elegir a sus clientes o ciudadanos, no deberían obligarlos a usar un software patentado costoso y formatos de datos cerrados.

La necesidad de formatos de datos públicos abiertos guarda relación directa con las exigencias de una mayor rendición de cuentas y transparencia en el sector público. Como ya se dijo, el congresista peruano Edgar Villanueva presentó un proyecto de ley para hacer obligatorio el uso de FOSS en las administraciones públicas. En un canje de cartas con Microsoft Perú⁵⁵, insistió en que, para garantizar el libre acceso de los ciudadanos a información pública, es indispensable que la codificación y el procesamiento de datos no dependan de un solo proveedor. La utilización de formatos abiertos y estandarizados garantiza el libre acceso. Y si se quiere asegurar la perennidad de los datos públicos, la facilidad de uso y el mantenimiento del software no deberían depender de la buena voluntad de los proveedores o de las condiciones impuestas por ellos en un mercado monopolista. En un nivel fundamental, los países, para garantizar la seguridad nacional, han de poder basarse en sistemas sin elementos controlados a distancia. En el recuadro 4.6 se presenta un resumen de las posturas adoptadas en respuesta a los argumentos de Microsoft sobre los inconvenientes de legislar contra el uso de software patentados en las instituciones públicas del Perú.

La necesidad de contar con normas públicas y abiertas para las aplicaciones de software y los ficheros de datos que manejan información pública está hoy universalmente aceptada. El software que se usa en los registros públicos, en la tributación o, en el futuro, en las votaciones puede tener que seguir normas de FOSS. Además, los gobiernos necesitan mantener ciertos datos públicos

fundamentales y rendir cuentas de su elaboración. Con un software patentado y unos formatos de ficheros de datos de fuente cerrada, si el vendedor decidiera no seguir prestando su apoyo por razones técnicas (por ejemplo, porque mantener la retrocompatibilidad supone recargar el código fuente de versiones actuales y nuevas) o financieras (verbigracia por una merma de los ingresos o una bancarrota), las administraciones públicas podrían verse obligadas a cambiar el hardware o el software (o ambos) o a pasar a otro sistema, con los consiguientes costos.

Un estudio sobre el uso de FOSS por los gobiernos de Europa (Berlecon/III 2002) expresa muchas de las preocupaciones manifestadas por el congresista Villanueva. En él se aduce que los FOSS, por sus características, cumplen mejor las funciones de las administraciones públicas, como la de satisfacer el derecho de las personas a tener acceso a cierta información y a conocer cómo esa información se procesa, y la de mantener la seguridad y perennidad de los datos públicos.

Otros países en desarrollo han expresado también su insatisfacción con el modelo de desarrollo y comercialización del software patentado, señalando en particular la influencia prácticamente nula que, como clientes "menores", ejercen sobre el desarrollo de ese software. Se espera que los FOSS brinden más flexibilidad y permitan aportaciones más autónomas a su elaboración. Esto puede concebirse como una cuestión de identificación con el proceso: los países en desarrollo quieren tener la oportunidad de manifestar sus necesidades de software y de participar en el trabajo de innovación como usuarios finales de los programas. Además, ven con buenos ojos la posibilidad de que un sector nacional pueda participar tanto en la identificación como en la satisfacción de las necesidades de software.

3. Derechos de propiedad intelectual (DPI)

Al hacerse mayor hincapié en la aplicación de los DPI en el plano internacional, las posibilidades para los usuarios de software están volviéndose más claras. Conforme los países abandonan las opciones

intermediarias del software pirateado y empiezan a cumplir más estrictamente las normas de propiedad intelectual, se ven obligados a elegir un camino. Si bien el software patentado para computadoras de escritorio se sigue considerando en buena medida más fácil de usar que los productos alternativos, su penetración en el mercado y su precio no están correlacionados en los países donde abunda el software pirateado. Así, todos los esfuerzos de los productores internacionales de software patentados para reducir la piratería en realidad mejoran las condiciones fundamentales para una mayor adopción de software de fuente abierta.

También hay que tener presente que, históricamente, la condición básica que condujo a la aparición, como concepto, de los DPI y de la legislación correspondiente con respecto a los bienes y servicios creativos fue el elevado costo de reproducción de los soportes (libros impresos, discos de vinilo, filmes y medios digitales magnéticos y ópticos), no la capacidad de los Estados y gobiernos para aplicar la legislación. Con la tecnología de hoy esta condición ha pasado a la historia, y los gobiernos se ven ahora enfrentados a tener que actuar con arreglo a sus propias leyes, lo que no parecía ser una consideración práctica hasta hace unos pocos años.

En el polo opuesto, creer que los FOSS presentan una alternativa al respeto de los DPI es una gran equivocación. En realidad, los FOSS piden a los usuarios que, sin excepción, respeten la propiedad intelectual del autor o los autores del software, tal como se dice en las licencias correspondientes LPG u OSD, y se requieren que los gobiernos brinden protección legal y medidas correctoras cuando sea necesario y se justifique. En los anexos 1 y 2 de este capítulo figura el texto completo de la Licencia Pública General GNU y los criterios para las licencias OSD.

Se plantea así para los gobiernos una cuestión más amplia que la simple tolerancia (o intolerancia) de un cierto grado de pirateo de software. De lo que se trata es de saber qué régimen de propiedad y distribución de los instrumentos de TI favorece más a los países en desarrollo y a la economía mundial en su conjunto. Pensar en los FOSS simplemente como una

Recuadro 4.6

Resumen de los principales puntos de la carta de E. Villanueva a Microsoft Perú

El proyecto de ley N° 1609 (Software libre en la administración pública), presentado por el congresista Edgar Villanueva, se propone exigir que se usen FOSS en todos los sistemas públicos, cuando se pueda elegir entre FOSS y software patentados.

En la carta del congresista Villanueva a Microsoft Perú (8 de abril de 2002) se expresaban los principios siguientes:

- Para garantizar el libre acceso de los ciudadanos a la información pública, resulta indispensable que la codificación de los datos no esté ligada a un único proveedor. El uso de formatos estándar y abiertos permite garantizar este libre acceso.
- Para garantizar la perennidad de los datos públicos, es indispensable que la utilización y el mantenimiento del software no dependan de la buena voluntad de los proveedores, ni de las condiciones monopólicas impuestas por éstos.
- Para garantizar la seguridad nacional, el Estado debe poder contar con sistemas desprovistos de elementos que permitan el control a distancia. Los sistemas con código fuente abierto permiten al Estado y a los ciudadanos inspeccionar por sí mismos el código y comprobar si hay "puertas traseras" y espionaje.

En respuesta a las preocupaciones planteadas por Microsoft Perú, el congresista Villanueva aduce lo siguiente:

- El proyecto no se inmiscuye en las transacciones del sector privado y protege la igualdad ante la ley (es decir, a nadie se le negará el derecho de ofrecer estos bienes al Estado). No hay discriminación alguna, pues el proyecto sólo establece cómo han de proveerse los bienes, no quién ha de proveerlos. Las compañías de software patentados son libres de ofrecer soluciones FOSS al gobierno en una licitación.
- El proyecto de ley estimula la competencia, pues alienta a generar una oferta de software con mejores condiciones de utilización y a optimizar el trabajo ya existente, en un proceso de mejora constante.
- El software patentado crea mayoritariamente "tareas técnicas de poco valor agregado" en países como el Perú; con el software libre y abierto se crea empleo técnicamente más calificado, se estimula el mercado y se incrementa el patrimonio común de conocimientos, abriendo alternativas para generar servicios que redunden en beneficio de productores, prestadores de servicios y consumidores.
- En cuanto a la seguridad, los errores en el software libre son menos frecuentes y se reparan mucho más rápidamente que en el software patentado.
- El software libre no implica en modo alguno desconocer las leyes sobre la propiedad intelectual; la amplísima mayoría del software libre está amparado por derechos de autor.
- El proyecto de ley no se equivoca al hablar de los costos del software libre: si bien se mencionan las posibilidades de ahorro en términos de lo pagado por las licencias del software patentado, los fundamentos del proyecto hacen clara mención de las garantías fundamentales que se pretende preservar (acceso libre, perennidad y seguridad) y de estímulos al desarrollo tecnológico local.
- El empleo de software libre contribuye significativamente a disminuir los restantes costos del ciclo de vida: las labores de apoyo y mantenimiento pueden contratarse libremente con toda una gama de proveedores que compiten en calidad y costos de instalación, facilidades, respaldo y mantenimiento; los trabajos de mantenimiento pueden repetirse fácilmente sin incurrir en grandes costos, ya que cabe incorporar las modificaciones al patrimonio común de conocimientos; y el enorme costo causado por la inoperatividad se reduce al emplear un software más estable, que es una de las virtudes del software libre.
- La migración a sistemas nuevos es en realidad más barata cuando se usan FOSS, ya que todos los datos se almacenan en un formato abierto.
- La interoperabilidad queda garantizada tanto por el empleo de formatos estándar (exigido en el proyecto de ley) como por la posibilidad de construir software interoperable a partir de la disponibilidad del código fuente.

alternativa menos costosa al software patentado soslaya un aspecto importante de lo que los FOSS permiten hacer. En un entorno de FOSS, el grado en el que un instrumento informático puede utilizarse y ampliarse viene limitado sólo por el conocimiento, el aprendizaje y la energía innovadora de los usuarios potenciales y no por derechos excluyentes de propiedad, los precios o el poder de países y empresas.

El debate actual a menudo enfrenta las licencias en propiedad con la LPG. Los productores de software comercial aducen que promover la LPG significa que todo producto que se desarrolle en el futuro quedará al margen de una posible comercialización. Como se indicó en la sección anterior, el grueso de los ingresos del software procede de las prestaciones adoptadas al cliente, los servicios posventa o el hardware, o de todo ello junto en forma de soluciones. De hecho, IBM *ganó* efectivamente 1.000 millones de dólares gracias a la LPG GNU/Linux. Por último, las licencias en propiedad permiten únicamente al titular comercializar la propiedad intelectual de que se trate, haciéndola inaccesible a los demás. Redistribuir una versión derivada de un programa patentado no está permitido según los términos de la licencia. Así, el resultado formal no es tan diferente del de la LPG (Lessig 2002). Desde el punto de vista de la estrategia de las TIC y su relación con la innovación y el desarrollo, hay razones para pensar que el modelo de los derechos de propiedad puede fomentar unos derechos de autor excesivos y una acumulación de patentes, con la consecuencia final de reducir la inversión en actividades de investigación y desarrollo y producir un declive de la innovación cuando los fondos para esas actividades se destinan a la adquisición de patentes y al pago de regalías (Bessen 2002, Bessen y Hunt 2003).

Los FOSS suponen una gran oportunidad para el desarrollo en razón del papel fundamental que pueden desempeñar los usuarios en la determinación de productos nuevos y en la trayectoria general de la evolución tecnológica. Las innovaciones en materia de software podrían y deberían proceder cada vez más de los países en desarrollo. Éstos están ahora implícitamente bloqueados al depender de la innovación que viene ya lista para su uso desde el mundo desarrollado. En un entorno de FOSS, los propios usuarios principales pueden impulsar el desarrollo de la tecnología hacia aplicaciones que se ajusten específicamente a las necesidades y demanda locales. No obstante, para que esta demanda se

exprese, los usuarios han de comprender las posibilidades que se les ofrecen y de qué manera una infraestructura digital puede contribuir a mejorar su vida.

Con una evolución tan rápida de esas posibilidades como la que está ocurriendo hoy, parece cierto que los consumidores de TI generan demanda principalmente mediante un proceso de aprendizaje práctico⁵⁶. Al crecer la familiaridad, los usuarios pueden llegar gradualmente a comprender lo que las tecnologías pueden hacer por ellos, e imaginar luego posibilidades nuevas, siempre que conozcan bien sus opciones entre todas las que existen en materia tecnológica. Así pues, un software que se use normalmente mediante licencias restrictivas de la propiedad intelectual puede no ser el medio más propicio para desarrollar las TIC y salvar la brecha digital. Las ventajas que reporta el acceso libre a un código fuente no son simplemente una cuestión de precio sino que representan más bien un requisito económico necesario para que la demanda evolucione. Las aplicaciones que encuentran una aceptación general e impulsan la tecnología y el despliegue de infraestructura en las economías en desarrollo pueden ciertamente proceder de esos mismos países.

H. Opciones de política respecto de los FOSS

Hay dos opciones generales de política que los gobiernos han de considerar, cada una de ellas con dinámicas diferentes para el sector público, la sociedad civil y el sector privado. Esas vías posibles presentan limitaciones u obstáculos que los países en desarrollo en particular han de tener presentes al examinar las diversas opciones de que disponen para adoptar los FOSS.

- **Enfoques formales o informales:** Las ventajas de un enfoque formal, como una legislación o un plan estratégico del gobierno, pueden sopesarse comparándolas con las de enfoques más informales y flexibles que dejen que la utilización de los FOSS evolucione sin un amparo normativo.
- **Estrategia y nivel de participación:** Las iniciativas estratégicas pueden tomarse en el plano subnacional, nacional o regional, y también entrañar diferentes grados de participación, desde fomentar una toma de conciencia hasta una política de compras o de

financiación de la labor de investigación y desarrollo.

Esas opciones no se excluyen entre sí sino que representan más bien espectros en los que los gobiernos pueden enmarcar medidas específicas o un enfoque más general del uso de los FOSS. La relación entre gobierno, sociedad civil y empresas puede también variar, con iniciativas procedentes de una mezcla de los intereses en juego. No existen prescripciones ni esquemas comprobados: los responsables de las políticas tendrán que considerar las circunstancias nacionales y las prioridades del desarrollo de las TIC. En este capítulo se examinan varias opciones de ejemplos de aplicaciones en todo el mundo.

1. Participación formal

Varios gobiernos han aplicado enfoques formales respecto de la adopción de FOSS en el sector público, estudiando leyes que obliguen a usar soluciones de fuente abierta en ese sector, o al menos a considerarlas seriamente como una alternativa al software patentado. En el mundo industrializado, esta tendencia se ha dejado sentir con más fuerza en Europa y particularmente en Alemania y Francia. En el Parlamento francés se presentó un proyecto de ley que se refería tanto al uso de normas abiertas como a la disponibilidad de códigos fuente para el software utilizado en la administración pública. Un proyecto de ley que se está estudiando en Italia obliga a que se prefieran los FOSS en todas las oficinas públicas, y una ley española requiere que los gobiernos autonómicos prefieran y promuevan productos de fuente abierta. En abril de 2002, el Gobierno de la Comunidad Autónoma de Extremadura en España estableció un plan para cambiar todos los sistemas informáticos en las oficinas públicas, las empresas y los hogares a Linux y aplicaciones de FOSS⁵⁷. El Gobierno del Reino Unido ha establecido la política de considerar las soluciones de fuente abierta juntamente con las patentadas en las compras de TI; utilizar productos que apoyen las normas y especificaciones abiertas en todo desarrollo futuro de esa tecnología; considerar la posibilidad de obtener plenos derechos sobre los códigos a medida en el software patentado que se compra; y explorar más las posibilidades de utilizar FOSS como la vía de explotación por defecto del software para la labor de investigación y desarrollo financiada con fondos públicos⁵⁸.

Varios gobiernos latinoamericanos han presentado o aprobado leyes en el plano nacional y local sobre el uso de soluciones de FOSS en el sector público. El caso del Perú ya se examinó antes. El Parlamento argentino examinó una propuesta que obliga, con contadas excepciones, a emplear FOSS en todas las oficinas y empresas públicas, pero dicho Parlamento se hundió con la crisis fiscal de 2001 antes de que se tomara una decisión al respecto. En el Brasil, cuatro ciudades -Amparo, Recife, Ribeirao Pires y Solonopole- han aprobado leyes que dan preferencia o exigen el uso de FOSS, y otros municipios y estados, así como el Gobierno nacional, han estudiado una legislación similar.

Otros países han dado pasos algo menos formales hacia el empleo de FOSS en las administraciones públicas. Francia, además de examinar una posible legislación, ha creado una Agencia de Tecnologías de la Información y las Comunicaciones en la Administración (ATICA), que persigue, entre otras cosas, fomentar el uso de software libre y de normas abiertas.

Un planteamiento menos formal y más flexible tiene sus ventajas. La principal es dejar que el fenómeno del FOSS se desarrolle por sí mismo, juntamente con las innovaciones que se pueden producir en la organización. Diferentes comunidades de usuarios tienen la oportunidad, mediante el procedimiento de fuente abierta, de aplicar modelos tecnológicos y organizativos únicos y contextualmente apropiados, consiguiendo al mismo tiempo un compromiso y una identificación con estas iniciativas a nivel nacional.

A menudo se aduce que los gobiernos no tienen un historial envidiable en lo que respecta a legislar y promover políticas en el sector industrial y que mejor sería no intervenir en el desarrollo del software (Evans 2002). Si bien esta cuestión puede tener algo de cierto en los países con economías de mercado avanzadas, en el contexto del desarrollo cabe preguntarse si los conceptos puristas y no intervencionistas del consenso de Washington son sólo para la exportación. Otra cosa es decidir, en el caso de que un gobierno decida adoptar una legislación favorable a los FOSS, cómo tendría que hacerse y cuán formal debería ser el proceso desde el punto de vista normativo.

2. Estrategia y participación directa

Ya que los gobiernos son consumidores importantes de TIC en los países en desarrollo, su participación resulta crucial para el éxito de cualquier iniciativa de fuente abierta. Un gobierno puede adoptar una política estratégica, o bien fomentar una toma de conciencia y promover elecciones deliberadas e informadas en su administración, así como en el sector privado y la sociedad civil. Puede actuar de intermediario y puede financiar directamente actividades de investigación y desarrollo. En esta sección se examinan los diferentes niveles en que los gobiernos pueden aplicar una estrategia de FOSS.

Un buen ejemplo de pensamiento estratégico de alto nivel es el caso del Gobierno de Sudáfrica. A principios de 2003 se convocó un consejo para

examinar el uso de FOSS. Ese órgano presentó una recomendación oficial que promovía el uso de aplicaciones de fuente abierta cuando las alternativas patentadas no ofrecieran claras ventajas. Las recomendaciones se formularon en el plano estratégico y se describen en el recuadro 4.7. La ventaja de un planteamiento estratégico estriba en la índole del suministro de software. Al tratarse de un producto que es fruto de conocimientos complejos, el software requiere una infraestructura tecnológica y social que facilite su suministro. Un enfoque estratégico permitiría a los gobiernos trabajar en colaboración con donantes para explorar las áreas potenciales de la asistencia al desarrollo, identificando en particular las posibilidades de fomento de la capacidad en materia de recursos humanos, así como las necesidades de asistencia técnica.

Recuadro 4.7

Resumen de las medidas estratégicas recomendadas por un consejo gubernamental de Sudáfrica sobre la política de fuentes abiertas

En la estrategia sobre los FOSS propuesta por el Consejo de Funcionarios de la Información del Gobierno de Sudáfrica figuran medidas encaminadas a consolidar y ampliar la capacidad para aplicar y apoyar soluciones de FOSS, entre ellas las siguientes:

- Suministro de información a los responsables de las decisiones fundamentales (teniendo en cuenta la necesidad de demostrar convincentemente las medidas de seguridad y los principios económicos de los FOSS).
- Asesoramiento de expertos sobre la idoneidad de las soluciones de FOSS.
- Solución de problemas en las aplicaciones de FOSS nuevas.
- Asistencia para el desarrollo del software.
- Formación de desarrolladores y usuarios de FOSS (concentrándose en las instituciones docentes existentes).
- Elaboración de un programa de investigaciones que permita optimizar la comprensión y la adopción de decisiones en materia de FOSS (basado en el carácter de red del modelo de desarrollo de los FOSS).
- Creación de estructuras de apoyo a los FOSS (lo que requerirá cierto desarrollo institucional).

Fuente: Open Source Software in Government, www.oss.gov.za.

El Informe recomienda que se creen fuertes vínculos con instituciones de enseñanza superior para implantar una red nacional de colaboración que pueda ampliarse al plano internacional. También se propugna el establecimiento de asociaciones dentro de los sectores público y privado y la sociedad civil, así como regionalmente dentro de África y a nivel mundial. La estrategia destaca la importancia de lograr el

apoyo de los principales interesados, como los políticos, directores de empresas, profesionales de la TI y administraciones públicas.

Siempre en el plano estratégico, pero apuntando a la colaboración internacional, los FOSS pueden tener el potencial de generar grandes economías de escala y efectos indirectos positivos en el fomento de la capacidad regional y el

desarrollo de infraestructuras. Algunas regiones han comenzado a colaborar en materia de FOSS y esa cooperación ha sido más pronunciada en África. A principios de 2003, países africanos de todas partes del continente lanzaron la Fundación para el software libre y de código fuente abierto para África (Free and Open Source Software Foundation for Africa - FOSSFA), una organización dirigida a promover el uso de FOSS a lo largo y ancho del continente⁵⁹. En el recuadro 4.8 se presentan las recomendaciones de la FOSSFA para formular una política sobre los FOSS.

La FOSSFA prevé que este software brindará oportunidades para desarrollar programas locales establecidos por africanos para su uso en África. Las organizaciones regionales como la FOSSFA ven así las posibilidades que ofrecen las fuentes abiertas para el desarrollo en términos amplios. Un aspecto importante de esas estrategias es la insistencia en el aspecto del fomento de la capacidad asociado con la tecnología de fuente abierta. Las organizaciones regionales tienen la posibilidad de trabajar con educadores a gran escala para introducir las fuentes abiertas en las escuelas, donde los jóvenes pueden aprender a usar, mantener y modificar el software. El futuro se ve como una suerte de revolución técnica regional, en la que los gobiernos y el sector privado hacen suyos los FOSS y utilizan en software y conocimientos desarrollados regionalmente.

Por lo que atañe a las medidas prácticas, varios gobiernos han puesto a prueba los FOSS en organismos de servicios públicos a nivel subnacional. En Sudáfrica, por ejemplo, algunas provincias y departamentos nacionales están usando GNU/Linux y otras aplicaciones de FOSS a título experimental, y el Departamento de Salud ha establecido un sistema de información sobre la salud de tipo FOSS en oficinas nacionales y provinciales, que ahora se está también usando en otros países africanos.

Algunos gobiernos europeos han empezado a prestar un fuerte apoyo a la fuente abierta en el plano nacional. Por ejemplo, los Ministerios de Defensa, Cultura y Economía de Francia han adoptado sistemas operativos de fuente abierta. El Instituto Federal de Agricultura y Alimentación de Alemania ha instalado sistemas operativos de fuente abierta en servidores y estaciones de trabajo. En Gran Bretaña, el servicio nacional de salud ha adoptado un principio de fuente abierta⁶⁰.

En algunos países en desarrollo el sector privado ha tomado la iniciativa de cooperar con el gobierno en el desarrollo de software de fuente abierta. En la India, por ejemplo, mientras los organismos públicos han comenzado a explorar el potencial de las aplicaciones FOSS, especialmente en la educación, empresarios privados han desarrollado el Simputer, un dispositivo de mano basado en un FOSS. (En el capítulo 3 figura un examen del Simputer). La colaboración entre los sectores público y privado es esencial para una adopción sistemática y satisfactoria de soluciones de FOSS. El Simputer demuestra que en el sector privado son posibles soluciones innovadoras de esa índole. Aun así, los desarrolladores vieron que necesitarían asistencia del Gobierno para difundir el dispositivo: era preciso que el Gobierno actuara como un consumidor importante a fin de lograr la masa crítica necesaria para popularizar el producto.

Algunos países han fomentado más explícitamente la colaboración entre los sectores público y privado en la producción y adopción de aplicaciones de fuente abierta. En un intento de fomentar el desarrollo continuo del sector local del software, el Gobierno de Alemania ha suscrito con la IBM un contrato que ofrece a la administración pública descuentos en las computadoras IBM con software GNU/Linux preinstalado proporcionado por el distribuidor alemán de este último, SuSe. Singapur, por medio de su Junta de Desarrollo Económico, encargada de ejecutar las estrategias para impulsar la economía del país, está ofreciendo desgravaciones fiscales a las compañías que usen el sistema operativo GNU/Linux en lugar de las alternativas patentadas.

Por último, hay que referirse a la cuestión de la financiación directa de los proyectos y el desarrollo de FOSS. En el estudio que sigue se citan varios ejemplos. Un punto importante es si, en el software producido con fondos públicos, debería haber alguna preferencia por un modelo de licencia específico. Las autoridades deberían estudiar las licencias OSD disponibles así como la LPG, y reflexionar sobre los extremos del debate entre Microsoft Perú y el congresista peruano Villanueva. Si bien a veces es tentador preferir el espíritu abierto de la LPG, hay que señalar que el software del servidor Apache de tanto éxito y el sistema operativo BSD se distribuyen bajo

Recuadro 4.8

Las recomendaciones de la Fundación para el software libre y de código fuente abierto para África

En su Plan de Acción para el período que va de junio de 2003 a junio de 2005, la FOSSFA propone tres planteamientos distintos que podría adoptar el Gobierno al formular su política sobre los FOSS. Se señala que todo país debería buscar la combinación de planteamientos que refleje mejor su política de TIC y su grado de desarrollo. (En el anexo III de este capítulo figura el texto de la declaración de constitución de la FOSSFA.)

1. *El planteamiento neutral*

Los gobiernos pueden adoptar un planteamiento neutral asegurándose de que se apoye la posibilidad de elección y se elimine la discriminación contra los FOSS. A tal fin, los gobiernos deben:

- Adoptar políticas que garanticen que los FOSS se tengan cuidadosamente en cuenta en los procedimientos de adquisición de TI.
- Establecer criterios para evaluar los productos de fuente abierta, y los procedimientos para adoptar y mantener normas abiertas.
- Dejar que el software de fuente abierta compita en pie de igualdad con las alternativas patentadas.
- Iniciar una labor de comunicación para fomentar el conocimiento y la comprensión de los FOSS.

2. *El planteamiento favorable*

En un planteamiento favorable, la política se orienta a crear capacidad para usar los FOSS. Además de lo indicado en el planteamiento neutral, los gobiernos deben:

- Desarrollar la capacidad de proporcionar orientaciones sobre cómo seleccionar y aplicar los FOSS.
- Promover la educación y la formación en el uso de productos de FOSS.
- Apoyar el establecimiento de asociaciones entre usuarios y desarrolladores.

3. *El planteamiento de apoyo resuelto*

En este planteamiento, los gobiernos fomentan activamente el desarrollo de FOSS por medio tanto de leyes como de medidas concretas. Los gobiernos deben:

- Respalda activamente las comunidades y los proyectos de los desarrolladores de FOSS.
- Adoptar estrategias para incrementar el compromiso con los productos de fuente abierta.
- Hacer periódicamente estudios sobre la repercusión de los FOSS en la prestación de servicios públicos.
- Participar en programas que minimicen los riesgos relacionados con los FOSS.
- Estandarizar el uso de FOSS cuando los análisis indiquen que constituyen la mejor alternativa.

Fuente: Plan de Acción de 2003-2005 de la FOSSFA, www.fosssa.org/resources.html.

licencias OSD menos restrictivas que, de hecho, programas siguen siendo los favoritos en sus
 permiten la protección del código fuente con respectivos campos.
 derechos de propiedad. Sin embargo, esos

3. Ejemplos de medidas de política relativas a los FOSS en países en desarrollo

A continuación figuran ejemplos del uso de FOSS en países en desarrollo. Cuando procede, se describe el marco normativo y se señalan las formas principales de participación. El estudio no es exhaustivo y se basa en información encontrada en buscadores de Internet.

Argentina ⁶¹

- Un proyecto de ley sobre política para el uso de software libre en el Estado Federal presentado en la Cámara de Diputados argentina en abril de 2001 requería el uso obligatorio de FOSS en la administración pública. La crisis económica hizo dimitir al Gobierno antes de que se pudiera votar. En marzo de 2002 se presentó un proyecto similar que está en estudio.
- El proyecto de ley actual propone que los FOSS sean un componente de la campaña nacional contra el pirateo de software.

Brasil ⁶²

- Rio Grande do Sul fue la primera administración que aprobó una ley que hacía obligatorio el uso de FOSS tanto en los organismos públicos como en las compañías de agua, gas y electricidad de gestión privada.
- Cuatro ciudades del Brasil han aprobado una legislación que establece la preferencia por el software libre cuando se dispone de una opción de fuente abierta.
- El sistema nacional de salud se propone liberar 10 millones de líneas de código fuente.
- El primer Foro Internacional anual del Software Libre se celebró en el Brasil en mayo de 2000.
- En la provincia de Pernambuco, se aprobó en marzo de 2000 la primera ley del mundo sobre el uso de software de fuente abierta.

China ⁶³

- La oficial Academia China de Ciencias, juntamente con la Shanghai New Margin Venture Capital, de propiedad del

Gobierno, establecieron Red Flag Linux, una distribución de Linux en idioma chino.

- El Ayuntamiento de Beijing creó el Centro de productividad del sector del software, que ha lanzado un proyecto denominado "Yangfan" para mejorar el desempeño de las distribuciones locales de GNU/Linux.
- Es de señalar la gran presencia de desarrolladores internacionales de FOSS, como Turbo Linux, Red Hat e IBM.

Filipinas ⁶⁴

- Bayanihan Linux, desarrollado en el marco del Proyecto de Fuente Abierta del Instituto de Ciencias Avanzadas y Tecnología de Filipinas, ha sacado su segunda versión y está asociada con los editores más recientes de conjuntos para oficina, imagen y textos, con instrumentos para Internet y trabajos en red y aplicaciones multimedia. Bayanihan es una instalación en un CD adaptada a la demanda local.

India ⁶⁵

- La creciente difusión de Linux en la India ha persuadido a Microsoft a compartir el código fuente con un organismo público.
- El Simputer fue desarrollado por un grupo de científicos del Instituto Indio de Ciencias y por Encore Software. (Véase el recuadro 3.3 en el capítulo 3).
- Los organismos oficiales promueven el uso de soluciones locales en idioma hindi. El Centro de Desarrollo del Cálculo Avanzado y el Departamento de Tecnología de la Información están así apoyando una distribución de GNU/Linux en hindi llamada Indix.
- El Departamento de Tecnología de la Información ha expresado su intención de introducir Linux como la norma *de facto* en las instituciones académicas; los establecimientos de investigación desarrollarán y distribuirán los instrumentos oportunos; y se pedirá a los gobiernos central y estatales que utilicen las ofertas basadas en Linux.
- La empresa West Bengal Electronics Industry Development Corp Ltd., el órgano principal de TI de ese Estado, ha formado una célula Linux para apoyar varios

proyectos públicos de TI dentro y fuera del Estado.

- Están celebrándose conversaciones con los principales actores del sector de los FOSS para establecer proyectos conjuntos.

Malasia ⁶⁶

- El Gobierno se comprometió en noviembre de 2001 a usar FOSS en los principales organismos, tales como el Tesoro, y en las esferas como la contratación pública electrónica.
- La Confederación Nacional de Cálculo del país cuenta con un grupo dedicado especialmente a los FOSS.
- El Primer Ministro inauguró el Komnas (Komputer Nasional) Twenty20 Personal Computer, creado por el sector privado haciendo uso de FOSS.
- El Instituto de Sistemas Electrónicos de Malasia, que es el asesor del Gobierno en materia de TIC, está impulsando el cambio hacia los FOSS, incluido un intento de crear un PC de bajo costo basado en GNU/Linux.

Pakistán ⁶⁷

- La Dependencia de Movilización de Recursos Tecnológicos del Gobierno ha creado un grupo especial "Fuerza Linux" que ayudará al país a avanzar en el campo de los FOSS. Entre otras cosas, se financiarían programas de investigación y desarrollo para crear software a medida de los clientes, impartir formación y desarrollar aplicaciones en idioma local.

Perú ⁶⁸

- El congresista Edgar Villanueva ha presentado el proyecto de ley N° 1609, "Software libre en la administración pública", para hacer obligatorio el uso de FOSS en todos los sistemas públicos.
- Su enfrentamiento abierto con Microsoft Perú le ha granjeado a él y al Perú la reputación de ser los innovadores en materia de FOSS en el mundo en desarrollo.

República de Corea ⁶⁹

- La compañía local HancomLinux firmó en enero de 2003 un acuerdo con la Oficina Central de Compras del país para suministrar al Gobierno 120.000 ejemplares de su software Linux de escritorio para la productividad en la oficina, denominado HancomOffice. Se espera que este software de fuente abierta, que es compatible con las aplicaciones Office de Microsoft, como Word y Excel, permita ahorrar dinero público a la larga y estimule la actividad de las empresas locales que compiten con Microsoft en el sector del software.

Sudáfrica ⁷⁰

- Un consejo del Gobierno convocado para estudiar el empleo de FOSS hizo pública una recomendación oficial que promovía el uso de aplicaciones de fuente abierta cuando las alternativas patentadas no ofrecieran claras ventajas, y señalaba las medidas estratégicas necesarias.
- En enero de 2003, el Gobierno declaró que usaría FOSS y estableció un consejo de investigaciones científicas e industriales para desarrollar conocimientos de programación.
- Sudáfrica se ha puesto a la cabeza en la colaboración regional sobre software de frente abierta, con la Fundación para el software libre y de código fuente abierto para África (FOSSFA).

Tailandia ⁷¹

- El grupo de desarrollo de tecnología NECTEC, que cuenta con apoyo oficial, ha elaborado una distribución GNU/Linux para computadoras y servidores de escuelas y de oficinas públicas, a saber, el Linux-SIS (School Internet Server) para servidores y el Linux TLE (Thai Linux Extension) para las computadoras de escritorio de las oficinas públicas. El proyecto se propone acortar las distancias entre el uso de software legal y pirateado, y promover el desarrollo de empresas locales.

Viet Nam⁷²

- Los delegados del Gobierno en un seminario sobre software celebrado en Hanoi llegaron a la conclusión de que el país podría ahorrar centenares de millones de dólares anualmente y garantizar mejor la seguridad de la información si se pasaba a los FOSS.
- Las compañías vietnamitas de TI están trabajando en proyectos FOSS mediante subcontratas con empresas extranjeras.
- Los FOSS se incluyeron en el Programa Nacional de Tecnología de la Información.

I. Conclusiones

Internet, o la rápida introducción en el quehacer humano de la telecomunicación en banda ancha, configurada como una red neutral y pública, cambia algunas cosas muy importantes en las limitaciones y oportunidades que personas, organizaciones y países encuentran al moverse hacia unas economías cada vez más intensivas en conocimientos. Los países en desarrollo tendrán que afrontar al mismo tiempo problemas nuevos y viejos: la promesa de un desarrollo impulsado por la información; el desafío de mantener relaciones complejas y de contenido tecnológico con empresas multinacionales y con el mundo desarrollado; y la cuestión de cómo configurar los regímenes de DPI que, cada vez más, son pilares cruciales del crecimiento económico. Las ventajas para los países en desarrollo de promover una política que ofrezca un entorno positivo para una TI de fuente abierta son múltiples, y las diferencias que pueda haber con el mundo desarrollado son generalmente de grado y no de dirección.

Los FOSS no desaparecerán en el futuro previsible. La experiencia hasta ahora ha mostrado que los entornos de fuente abierta suelen producir un software fiable, seguro y mejorable a un costo relativamente bajo. Por definición, los FOSS ofrecen un enfoque más adecuado en materia de seguridad y cubren mejor la necesidad de normas públicas y abiertas, asunto éste de gran interés para las instituciones públicas. La fuente abierta elimina las pérdidas económicas en el plano nacional que se producen como consecuencia de la duplicación

en el desarrollo de software, en particular si éste se ha hecho en una institución pública o académica. Apoyar los FOSS puede tener un efecto antimonopolista en el mercado y el sector de TI de un país y en el plano mundial, reduciendo con ello la amenaza de un encierro tecnológico y financiero.

Los gobiernos, después de estudiar la experiencia de los países en desarrollo y desarrollados que hayan establecido una política e iniciado actividades en materia de FOSS, deberían decidir qué planteamiento se ajusta mejor a sus necesidades. Si bien algunos países pueden contar con muchos expertos técnicamente cualificados e interesados, esto no ocurre necesariamente en todo el mundo en desarrollo. Así, la política oficial de recursos humanos para el desarrollo de las TIC puede tener que incluir un programa relativo a los FOSS. Si bien el bajo costo no impulsa el desarrollo de los FOSS en el plano mundial, en los países en desarrollo puede muy bien acelerar su adopción, habida cuenta en particular de la aplicación cada vez más estricta de los DPI exigida por los productores de software patentados. El dinero de las licencias puede utilizarse mejor formando a expertos en TIC que sean capaces de desarrollar software y no se limiten a "pulsar el menú" para utilizarlo. Por último, la creciente adopción de FOSS en el mundo desarrollado está creando oportunidades de exportación de software a medida para los incipientes sectores de TI de los países en desarrollo.

Por último, hay muchos modos diferentes de arbitrar la transición a una economía del conocimiento o la información. Pero si la producción, el flujo y el control de la información son rasgos distintivos de una comunidad, una economía y una sociedad, entonces las normas que rigen la información resultan fundamentales. El software es una de las fuentes más importantes de esas normas. Al igual que en cualquier conjunto de normas, importa no sólo lo que las normas dicen sino también cómo han llegado a redactarse y quién puede cambiarlas y en qué condiciones. Los FOSS deberían contemplarse, entonces, como algo más que un tipo diferente de producto. Es un tipo distinto de proceso para establecer, mantener y cambiar las normas por las que se rigen los flujos de información.

NOTAS

1. Trasladar un programa del sistema operativo en que se creó a otro; crear una versión ejecutable en otro sistema.
2. En Tuomi (2002) figura una evaluación crítica de los datos.
3. La competencia, según los datos proporcionados por Hoover's (www.hoovers.com), está integrada por las diez empresas siguientes, ordenadas por ingresos: Siebel, BMC Software, Novell, Network Associates, Activision, Sage Group, Infosys (Bangalore), Business Objects, Legato Systems y RSA Security.
4. Software producido para la venta masiva al por menor y que no se ha adaptado a las necesidades concretas del usuario. Suele tratarse de sistemas operativos, utilidades, aplicaciones y lenguajes de programación.
5. Véase www.businessweek.com/magazine/content/03_02/b3815723.htm.
6. Los ingresos totales de IBM, Sun Microsystems, EMC, Dell, Hewlett-Packard, Gateway, Apple, Fujitsu y NEC son de aproximadamente 276.000 millones de dólares de los EE.UU., pero es difícil determinar qué porcentaje procede de actividades relacionadas con el software.
7. Esta analogía se atribuye a Mitchell Stoltz, de Mozilla.org.
8. Puede incluirse en el conjunto de software o en los CD-ROM, o puede incorporarse a una página web e indicarse su dirección (URL).
9. A menudo se plantea la pregunta de por qué los desarrolladores de software optarían por hacer público el código fuente de manera que no estuviera protegido ni tuviera derechos de autor. Ese debate plantea una serie de cuestiones económicas y sociológicas que se estudian más a fondo en la sección F de este capítulo.
10. El significado implícito del término "fordista" se originó en el estudio de Adam Smith de la división del trabajo. El proceso de fabricación de cualquier producto se puede dividir en etapas, y hacer que cada trabajador se especialice en una de esas etapas permite aumentar notablemente la productividad. La teoría fue perfeccionada en las fábricas de automóviles de Ford, y de ahí el término.
11. Una de las consecuencias más generalizadas y perjudiciales de la gran publicidad que se dio a Internet en los últimos años noventa fue la popularización de un concepto excesivamente simplista de la "autoorganización". Las nuevas tecnologías parecían estar socavando o cuando menos presentando alternativas a las jerarquías tradicionales de mando y control en las empresas, la administración pública y en otros lugares.
12. Se suele reconocer a Ken Thompson como el "inventor" de UNIX y a Dennis Ritchie como el de C. Ambos eran empleados de los Laboratorios Bell.
13. El grupo de la Universidad de California en Berkeley fue particularmente influyente. Bill Joy, que más tarde fundaría Sun Microsystems, dirigió el primer proyecto Berkeley Software Distribution (BSD) de UNIX en 1978.
14. Esta causa permaneció abierta durante 13 años, hasta que la administración Reagan la desestimó en 1981. Véase DeLamarter (1986).
15. A juicio de Stallman, "el intercambio de recetas es tan antiguo como la cocina", pero el software con derechos de autor significaba que "el primer paso para usar una computadora era prometer no ayudar al vecino". Consideró que así se "dividía al público y se dejaba al usuario desamparado" (1999, pág. 54). Se puede obtener el texto completo en www.gnu.org/philosophy/why-free.html.
16. En el recuadro 4.1 del texto figura una descripción de GNU.
17. GNU.org, en www.gnu.org/licenses/gpl.html.
18. Se han efectuado varias modificaciones a estas disposiciones específicas, pero no se ha cambiado el principio general.
19. En la sección "Acknowledgements" de la antología *Open Sources: Voices from the Open Source Revolution* (1999) se describe Emacs con cierta extensión: "Decir que Emacs es un editor es como decir que nuestro planeta es un pedazo de tierra. Emacs es un editor, un navegador de Internet, un servidor de noticias, un programa de correo, un administrador de información personal, un programa de composición, un editor de programación, un editor hexagonal, un procesador de textos y varios videojuegos. Muchos programadores usan un fregadero de cocina como icono para su ejemplar de Emacs. Hay muchos programadores que entran en Emacs y no tienen que hacer nada más en su computadora. Emacs no es sólo un programa, sino una religión, y RMS (Richard M. Stallman) es su profeta."
20. Véase <http://gcc.gnu.org> para más información.
21. Véase <http://sources.redhat.com/gdb/> para más información.
22. Véase <http://opensource.org/osd.html> para más información.

23. "Creemos que los argumentos de interés propio económico para usar un código fuente abierto tienen tanta fuerza que no es necesario emprender ninguna cruzada moral al respecto". Véase www.opensource.org para más información.
24. Gomes L (1998), Microsoft acknowledges growing threat of free software for popular functions, *Wall Street Journal*, 3 de noviembre: B6; y "Halloween Memo", en GNU/Linux.miningco.com/library/blhalloween.html.
25. En www.scripting.com/misc/halloweenMemo.html se puede obtener una versión íntegra no autorizada del llamado "Memo de Halloween". La OSI ha incorporado la versión del memorándum que se filtró al sitio www.opensource.org/halloween/halloween1.php. Según la OSI, el Memo de Halloween de Microsoft decía explícitamente lo siguiente:

"El OSS es creíble a largo plazo... [porque] lo verdaderamente importante de GNU/Linux no es la versión estática del producto, sino el proceso que lo rodea. Ese proceso ofrece credibilidad y cierto aire de seguridad futura para las inversiones de los clientes de GNU/Linux. Se ha implantado GNU/Linux en entornos comerciales de gran importancia con críticas muy positivas... Estudios recientes demuestran claramente que se puede lograr o perfeccionar la calidad comercial mediante proyectos de OSS. Internet representa un escaparate ideal y de gran visibilidad para el mundo del OSS. La capacidad del proceso del OSS para reunir y aprovechar la inteligencia colectiva de miles de personas en Internet es sencillamente asombrosa. Y, lo que es más importante, la predicación del OSS aumenta con el tamaño de Internet mucho más rápido de lo que la predicación [de Microsoft] parece aumentar."

En 2002 y 2003, Microsoft empezó a permitir un acceso limitado a su código fuente a clientes importantes y administraciones públicas que estaban interesados por razones de seguridad. Para ello firmaron contratos especiales para evitar la revelación de la información o la posible competencia.

26. Véase la nota 3.
27. Kuan (2003) comenta un estudio empírico de la existencia de gazapos donde se compara el FOSS con una plataforma patentada.
28. Véase www.computerworld.com.au/index.php?id=2110919358&fp=16&fpid=0.
29. Las cifras de cuota de mercado del software siempre se deben interpretar con cautela, ya que las cuestiones de muestreo y medida complican cualquier evaluación simple para saber quién está usando qué software en estos mercados con un grado de distribución tan elevado. Los datos utilizados se refieren primordialmente a los países industrializados. No se dispone de datos sobre la cuota de mercado en los países en desarrollo.
30. E-soft, www.securityspace.com/s_survey/data/200303/index.html.
31. Los sistemas operativos efectúan tareas básicas, como reconocer datos procedentes del teclado, enviar datos a la pantalla, guardar archivos y directorios en el disco y controlar dispositivos periféricos como unidades de disco e impresoras. La mayoría de los usuarios de computadoras sólo conoce el sistema operativo Microsoft Windows.
32. Véase www.netcraft.com/Survey/index-200106.html#computers; véase también www.oss-institute.org/reference.html.
33. Véase www.businesswire.com/cgi-bin/f_headline.cgi?bw.111301/213170209.
34. Véase www.dwheeler.com/oss_fs_why.html.
35. Berkeley Software Distribution.
36. Esa concesión se deriva de la carta que Microsoft remitió al Diputado del Perú Edgar Villanueva, en la que le reprochaba su deseo de designar oficialmente el FOSS como opción preferente para la administración pública.
37. Véase www.eweek.com/article2/0,3959,840669,00.asp para más información.
38. En concreto, el consumo de un bien indivisible (*non-rival*) por un consumidor no disminuye su utilidad para otro consumidor. El carácter no excluyente supone que es difícil, si no imposible, cobrar dinero a alguien por usar el producto, de la misma manera que es imposible cobrar por respirar o por pasear en un parque. Los bienes públicos son los que cumplen ambos criterios: ser indivisibles y no ser excluyentes.
39. Se pueden obtener los resultados completos en www.psychologie.uni-kiel.de/GNU/Linux-study/. Las tres ventajas más importantes (todas recibieron 4,6 puntos en una escala de 1 (nada importante) a 5 (muy importante)) eran "divertirse programando", "mejorar mis aptitudes de programación" y "facilitar mi trabajo diario gracias a un software mejor". La "falta de remuneración" era mucho menos importante (2,2) y el "tiempo perdido por dedicarme a GNU/Linux" era algo más importante (2,6).
40. La versión 0.3 de la encuesta efectuada a piratas informáticos por el Grupo de Consultoría Boston fue presentada en el GNU/LinuxWorld el 31 de enero de 2002; www.bcg.com/opensource/BCGHACKERSURVEY.pdf. El Grupo

de Consultoría Boston entrevistó a una selección aleatoria de desarrolladores de SourceForge; los resultados se basan en las respuestas de 526 de ellos (un índice de respuesta del 34,2%).

41. Véase Berlecon/III (2002), parte 4.
42. Cualquier desarrollador participante puede expresar su opinión votando sobre cualquier cuestión relacionada con el proyecto, pero los votos de los miembros del Grupo Apache son los únicos vinculantes. Para efectuar cambios en el código es necesario un mínimo de tres votos a favor y ninguno en contra; los vetos deben ir acompañados de una explicación convincente. Para adoptar otras decisiones es necesario un mínimo de tres votos a favor y una mayoría general a favor. La elección de miembros del Grupo Apache se basa en el principio de la meritocracia entre iguales: alguien que haya incorporado muchas mejoras a alguna parte del código puede ser presentado como candidato por un miembro del grupo y entrar a formar parte de éste si los miembros existentes lo eligen por unanimidad. Entrevista con los miembros del Grupo Apache; Fielding (1999).
43. Para más información, véase www.apache.org.
44. Para más información sobre la postura de Sudáfrica, véanse Government IT Officers Council of South Africa (2002), *Using Open Source Software in Government*; y National Advisory Council on Innovation of South Africa (2002), *Open Software and Open Standards in South Africa*.
45. Véase <http://tdil.mit.gov.in/>, donde figura un vínculo a Indix (versión de GNU/Linux en hindi); véanse también www.crn-india.com/features/stories/39090.html y www.zdnetindia.com/techzone/linuxcentre/stories/70365.html.
46. Véase www.redflag-linux.com/.
47. Para más información, véase www.pernambuco.com/tecnologia/arquivo/softlivre1.html.
48. Véase www.wikipedia.org/wiki/Open_content, donde figura una lista de proyectos de contenido abierto y vínculos a éstos.
49. Véase www.lightandmatter.com/article/article.html.
50. Véase www.sanger.ac.uk/HGP/.
51. Véanse www.oreillynet.com/pub/a/network/2002/04/05/kent.html y www.wired.com/news/medtech/0,1286,46154,00.html para más información.
52. Véase <http://bioinformatics.org/>.
53. Véanse www.newamerica.net/index.cfm?pg=article&pubID=901 y www.cellularsignaling.org/.
54. Programas que reúnen en secreto información sobre el usuario de una computadora y la envían a empresas de publicidad y a otros interesados.
55. Se puede obtener el texto completo en www.theregister.co.uk/content/4/25157.html y en www.pimientolinux.com/peru2ms/.
56. Véase Bar F y Borrus M (1998), The path not yet taken: User-driven innovation and U.S. telecommunications policy. Presentado en la Cuarta Conferencia Anual del Consorcio para la Investigación de Políticas y Estrategias en materia de Telecomunicaciones (CRTPS), University of Michigan Business School (Ann Arbor (Michigan, EE.UU.)), 5 a 6 de junio.
57. Para más información véanse el sitio web de Extremadura dedicado al FOSS, www.linex.org o *The Washington Post* (2002), Europe's Microsoft alternative: Region in Spain abandons windows, embraces Linux (3 de noviembre) y *Wired*, Extremadura measures: Linux, en www.wired.com/news/business/0,1367,51994,00.html.
58. Office of the E-Envoy, Open Source Software Use in UK Government, [www.e-envoy.gov.uk/oe/oe.nsf/sections/frameworks-oss-policy/\\$file/oss-policy.htm](http://www.e-envoy.gov.uk/oe/oe.nsf/sections/frameworks-oss-policy/$file/oss-policy.htm).
59. Véase www.fossfa.org.
60. Para más información, véase ZDNet en <http://news.zdnet.co.uk/story/0,,t269-s2121266,00.html>.
61. Véase www.lugcos.org.ar/serv/mirrors/proposicion/proyecto/leyes/#ref.#1.
62. Véanse www.softwarelivre.org/index.php?menu=projeto y www.pernambuco.com/tecnologia/arquivo/softlivre1.html.
63. Véase www.redflag-linux.com/eindex.html y www.bsw.gov.cn.
64. Véase <http://bayanihan.asti.dost.gov.ph/>.

65. Véanse www.zdnetindia.com/techzone/enterprise/stories/74137.html; www.simputer.org/simputer/; <http://rohini.ncst.ernet.in/indix/>; <http://economictimes.indiatimes.com/cms.dll/xml/uncomp/articleshow?artid=24598339>. www.zdnetindia.com/news/national/stories/71697.html; y <http://ebb.antville.org/stories/362705/>.
66. Véanse <http://asia.cnet.com/newstech/systems/0,39001153,39071821,00.htm>; <http://star-techcentral.com/tech/story.asp?file=/2002/9/9/technology/09oss&sec=technology>; www.mncc.com.my/oscc/oscc-main.html; y <http://opensource.mimos.my/>.
67. Véase www.tremu.gov.pk/task/Linux.htm.
68. Véase <http://odfi.org/archives/000004.html#4>.
69. Véase <http://en.hancom.com/index.html>.
70. Véase www.oss.gov.za/.
71. Véase www.nectec.or.th/linux-sis/.
72. Véase www.idg.com.sg/idgwww.nsf/unidlookup/21744381DA98B64148256CA80007772E?OpenDocument.

REFERENCIAS Y BIBLIOGRAFÍA

- Berinato S (1999). Catering to the GNU/Linux Appetite. *PC Week*, 7 de junio: 103.
- Berinato S (2000). GNU/Linux Graduates to Mainframes. *Industry Standard*, 17 de mayo.
- Berlecon Research and the International Institute of Infonomics (III), Universidad de Maastricht (2002). *Free/Libre and Open Source Software: Survey and Study*. <http://www.infonomics.nl/FLOSS>.
- Bessen J (2002). What good is free software? In: Hahn R, ed. (2002). *Government Policy toward Open Source Software*. Washington, DC, AEI-Brookings Joint Center for Regulatory Studies.
- Bessen J y Hunt R (2003). An empirical look at software patents. *Research on Innovation*. www.researchoninnovation.org
- Business Week* (2003). The GNU/Linux uprising. 3 de marzo.
- CNET (2000). IBM to join in GNU/Linux supercomputing effort. 21 de marzo.
- DeLamarter RT (1986). *Big Blue: IBM's Use and Abuse of Power*. Nueva York, Dodd, Mead.
- Evans SD (2002). Politics and programming: Government preferences for promoting open source software. En: Hahn R, ed. (2002). *Government Policy toward Open Source Software*. Washington, DC, AEI-Brookings Joint Center for Regulatory Studies.
- Fielding RT (1999). Shared leadership in the Apache Project. *Communications of the ACM* 42 (2): 42-43.
- Free Software Foundation) (FSF) (1991). *GNU General Public License, v. 2.0*. www.gnu.org/copyleft/gpl.html
- Free Software Foundation (FSF) (1996). The free software definition. www.fsf.org/philosophy/free-sw.html
- Ghosh RA (1998). Cooking pot markets: An economic model for the trade in free goods and services on the Internet. *First Monday* 3 (3).
- Goldhaber MH (1997). The attention economy and the Net. *First Monday* 2 (4).
- Holmström B (1999). Managerial incentive problems: A dynamic perspective. Working Paper 6875. Cambridge, MA, National Bureau of Economic Research.
- Iannacci F (2002). The economics of open-source networks. *Communications & Strategies* 48. International Telecommunications Society.
- Keats D (2003). Collaborative development of open content: A process model to unlock the potential for African universities. *First Monday* 8 (2). www.firstmonday.dk/issues/issue8_2/keats/

- Kuan J (2003). Open source software as lead user's make or buy decision: A study of open and closed source quality. Paper presented at the second conference on "The Economics of the Software and Internet Industries", Toulouse, Francia, 17 y 18 de enero.
- Lancashire D (2001). Code, culture and cash: The fading altruism of open source development. *First Monday* 6 (12).
- Lerner J y Tirole J (2000). The simple economics of open source. Working Paper 7600. Cambridge, MA, National Bureau of Economic Research.
- Lerner J y Tirole J (2001). The open source movement: Key research questions. *European Economic Review* 45.
- Lessig L (2002). Open source baselines: Compared to what? In: Hahn R, ed. (2002). *Government Policy toward Open Source Software*. AEI-Brookings Joint Center for Regulatory Studies, Washington, DC.
- Open Sources: Voices from the Open Source Revolution* (1999). DiBona C, Ockman S and Stone M, eds. O'Reilly & Associates, Sebastopol, CA.
- Pappas Johnson J (2001). Economics of open source software. F/OSS, Massachusetts Institute of Technology. <http://opensource.mit.edu/>
- Raymond ES (1999a). The revenge of the hackers. In: *Open Sources: Voices from the Open Source Revolution*.
- Raymond ES (1999b). The magic cauldron. En <http://www.catb.org/~esr/writings/magic-cauldron/>
- Raymond ES (2000). The cathedral and the bazaar. www.catb.org/~esr/writings/cathedral-bazaar/
- Stallman R (1999). The GNU operating system and the free software movement. En: *Open Sources: Voices from the Open Source Revolution*
- Stallman R (2002). Free as in freedom. Ongoing. www.oreilly.com/openbook/freedom/
- Tuomi I (2002). The lives and death of Moore's Law. *First Monday* 7 (11).
- Weber S (2000). The political economy of open source software. Working Paper 140. Berkeley Round Table on the Information Economy. <http://brie.berkeley.edu/~briewww/research/workingpapers.htm>

Anexo I

El texto que figura a continuación es una traducción del texto en inglés de la Licencia Pública General GNU que aparecía en el sitio de la Fundación para el Software Libre en Internet <http://www.gnu.org/licenses/gpl.txt> el 13 de agosto de 2003.

LICENCIA PÚBLICA GENERAL GNU

Versión 2, junio de 1991

Copyright © 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307, Estados Unidos de América

Está permitida la copia y la distribución de copias fieles de este documento de licencia, pero no su modificación.

Preámbulo

Los contratos de licencia de la mayor parte de los programas informáticos (software) están diseñados para privar al usuario de la libertad de compartir y modificar dichos programas. En cambio, la Licencia Pública General GNU tiene por objeto garantizar la libertad de compartir y modificar el software libre -es decir, asegurar que los programas sean libres para todos sus usuarios. Esta Licencia Pública General se aplica a la mayoría de los programas de la Fundación para el Software Libre y a cualquier otro programa informático cuyos autores así lo establezcan. (Algunos otros programas de la Fundación para el Software Libre están ligados a la Licencia Pública General para Bibliotecas GNU). Usted también puede aplicar esta licencia a sus programas.

Cuando hablamos de software libre, hablamos de libertad, no de gratuidad. Nuestras Licencias Públicas Generales están diseñadas para garantizar que el usuario tenga la libertad de distribuir copias de los programas libres (y de cobrar por ese servicio si así lo desea), reciba el código fuente o tenga la posibilidad de obtenerlo, si lo desea, pueda modificar los programas o utilizar partes de ellos en nuevos programas libres, y sepa que tiene la posibilidad de hacer todas estas cosas.

Para proteger los derechos del usuario, necesitamos imponer restricciones que impidan que se le puedan negar esos derechos o que se le pueda pedir que renuncie a ellos. Estas restricciones se traducen en algunas responsabilidades para el usuario si distribuye copias del programa, o si lo modifica.

Por ejemplo, si usted distribuye copias de un programa, ya sea gratuitamente o mediante pago, debe dar a quienes reciban el programa los mismos derechos de los que usted goza. Debe asegurarse también de que reciban el código fuente, o de que puedan obtenerlo si lo desean. También debe mostrarles los presentes términos para que conozcan los derechos de que gozan.

Nosotros protegemos sus derechos en dos niveles: (1) protegiendo los derechos de autor del programa, y (2) ofreciéndole este contrato de licencia que le otorga permiso legal para copiarlo, distribuirlo o modificarlo.

Además, para nuestra protección y la de los autores de los programas, queremos asegurarnos de que todos entiendan que los programas libres no tienen ninguna garantía. Si alguien modifica el programa y lo

distribuye, queremos que quienes lo reciban sepan que lo que han recibido no es el programa original, de manera que cualquier problema introducido por otros no afecte a la reputación del autor original.

Por último, todo programa libre se encuentra bajo la constante amenaza de las patentes. Deseamos evitar el peligro de que los redistribuidores de un programa libre obtengan para sí los derechos de patente, con lo que el programa dejaría de ser libre. Para ello, hemos dejado claro que sólo están autorizadas las patentes que permiten el uso libre del programa por cualquier persona.

A continuación se exponen los términos y condiciones precisos para la copia, distribución y modificación de los programas.

LICENCIA PÚBLICA GENERAL GNU TÉRMINOS Y CONDICIONES PARA LA COPIA, MODIFICACIÓN Y DISTRIBUCIÓN

La presente licencia se aplica a cualquier programa u obra de otro tipo que incluya una nota del titular de los derechos de autor en que se declare que puede ser distribuido conforme a los términos de la presente Licencia Pública General. En lo sucesivo, por "Programa" se entiende cualquier programa u obra de esa naturaleza, y por "obra basada en el Programa" se entiende el Programa o cualquier obra derivada de él con arreglo a la legislación sobre los derechos de autor: es decir, una obra que contenga el Programa o alguna porción de él, ya sea en forma literal o con modificaciones o traducciones a otros idiomas. (En lo sucesivo se entiende que la traducción está incluida en el término "modificación".)

La presente licencia no se aplica a actividades distintas de la copia, distribución o modificación; sólo estas actividades están comprendidas en su ámbito. El acto de ejecutar el Programa no está restringido, y la licencia sólo se aplica a un resultado obtenido del Programa si su contenido constituye una obra basada en el Programa (independiente de que se haya obtenido ejecutando el Programa). Si es así o no, depende de la función del Programa.

1. Está permitido copiar y distribuir copias fieles del código fuente del Programa tal como se recibió, en cualquier soporte, siempre que en cada copia se incluyan, de manera clara y adecuada, una nota sobre los derechos de autor y una declaración de ausencia de garantía y se mantengan intactas todas las notas en que se haga referencia a la presente Licencia y a la ausencia de garantía, y que se proporcione a quienes reciban el Programa, junto con él, una copia de la presente Licencia.

Está permitido cobrar una tarifa por el acto físico de transferir una copia; si se desea, se puede ofrecer una garantía a cambio de un pago.

2. Está permitido modificar la propia copia del Programa o cualquier parte de él, lo que constituirá una obra basada en el Programa, y copiar y distribuir tales modificaciones o tal obra según lo dispuesto en la sección 1 *supra*, siempre que se cumplan las siguientes condiciones:

- a) En los archivos modificados deben incluirse de forma clara notas que indiquen que los archivos se han modificado, y la fecha de los cambios.
- b) En cualquier obra que se distribuya o publique, que contenga total o parcialmente el Programa o que se derive del Programa o de partes de él, debe indicarse claramente que está autorizado el uso gratuito por terceros de la totalidad de la obra en los términos de la presente Licencia.

- c) Si el Programa modificado lee normalmente comandos de forma interactiva cuando se ejecuta, debe velarse por que, cuando el Programa inicie su ejecución en ese modo interactivo de la forma más habitual, aparezca un anuncio que contenga una nota sobre los derechos de autor y un aviso de la ausencia de garantía (o de la provisión de una garantía por el autor de las modificaciones), una indicación de que los usuarios pueden redistribuir el Programa con estas condiciones, e información sobre cómo obtener una copia de la presente Licencia. (Excepción: si el Programa en sí es interactivo pero normalmente no muestra ese anuncio, no es obligatorio que la obra basada en el Programa lo muestre.)

Estas exigencias se aplican a la obra modificada considerada como un todo. Si existen secciones identificables de esa obra que no se derivan del Programa y que pueden considerarse razonablemente como trabajos separados e independientes, la presente Licencia y sus términos no se aplicarán a esas secciones cuando se distribuyan como obras separadas. Sin embargo, cuando esas secciones se distribuyan como parte de un conjunto que sea una obra basada en el Programa, la distribución del conjunto deberá hacerse con arreglo a los términos de la presente Licencia, cuyos permisos a otras personas que reciban la licencia para su uso se extenderán a todo el trabajo, y por lo tanto a todas y cada una de sus partes, independientemente de quién las haya escrito.

Así pues, el propósito de esta sección no es exigir derechos o poner en duda los derechos sobre obras escritas enteramente por un autor; el propósito es, más bien, ejercer el derecho de controlar la distribución de las obras derivadas o colectivas basadas en el Programa.

Además, la simple adición de otro trabajo que no esté basado en el Programa al Programa en cuestión (o a una obra basada en el Programa) en algún soporte de almacenamiento o distribución no entraña que ese trabajo quede sometido a los términos de la presente Licencia.

3. Está permitido copiar y distribuir el Programa (o una obra basada en él, con arreglo a la sección 2) en forma de código objeto o en forma ejecutable según lo dispuesto en las secciones 1 y 2 *supra* siempre que se cumpla uno de los siguientes requisitos:

- a) Que se adjunte el correspondiente código fuente íntegro en lenguaje de máquina, que debe distribuirse conforme a lo establecido en las secciones 1 y 2 *supra* en un soporte utilizado habitualmente para el intercambio de programas informáticos; o
- b) Que se adjunte un ofrecimiento por escrito, válido por un período mínimo de tres años, de proporcionar a cualquier interesado, por un pago no superior al costo de la distribución física del código fuente, una copia íntegra del correspondiente código fuente en lenguaje de máquina, que se distribuirá conforme a lo establecido en las secciones 1 y 2 *supra* en un soporte utilizado habitualmente para el intercambio de programas informáticos; o
- c) Que se adjunte la información que se haya recibido sobre el ofrecimiento de distribución del código fuente correspondiente. (Esta opción se permite sólo para la distribución no comercial y únicamente si se ha recibido el Programa en forma de código objeto o en forma ejecutable con ese ofrecimiento de acuerdo con la subsección b) *supra*).

El código fuente de una obra se refiere a la forma preferida para introducir modificaciones. En el caso de un trabajo ejecutable, se entiende por código fuente íntegro todo el código fuente de todos los módulos que contenga, más cualquier archivo de definición de interfaz asociado, más los guiones (*script*) utilizados para controlar la compilación y la instalación del ejecutable. No obstante, como excepción especial, no se requiere que el código fuente distribuido incluya elementos que normalmente se distribuyan (en forma binaria o de código fuente) con los componentes principales (compilador, kernel, etc.) del sistema operativo en el cual se ejecuta el programa, a menos de que ese componente en particular acompañe al ejecutable.

Si la distribución del ejecutable o del código objeto se hace ofreciendo la posibilidad de copiarlo desde un lugar determinado, el ofrecer un acceso equivalente para copiar el código fuente desde el mismo lugar se considera una forma de distribución del código fuente, aunque los terceros no estén obligados a copiar el código fuente junto con el código objeto.

4. No está permitido copiar, modificar, o distribuir el Programa, ni otorgar licencias subsidiarias respecto de él, salvo en las condiciones expresamente determinadas en la presente Licencia. Todo intento de copia, modificación, otorgamiento de licencias subsidiarias o distribución del Programa de cualquier otra forma será nulo, y el autor de esos actos perderá automáticamente los derechos dimanantes de la presente Licencia. No obstante, las partes que hayan recibido copias o derechos en virtud de esta Licencia no verán revocados sus permisos ni perderán sus derechos mientras continúen cumpliendo los términos de la presente Licencia.

5. No es obligatorio aceptar la presente Licencia, si no se la ha firmado. Sin embargo, no puede obtenerse permiso para modificar o distribuir el Programa ni las obras derivadas de él por ningún otro medio. Esos actos están legalmente prohibidos si no se ha aceptado la presente Licencia. Por tanto, al modificar o distribuir el Programa (o cualquier obra basada en él) se manifiesta la aceptación de la presente Licencia y de todos sus términos y condiciones para la copia, distribución o modificación del Programa u otras obras basadas en él.

6. Cada vez que se redistribuye el Programa (o cualquier obra basada en él), el receptor recibe automáticamente una licencia del autor original para copiar, distribuir o modificar el Programa con sujeción a los presentes términos y condiciones. No puede imponerse a quienes reciban el Programa ninguna restricción adicional de los derechos que se les otorgan en la presente Licencia. La persona que redistribuye el Programa no está obligada a garantizar el cumplimiento de la presente Licencia por parte de terceros.

7. Si como consecuencia de una sentencia judicial o de una acusación de infracción de una patente o por cualquier otra razón (no sólo las relacionadas con cuestiones de patentes) se imponen a un usuario condiciones contrarias a los términos y condiciones de la presente Licencia, (ya sea por orden judicial, por acuerdo o de otra manera) ello no le eximirá del cumplimiento de los términos y condiciones establecidos en la presente Licencia. Si no es posible distribuir el producto cumpliendo simultáneamente las obligaciones establecidas en la presente Licencia y cualesquiera otras obligaciones, no se podrá distribuir el Programa. Por ejemplo, si una licencia de patente no permite la redistribución del Programa sin pago de regalías por parte de quienes lo reciban directa o indirectamente, la única forma de cumplir con ambas obligaciones será renunciar a su distribución.

Si cualquier parte de la presente sección resultase inválida o no se pudiera ejercitar en una circunstancia concreta, el resto de la sección será aplicable, y en las demás circunstancias se aplicará la sección completa.

La presente sección no tiene por objeto inducir a infringir ninguna patente ni ningún otro derecho de propiedad intelectual, ni poner en duda la validez de tales derechos; su único propósito es proteger la integridad del sistema de distribución de software libre, por la que velan las prácticas de la licencia pública. Muchas personas han hecho generosas contribuciones a la amplia gama de programas que se distribuyen al amparo de este sistema confiando en su aplicación coherente; corresponde al autor o donante decidir si desea que sus programas se distribuyan al amparo de otro sistema, y ninguna licencia puede imponer esa elección.

El propósito de la presente sección es aclarar de forma exhaustiva lo que se cree que es una consecuencia del resto de la presente Licencia.

8. Si la distribución o el uso del Programa estuvieran restringidos en algunos países por patentes o por interfaces protegidas por derechos de autor, el titular original de los derechos de autor que ponga su Programa al amparo de la presente Licencia podrá agregar una restricción expresa de la distribución

geográfica en la que se excluyan esos países, de forma que su distribución sólo se permita en los países no excluidos o entre esos países. En ese caso, la presente Licencia incorporará esa restricción como si estuviera incluida en su propio texto.

9. La Fundación para el Software Libre podrá publicar periódicamente versiones nuevas o revisadas de la Licencia Pública General. Esas nuevas versiones se inspirarán en el espíritu de la presente versión, pero podrán diferir en los detalles para abordar nuevos problemas o preocupaciones.

Cada versión lleva un número que la distingue. Si en el Programa se especifica que se aplicará a él una determinada versión de la presente Licencia y "cualquier versión posterior", el usuario tendrá la opción de acogerse a los términos y condiciones de dicha versión o a los de cualquiera de las versiones posteriores que publique la Fundación para el Software Libre. Si en el Programa no se especifica una versión concreta de la presente Licencia, el usuario podrá elegir entre cualquiera de las versiones que haya publicado la Fundación para el Software Libre.

10. Si se desea incorporar partes del Programa en otros programas libres cuyas condiciones de distribución sean distintas, deberá recabarse por escrito la autorización del autor. En el caso de los programas registrados por la Fundación para el Software Libre, habrá que dirigirse a dicha Fundación. En ocasiones hacemos excepciones. Nuestra decisión se guiará por los objetivos de garantizar que todos los derivados de nuestros programas libres sigan siendo libres y de promover el intercambio y la reutilización de los programas informáticos en general.

NO EXISTE GARANTÍA ALGUNA

11. DEBIDO A QUE LA LICENCIA PARA LA UTILIZACIÓN DEL PROGRAMA SE OTORGA GRATUITAMENTE, NO EXISTE NINGUNA GARANTÍA SOBRE EL PROGRAMA, EN LA MEDIDA EN QUE LA LEGISLACIÓN APLICABLE LO PERMITE. SALVO QUE SE INDIQUE OTRA COSA POR ESCRITO, LOS TITULARES DE LOS DERECHOS DE AUTOR PROPORCIONAN EL PROGRAMA "TAL COMO ES" SIN GARANTÍA DE NINGÚN TIPO, NI EXPLÍCITA NI IMPLÍCITA, NI SIQUIERA LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN Y ADECUACIÓN A UN PROPÓSITO CONCRETO. EL USUARIO ASUME TODO EL RIESGO RELACIONADO CON LA CALIDAD Y EL FUNCIONAMIENTO DEL PROGRAMA. SI EL PROGRAMA RESULTARA SER DEFECTUOSO, EL USUARIO ASUMIRÁ EL COSTO DE CUALQUIER SERVICIO, REPARACIÓN O CORRECCIÓN QUE SEAN NECESARIOS.

12. EN NINGÚN CASO, SALVO QUE LA LEY APLICABLE LO EXIJA O QUE SE ACUERDE POR ESCRITO, EL TITULAR DE LOS DERECHOS DE AUTOR O TERCEROS QUE HAYAN MODIFICADO Y/O REDISTRIBUIDO EL PROGRAMA EN LAS CONDICIONES ESTABLECIDAS EN LA PRESENTE LICENCIA, SERÁN RESPONSABLES DE LOS DAÑOS QUE EL PROGRAMA PUEDA CAUSAR, INCLUIDOS LOS DAÑOS GENERALES Y ESPECIALES Y LOS DAÑOS EMERGENTES DIRECTOS E INDIRECTOS, DERIVADOS DEL USO O DE LA IMPOSIBILIDAD DE USAR EL PROGRAMA (EN PARTICULAR LA PÉRDIDA O EL DETERIORO DE DATOS O LAS PÉRDIDAS QUE EL USUARIO U OTROS PUEDAN SUFRIR O EL HECHO DE QUE EL PROGRAMA NO PUEDA EJECUTARSE CON ALGÚN OTRO PROGRAMA), INCLUSO SI EL TITULAR DE LOS DERECHOS DE AUTOR U OTROS ESTABAN AL CORRIENTE DE LA POSIBILIDAD DE QUE OCURRIERAN ESOS DAÑOS.

FIN DE LOS TÉRMINOS Y CONDICIONES

Cómo aplicar estas condiciones a los nuevos programas

Si se desarrolla un nuevo programa y se desea que tenga la mayor utilidad posible para el público, el mejor modo de conseguirlo es hacer de él un software libre que se pueda redistribuir y modificar en las condiciones establecidas en la presente Licencia.

Para ello, deben añadirse al programa las notas que figuran más abajo. Es más seguro añadirlas al inicio de cada archivo fuente para indicar de manera más eficaz la ausencia de garantía; y cada archivo debe contener al menos la línea del "copyright" (derechos de autor) y una referencia al lugar donde se puede encontrar la nota completa.

<una línea que contenga el nombre del programa y dé una idea de su función.> Copyright© <año>
<nombre del autor>

Este es un software libre; se puede redistribuir o modificar con arreglo a los términos de la Licencia Pública General GNU publicada por la Fundación para el Software Libre, bien sea la versión 2 de esa Licencia o (a elección del usuario) cualquier versión posterior.

Este programa se distribuye con la esperanza de que sea útil, pero SIN NINGUNA GARANTÍA; incluso sin la garantía implícita de la COMERCIABILIDAD o la ADECUACIÓN A UN PROPÓSITO CONCRETO. Para más detalles, véase la Licencia Pública General GNU.

Si no ha recibido una copia de la Licencia Pública General GNU junto con este programa, solicítela a: Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307, Estados Unidos de América.

Deben adjuntarse asimismo las propias señas, incluidas una dirección de correo electrónico y la dirección postal.

Si el programa es interactivo, al iniciarse el modo interactivo debe aparecer en la pantalla una breve nota que diga lo siguiente:

Gnomovision versión 69, Copyright© <año> <nombre del autor>
Gnomovision no tiene NINGUNA GARANTIA; para conocer más detalles, escriba 'show w'. Este es un software libre, que se puede redistribuir bajo ciertas condiciones; para conocer más detalles, escriba 'show c'.

Estos comandos hipotéticos 'show w' y 'show c' deben dar acceso a las partes pertinentes de la Licencia Pública General. Por supuesto, los comandos pueden ser distintos, pueden ser incluso "clicks" del ratón, opciones de menús, etc., lo que resulte más apropiado para el programa.

El empleador del autor (si se trata de un programador) o, en su caso, su centro de estudio deben firmar una "renuncia a los derechos de autor" respecto del programa, si es necesaria. A continuación figura un ejemplo, en el que deberían sustituirse los nombres:

Yoyodyne, Inc., renuncia a los derechos de autor en relación con el programa 'Gnomovision' (que hace pasadas en compiladores) escrito por James Hacker.

<firma de Ty Coon>, 1º de abril de 1989
Ty Coon, Presidente

Esta Licencia Pública General no permite incorporar un programa en programas protegidos por derechos de autor. Si el programa es una biblioteca de subrutinas, puede ser más útil permitir que las aplicaciones protegidas por derechos de autor se vinculen con la biblioteca. Si eso es lo que se desea, debe utilizar la Licencia Pública General para Bibliotecas en lugar de la presente Licencia.

Anexo II

El texto que figura a continuación es una traducción del texto en inglés de la Definición de Fuente Abierta que aparecía en el sitio de la Open Source Initiative en Internet:
<http://www.opensource.org/docs/definition.php> el 13 de agosto de 2003.

DEFINICIÓN DE FUENTE ABIERTA

Versión 1.9

Los párrafos sangrados y en bastardilla que figuran a continuación son anotaciones a la definición de fuente abierta (OSD) y **no** forman parte de ella.

Introducción

Fuente abierta no significa sólo acceso al código fuente. Los términos para la distribución de software de fuente abierta tienen que cumplir los siguientes criterios:

1. Redistribución libre

La licencia no debe impedir la venta o el ofrecimiento del software como componente de una distribución de programas agregados que contengan programas de diversas fuentes. La licencia no debe requerir el pago de derechos de autor u otra tasa por esa venta.

***Fundamento:** Al imponer que en la licencia se exija la redistribución libre, se elimina la tentación de prescindir de importantes ganancias a largo plazo para conseguir unos pocos ingresos en concepto de ventas a corto plazo. Si no se hiciera esto, habría una fuerte presión para que los cooperadores desertasen.*

2. Código fuente

El programa tiene que incluir el código fuente, y tiene que permitir la distribución tanto en código fuente como en forma compilada. Si alguna forma de un producto no se distribuye con el código fuente, tiene que haber un medio claramente indicado de obtener el código fuente por no más que un costo razonable de reproducción, preferentemente mediante una descarga a través de Internet sin costo alguno. El código fuente tiene que ser la forma preferida en que un programador modificará el programa. Los códigos fuente que deliberadamente se han hecho ininteligibles no están permitidos. Tampoco lo están las formas intermedias como la salida de un preprocesador o de un traductor.

***Fundamento:** Exigimos el acceso a un código fuente que no se haya hecho ininteligible porque los programas no pueden evolucionar si no se modifican. Puesto que nuestro propósito es facilitar la evolución, requerimos que se facilite la modificación.*

3. Obras derivadas

La licencia tiene que permitir modificaciones y obras derivadas, y que éstas se distribuyan en las mismas condiciones que las de la licencia del software original.

***Fundamento:** La mera capacidad de leer el código fuente no es suficiente para promover la revisión independiente por los colegas y la selección evolutiva rápida. Para que se produzca una*

evolución rápida, es necesario que el público pueda experimentar con modificaciones, y redistribuirlas.

4. Integridad del código fuente del autor

La licencia puede restringir la distribución del código fuente en forma modificada solamente si permite la distribución de "archivos parche" con el código fuente a fin de modificar el programa en el momento de la construcción. La licencia tiene que autorizar explícitamente la distribución de los programas informáticos construidos a partir del código fuente modificado. La licencia puede requerir que las obras derivadas lleven un nombre distinto o un número de versión distinto al del software original.

***Fundamento:** Alentar un gran número de mejoras es bueno, pero los usuarios tienen derecho a saber quién es el responsable del software que están utilizando. Los autores y quienes mantienen los programas tienen recíprocamente derecho a saber qué se les pide que apoyen y a proteger su reputación.*

Por consiguiente, una licencia de fuente abierta debe garantizar que esa fuente sea de fácil acceso, pero puede requerir que se distribuya como fuente base virgen más "parches". De esta manera, se pueden ofrecer cambios "no-oficiales" que se distinguen fácilmente del código fuente base.

5. No discriminación de personas o grupos

La licencia no tiene que discriminar a ninguna persona o grupo de personas.

***Fundamento:** Para obtener el máximo beneficio del proceso, debe poder contribuir por igual a las fuentes abiertas la mayor diversidad posible de personas y grupos. Por lo tanto, en una licencia de fuente abierta no está permitido excluir a ninguna persona o grupo de la participación en el proceso.*

Algunos países, entre ellos los Estados Unidos, han impuesto limitaciones a la exportación de determinados tipos de software. En una licencia de fuente abierta se puede advertir a los concesionarios de las restricciones aplicables y recordarles que están obligadas a cumplir la ley; sin embargo, no se pueden incorporar tales restricciones en la propia licencia.

6. No discriminación de ámbitos de aplicación

La licencia no puede prohibir a nadie que haga uso del programa en un ámbito específico. Por ejemplo, no puede prohibir que el programa se utilice en una empresa, o con fines de investigación genética.

***Fundamento:** El principal propósito de esta cláusula es evitar que se incluyan en las licencias trampas que impidan que la fuente abierta se utilice comercialmente. Queremos que los usuarios comerciales se unan a nuestra comunidad, no que se sientan excluidos de ella.*

7. Distribución de la licencia

Los derechos vinculados al programa han de aplicarse a todos aquellos a quienes se redistribuya el programa, sin que esas partes tengan que disponer de una licencia adicional para la ejecución.

***Fundamento:** Esta cláusula tiene por objeto prohibir que los programas se cierren por medios indirectos, como la imposición de un acuerdo de confidencialidad.*

8. La licencia no debe ser específica de un producto

Los derechos vinculados al programa no tienen que depender de que el programa forme parte de una distribución de software particular. Si el programa se extrae de esa distribución y se utiliza o distribuye de acuerdo con las condiciones de la licencia, todas las partes a las que se redistribuya deben tener los mismos derechos que se conceden con la distribución original del programa informático.

Fundamento: Con esta cláusula se evita que se incluyan otras trampas en la licencia.

9. La licencia no debe imponer restricciones a otros programas

La licencia no puede imponer restricciones a otros software que se distribuyan con el programa autorizado. Por ejemplo, la licencia no debe insistir en que todos los demás programas distribuidos por el mismo medio sean programas de fuente abierta.

Fundamento: Los distribuidores de programas de fuente abierta tienen derecho a tomar sus propias decisiones sobre sus propios programas.

Sí, la Licencia Pública General (LPG) cumple este requisito. El software vinculado con bibliotecas que se han acogido a esta licencia hereda la LPG solamente si conforma una obra aparte, no un programa con el cual simplemente se distribuye.

10. La licencia tiene que ser tecnológicamente neutral

En la licencia no pueden introducirse disposiciones que favorezcan una tecnología o un tipo de interfaz concretos.

Fundamento: Esta disposición está dirigida específicamente a las licencias que requieren un gesto explícito de aprobación para establecer un contrato entre el licenciante y el concesionario. Las disposiciones por las que se impone la denominada "aprobación mediante click" pueden estar en conflicto con métodos importantes de distribución de programas como la descarga mediante FTP, las antologías en CD-ROM, y la replicación de sitios en la Web; esas disposiciones pueden también obstaculizar la reutilización del código. En las licencias acordes con la definición de fuente abierta se debe permitir la posibilidad de que (a) la redistribución del programa se realice por canales distintos de la Web que no utilicen la grabación mediante click a la descarga, y (b) el código cubierto (o las porciones reutilizadas del código cubierto) pueda funcionar en un entorno que no disponga de interfaz gráfica del usuario y que no utilice diálogos de aparición directa.

Anexo III

DECLARACIÓN DE LA FUNDACIÓN PARA EL SOFTWARE LIBRE Y DE CÓDIGO FUENTE ABIERTO PARA ÁFRICA (FOSSFA)

El texto que figura a continuación es una traducción del texto en inglés que aparecía en <http://www.prepcom.net/wsis/1046170300> el 13 de agosto de 2003.

Preámbulo

El potencial de la fuente abierta mejorará la productividad y la calidad de vida en los países en desarrollo. El proceso de transformación en sociedades de la información requiere la plena participación de todos los Estados miembros.

Visión

Nuestra visión es promover los programas informáticos sostenibles, viables y asequibles para África mediante la formación y la creación de capacidad local.

Principios

África debería estudiar la forma de aprovechar las oportunidades que ofrece la aparición del software de fuente abierta en un contexto de limitados recursos financieros y conocimientos especializados.

Aspectos específicos

África puede salvar la "brecha digital" mediante la adopción de la fuente abierta, con lo que se reducirá el efecto del colonialismo tecnológico.

Plan de acción

La FOSSFA, en asociación con gobiernos, organizaciones intergubernamentales, la sociedad civil y otros interesados, tiene previsto poner en marcha iniciativas de creación de capacidad mediante la formación y habilitación de las mujeres y los jóvenes.

La FOSSFA tratará de influir en todos los interesados para que adopten la fuente abierta como plataforma para el desarrollo de soluciones que satisfagan las necesidades de la población.

Estrategias

La FOSSFA se ocupará de:

- i. Fomentar en África la sensibilización acerca del software libre y del código fuente abierto.
- ii. Crear capacidad en la esfera del software libre y del código fuente abierto.
- iii. Desarrollar un acervo de conocimientos especializados en África.
- iv. Desarrollar el portal africano del software de fuente abierta.

Para alcanzar estos objetivos nos proponemos:

- i. Influir en órganos clave como la Unión Africana, la Comisión Económica para África de las Naciones Unidas, el Programa de las Naciones Unidas para el Desarrollo (PNUD), la Agencia de la Francofonía y la Nueva Alianza para el Desarrollo de África (NEPAD), entre otros, para que apoyen el desarrollo de la fuente abierta en África.
- ii. Aprovechar los diferentes recursos y capacidades de África en la esfera del software libre y de la fuente abierta.
- iii. Influir en los gobiernos donantes y otras instituciones para que vinculen la financiación de las tecnologías de la información y las comunicaciones a la utilización de software libre y de fuente abierta.
- iv. Influir en los gobiernos africanos para que adopten software libre y de fuente abierta.
- v. Fomentar, mediante la formación, el desarrollo de la capacidad y los conocimientos en la esfera de los programas informáticos de fuente abierta en África, especialmente entre las mujeres y los jóvenes.