Apuntes sobre software libre

Jesús M. González Barahona (y muchos otros)

18 de marzo de 1999

[1] `http://www.openresources.com/magazine/license/index.html`

NOTA IMPORTANTE:

Esta versión es un borrador preliminar, sobre el que se está trabajando aún activamente. Entre otras cosas, incluye notas sobre algunas de las modificaciones previstas (marcadas como “comment”).

Índice General

Prefacio

El origen de estos apuntes puede trazarse hasta un mensaje de Ismael Olea Gonzalez (olea@poboxes.com), puesto en la lista PDSOFT en mayo de 1998. El siguiente párrafo podría considerarse como el catalizador:

Desde hace tiempo llevo rumiando una idea que podría llegar a materializarse en este foro. Se me ha ocurrido la idea de crear, en plan cooperativo, un documento que profundice en el software libre que, puesto a disposición del público, sirva de base para impartir conferencias acerca del software libre. Yo mismo tenía la idea de crear unos contenidos para impartir conferencias acerca del software libre y Linux en Almería y en reuniones de linuxeros pero debido a mi falta de tiempo apenas puedo profundizar en el tema. Así pues lanzo este proyecto acompañándolo con un guión con los puntos a estudiar.

Por esa época, Jesús M. González Barahona (jgb@computer.org) y Joaquín Seoane (jseoane@dit.upm.es) estaban preparando una charla sobre software libre, y el mismo Jesús, junto con Pedro de las Heras (pheras@acm.org), planeaban un seminario sobre el mismo tema en su Universidad.

Este conjunto de situaciones llevó al primer borrador de este documento, basado en el esfuerzo de muchas personas.

Después de eso, mucha gente se unío al proyecto, de varias formas:

- Carmen Ugarte colaboró con comentarios, traducciones, y su investigación sobre la situación del software libre en IBM antes de 1968.

Capítulo 1

Introducción

La industria del software está centrada en un modelo de producción que a lo largo de estos apuntes denominaremos "propietario". Sus ingresos son el resultado de cobrar un precio por copia a sus clientes. La legislación actual de derechos de copia (*copyright*) está diseñada para favorecer esta práctica, y la insdustria del software propietario está intentando que se acomode aún más a sus intereses.

Como en este modelo económico los ingresos provienen de la venta de copias, las empresas de producción software están muy incentivadas para dedicar sus esfuerzos a vender más y mas copias. Por lo tanto, tienden a limitar lo más posible las posibilidades de redistribución de las copias que ellas no controlen (como por ejemplo, los préstamos entre usuarios). Por otra parte, como sólo la empresa que tiene los derechos de un programa dado puede mejorarlo y adpatarlo a las necesiades de sus clientes, se corta de raíz la competencia, que queda limitada a la posibilidad de los usuarios para adquirir un programa u otro. Sin embargo, una vez que el usuario ha elegido un programa, no puede acudir a cualquier profesional o empresa para que lo adapte a sus necesidades o lo mejore.

Sin embargo, este modelo no es el único posible. Al menos hay otro, que ya ha demostrado su viabilidad con la producción de sistemas tan completos como GNU/Linux, y ha sido utilizado de diversas formas por empresas tan importantes en la industria informática como Netscape, Motorola, IBM o Corel. El tipo de software desarrollado según este modelo se llama "software libre", por la libertad que tiene cualquier persona o empresa que lo usa para adaptarlo a sus necesidades y redistribuirlo a terceras partes.

Aunque el software libre no es algo que comenzara a desarrollarse ayer (sus comienzos pueden trazarse a la década de 1970), en los finales del siglo XX estamos asistiendo a una explosión en sus niveles de producción y de uso. La calidad que han alcanzado algunos de los programas libres disponibles en la actualidad, y las implicaciones que tiene su uso y desarrollo hacen absolutamente necesario que la sociedad en general y los usuarios informáticos en particular (ya sean personas o empresas) conozcan su existencia y consideren su utilidad. Lo contrario puede suponer la pérdida de enormes oportunidades.

En estos apuntes, intentaremos describir el software libre, sus modelos de desarrollo y distribución, sus ventajas y desventajas, la situación actual en torno a él y las evoluciones futuras previsibles y, en general, todo lo relevante para alguien que desee aproximarse a esta realidad.

> **Comment (jgb):** *We should rewrite and complete all this introduction. Let's inlclude some comments on the fact that the notes are completely written using free software, about where to get current versions from the network, etc.*

Capítulo 2

¿Qué es el software libre?

No es fácil definir con pocas palabras el término "software libre", debido a las muchas categorías y variantes que existen. Por otra parte, tampoco es excesivamente complicado, pues la idea en si misma sí es sencilla. Así pues, antes de entrar en definiciones más estrictas dedicaremos un momento a explicar, de una forma un tanto informal, qué se entiende por software libre. Más adelante, en este mismo capítulo, discutiremos el asunto más formalmente, estudiando en detalle varias licencias y definiciones de software libre.

2.1 Idea general de software libre

Cuando se habla en inglés de "*free software*", hay una peligrosa ambiguedad, debido que "*free*" significa tanto "libre" como "gratis". Afortunadamente, en castellano no tenemos esta ambiguedad[1], y el significado de "software libre" está mucho más claro.

De todas formas, para empezar es bueno dejar claro que el software libre no tiene porqué ser gratis. Es más, no suele serlo, o al menos no completamente. Su característica fundamental es la libertad que tiene cualquier usuario para:

- Disponer del software para adaptarlo a sus necesidades. Naturalmente, esto incluye hacerle mejoras, corregir erratas, aumentar su funcionalidad o estudiar su funcionamiento. Y, obviamente, usarlo como quiera.
- Redistribuir el software a otros usuarios, que a su vez podrán también disponer de él según sus necesidades. Esta redistribución puede hacerse gratuitamente o cobrando un precio, que no está fijado de antemano[2].

Es importante dejar claro ya que estamos hablando de libertad, y no de obligación. Esto es, un usuario de un programa libre puede modificarlo si lo estima conveniente, pero desde luego no está obligado a ello. De la misma manera, puede redistribuirlo, pero en general tampoco está obligado a ello.

Para poder cumplir las dos condiciones anteriores, hay una tercera que es básica, y se deriva necesariamente de ellas:

- Los usuarios del software en cuestión deben tener acceso a su código fuente.

El código fuente de un programa, habitualmente escrito en un lenguaje de programación de alto nivel, es absolutamente necesario para poder entender su funcionamiento, modificarlo y mejorarlo. Si un programador dispone del código fuente de un programa, puede estudiarlo, conocer todos sus detalles y trabajar sobre él como lo haría el autor original.

Paradójicamente, si se quieren garantizar estas libertades para un software dado, con la legislación actual, es necesario "protegerlo" con una licencia que imponga ciertas restricciones a la forma en que pueda utilizarse y distribuirse (como se verá más adelante). Este hecho provoca cierta controversia en algunos ambientes, al considerar que estas licencias hacen que el software que se distribuye con ellas sea "menos libre". Otra opinión, más pragmática, es que el software será tanto más libre cuanto más se garantice el mantenimiento de estas

[1] De hecho, en la comunidad software de habla inglesa se está extendiendo el término "libre software", usando el vocablo castellano, para evitar la ambiguedad de "*free*'.

[2] Puede ser revelador darse cuenta de que el precio que se cobra por la redistribución no está relacionado de ninguna forma con el precio que se pagó cuando se obtuvo el software.

libertades para todos sus usuarios, para lo que es imprescindible limitar los modos de uso y distribución. Por ello es muy importante, en el momento de considerar qué se puede hacer con un programa libre, consultar la licencia bajo la que se distribuye. Esa misma licencia servirá también para decidir si un programa es software libre o no.

2.2 ¿Qué no es software libre?

Hay ciertos tipos de programas que a veces se incluyen (equivocadamente) dentro de la categoría de software libre. Sin embargo, debido a sus caracterísiticas, presentan una casuística muy alejada de la del software libre, y por ello no serán objeto de estudio en este documento. Entre ellos están el software "a prueba", el *shareware*, el software "regalado" y el distribuido con código fuente pero sin licencia de redistribución.

2.2.1 Software "a prueba"

El modelo de distribución de este tipo de programas está basado en la disponibilidad gratuita del mismo durante cierto tiempo o en ciertas condiciones. Ese tiempo (o esas condiciones) suelen proponerse para que el usuario pueda probar el programa. Si su funcionamiento le convence, puede a continuación comprarlo, de forma similar a como lo haría con cualquier software propietario, y adquiriendo en general derechos y obligaciones iguales a las habituales en el software propietario. La redistribución del programa (incluso a veces durante el periodo de prueba) suele estar absolutamente prohibida.

Desde cualquier punto de vista, este software es completamente propietario, con la única diferencia que hay un tiempo o unas condiciones de prueba. Pero este hecho no cambia su naturaleza, ni permite a este tipo de programas beneficiarse de la mayor parte de las ventajas que se obtienen con el software libre.

> **Comment (jgb):** *More examples are needed. It is convenient to include here extracts of the corresponding licenses.*

2.2.2 *Shareware*

Los programas *shareware* habitualmente pueden redistribuirse con libertad, y utilizarse durante un tiempo a prueba. Pero cuando alguien decide utilizarlos más allá del periodo de prueba (o bajo ciertas condiciones, como por ejemplo uso empresarial) debe pagar una cierta cantidad a su autor. En algunos casos el programa se "protege" para que deje de funcionar, o funcione de forma limitada, si no se ha pagado al autor. En otros, el programa que se distribuye es perfectamente funcional tal cual, y la voluntad del autor se protege únicamente por una licencia. Además, normalmente la distribución se hace sólo en versión binaria, sin que el usuario tenga acceso al código fuente.

Por lo tanto, lo que se introduce con esta modalidad es fundamentalmente una forma "diferente" de cobrar por el software. La principal diferencia al compararlo con el software a prueba es la libertad de redistribución (que no de uso).

2.2.3 Software "regalado"

Hay veces en que el software propietario se regala. Las motivaciones para esto pueden ser muy variadas, desde promociones para conseguir que un programa se imponga en un determinado mercado hasta intentos de derribar a una empresa competidora. Este software es claramente propietario, y en general las licencias que lo cubren no se diferencian en nada de las "tradicionales". La única diferencia está en las condiciones de adquisición, dado que es gratis.

> **Comment (jgb):** *Examples? For instance, Explorer, Acroread*

2.2.4 Software con fuentes

A veces se distribuye el software incluyendo su código fuente, o permitiendo el acceso a él. Pero si no se permite redistribuir el programa y modificar el fuente y redistribuir las modificaciones, no estamos ante software libre. Por ejemplo, las primeras versiones de Unix, desarrolladas en los Bell Labs, se distribuían como código fuente, con una licencia de AT&T que impedía su redistribución libre a terceras partes.

Comment (jgb): *The Unix case should be documented, and perhaps talk about the consequences of that kind of license of ATT, and its relationship with the work of the CSRG at Berkeley. BSD/386 could also be an interesting case*

Comment (jgb): *Elaborate a bit more. We can talk about Qt for GNU/Linux, documenting it with its licences.*

2.3 Un nombre para el software "no libre"

En unos apuntes dedicados al software libre, parece importante encontrar una palabra que defina sin ambiguedades al software que no lo es. Sin embargo, la tarea no es fácil, pues hay que encontrar una palabra objetiva y clara para algo que ya recibe muchos nombres. Este término ha de cubir tanto a los programs no libres "clásicos" como a los descritos en los apartados anteriores. Los tres candidatos que nos han parecido más adecuados son[3]:

Software no libre. Posiblemente el término más claro y preciso, pero con la desventaja de ser una mera denominación por negación, que no revela ninguna característica común de los programas a los que denomina.

Software propietario. Traducción directa de un término muy utilizado en inglés (*propietary software*). Refleja el hecho de que la propiedad absoluta del programa permanece en manos de quien tiene sus derechos, y no del usuario, que únicamente puede utilizarlo bajo ciertas condiciones. Sin embargo, la semántica de esta palabra en castellano es cuando menos dudosamente aplicable de esta forma.

Software cautivo. Posiblemente la mejor denominación desde un punto de vista semántico. Mientras que el software libre es "libre", y puede ser utilizado como tal, sin más cortapisas que las que aseguren su libertad, el software no libre puede considerarse como "cautivo" de una licencia que impide hacer con él muchas cosas que por su propia naturaliza podrían hacerse (como mejorarse o adaptarse a las necesidades de un usuario concreto). Sin embargo, el término "cautivo" tiene muchas connotaciones que hacen que pueda ser entendido como poco objetivo.

En estos apuntes, utilizaremos los tres términos, aunque el preferido será, en general, "software propietario".

Comment (jgb): *Is this section needed in the English version???*

2.4 ¿Copiar o no copiar?

Aunque hoy día hay una tendencia por parte de ciertos sectores a pretender lo contrario, es importante darse cuenta de que, cuando hablamos de información, lo "natural" es copiar. Lo único que lo puede evitar es que el coste de la realización de la copia sea alto o que la legislación restrinja el derecho a hacerlo. Hay que tener en cuenta que la información no es (en general) un "bien escaso", tal y como se entiende en las teorías económicas. La información puede "duplicarse" (copiarse) sin que se degrade en ninguna medida el original, y teniendo el duplicado (la copia) exactamente la misma calidad que el original[4]. Por ello, las dos partes que intervienen directamente en la copia (el individuo que tiene el orginal y el que recibe la copia) están motivados para hacerla. En general, ambos salen beneficiados cuando se realiza una copia de inforamción, y especialmente quien la recibe.

Algunos economistas, que se han percatado del hecho de que la información no puede tratarse como un bien económico "clásico", están ya buscando formas de entender una economía basada en la información[5].

[3] Los tres términos que se proponen son el fruto de una discusión que tuvo lugar en junio de 1998 en la lista de correo *Spanglish* (spanglish@eunet.es), dedicada a temas de traducción de términos informáticos entre el inglés y el castellano.

[4] Esto es aplicable a la copia de información "pura", no a su soporte físico. Por ejemplo, si copiamos la "información" que compone una poesía (las palabras que componen sus versos) tenemos exactamente la misma inforamción que tenía la poesía original. Sin embargo, si fotocopiamos la página de un libro donde estaba escrita la poesía, la calidad de la fotocopia normalmente será diferente de la de la página (papel) orginal. Dicho esto, es importante darse cuenta que un programa informático es "información pura", en el sentido de que no importa en qué soporte físico esté grabada, en tanto en cuento se mantenga su integridad.

[5] Uno de estos economistas es Ted Lewis, que denomina a este tipo de economía "economía sin fricción" (*friction-free economy*)[4].

Comment (jgb): *Ted Lewis writes in Computer. I have to cite some of his articles, and perhaps include in an appendix an abstract of his theories.*

Antes de discutir los motivos que pueden aconsejar la restricción de los derechos de copia, vamos a presentar brevemente algunos ejemplos que refuerzen la idea de que la copia de información es algo habitual en nuestra experiencia diaria:

- El cotilleo (y el periodismo).
- Los exámenes no competitivos.
- Las recetas de cocina.
- Las composiciones musicales.
- Las teorías científicas (y su soporte matemático).
- In general, the spread of culture.

Comment (jgb): *All these examples should be detailed. Perhaps substitute or include some more clear examples*

Richard Stallman ilustra a menudo estas ideas con una analogía bastante reveladora. Imagina el invento de un aparato que permite "copiar" piezas de pan, con coste prácticamente cero. Las nuevas piezas son exactamente iguales que las "originales". Para tener una nueva pieza, todo lo que hace falta es que alguien te preste momentáneamente la suya, y producir una copia. Es sencillo entender que con este invento cualquiera estaría dispuesto a prestar su pieza de pan para que otros pudieran obtener nuevas piezas... El proceso de copia de información es muy parecido. Sin embargo, tendemos a pensar en él de forma diferente a lo que pensaríamos de ese "copiador de pan".

Comment (jgb): *Document that analogy in some article of rms*

Pero si la copia de información es tan sencilla y aparentemente beneficiosa para todos, ¿por qué la sociedad permite prohibir la copia? La legislación sobre derechos de copia (*copyright*) es relativamente moderna, y fue promulgada como un delicado balance entre el derecho a copiar información y la necesidad de promover la producción de nueva información, asegurando que los autores son recompensados adecuadamente, para aumentar su motivación.

Comment (jgb): *The origins of the copyright legislation should be documented, and we should be sure that these were the initial motivation of that legislation (see fsb archive).*

De hecho, la protección que obtienen los autores (o los poseedores de los derechos de copia) no son absolutos. Por ejemplo, esos derechos (a diferencia de los derechos de propiedad sobre bienes materiales) expira con el tiempo, y hay instituciones (como las bibliotecas públicas) que están sujetas a ellos de forma más flexible que el público en general.

Desde otro punto de vista, podría entenderse la legislación de copyright como una medida proteccionista por parte de la sociedad, que impone un "impuesto" a la copia de información. Ese impuesto es la cantidad que hay que pagar a quien detenta los derechos de copia (copyright) cada vez que esta copia se realiza (por ejemplo, cada vez que se compra una película de video, un disco de música o incluso un libro). Es un impuesto desde el momento que el hecho en sí (sacar una copia de la película, del disco o del libro) sería muy barato, o virtualmente gratis con las tecnologías actuales. Sólo porque la legislación lo impone hay que pagar un cierto sobrecoste. Curiosamente, este impuesto no lo recibe el estado, sino directa o indirectamente el dueño de los derechos de copia. La suposición implícita es que este dinero extra que se paga permite que se mantenga o incremente (en términos cualitativos o cuantitativos) la producción del tipo de información en cuestión (películas, música o libros, por ejemplo). Otra cosa es que esto sea realmente así, o al menos que sea así en todos los casos y para todos los tipos de información donde se aplica este modelo.

Por lo tanto, la sociedad permite la limitación legal del derecho de copia como una forma de asegurarse una mayor producción (cuantitativa y cualitativamente) de información. En cuanto el lector se percata de este hecho, surge inmediatamente la pregunta: "En el caso del software, ¿se beneficia la sociedad de la limitación del derecho de copia?".

La respuesta, desde luego, es un tema que permite largas discusiones. Sin pronunciarnos en ningún sentido, analicemos algunos de los mecanismos habituales de evolución y mejora en la industria software. Desde luego, la lista que ofrecemos a continuación no es exhaustiva, y cada desarrollador de software añadirá (o quitará) algún punto de ella. De todas formas, sí analizaremos el impacto que produce en cada mecanismo la disposición libre del código fuente, que a nuestro modo de ver es el aspecto más esencial (desde el punto de vista técnico). Esperamos que esta breve exposición pueda servir para que el lector vaya formándose sus propias ideas al respecto.

> **Comment (jgb):** *This last paragraph is no tin very good shape... We should rewrite it, perhaps changing the focus*

- **Mejora por imitación**. Es muy común hacer un programa similar a otro que ya existe, mejorando alguno de sus aspectos, pero manteniendo muchos otros. Esto puede hacerse reescribiendo completamente el programa, y de hecho es el caso más común, debido a que normalmente el código fuente del programa imitado no está disponible. Sin embargo, si lo estuviese, los "imitadores" podrían concentrarse en las mejoras que quieran realizar, y reutilziar completamente el resto del programa. Al poder concentrar sus esfuerzos en al innovación, esta se realizará con mucho menos esfuerzo. A igualdad de recursos, el resultado será en general mejor. Incluso si se decide empezar el nuevo programa de cero, el poder leer y estudiar el código del imitado allanará mucho el camino a sus desarrolladores.

  > **Comment (jgb):** *Probably this can be documented with the example of word processors for PC, from WordStar and its precursors to Word, Ami Pro and the like. Perhaps also with spread sheets, from Visi Calc to Excel. It can be c ompared, for instance, with the improvement by imitation in the free Unix-like systems (Linux, BSDs, etc., where many pieces of code has its source in one of them, then somebody ports it and improves it in other, etc.)*

- **Mejora incremental**. Es la forma habitual de mejorar un producto software dado, mediante una sucesión de versiones. La empresa que lo desarrolla parte del código fuente de una versión, y corrigiendo erratas y añadiendo nueva funcionalidad, produce una nueva versión del producto. Naturalmente, si sólo la empresa en cuestión puede usar ese código fuente, sólo ellos pueden mejorarlo incrementalmente. Sin embargo, si el código fuente está disponible para otros desarrolladores, todos ellos pueden realizar esta mejora incremental. Esto permite, por ejemplo, que un programa se desarrolle en direcciones no previstas por sus creadores, y que haya competencia en el desarrollo de nuevas versiones, lo que en general no puede sino benecificar los usuarios.

  > **Comment (jgb):** *As documentation, almost any program can be chosen, like for instance Windows, Novell, Solaris, etc. It can be compared to the incremental development of GNU/Linux, or Emacs and its "competition" with XEmacs, or the "competition" between *BSD. In fact, it could be concluded that when we speak about libre software, incremental improvement and improvement by imitation are almost the same, and what is usually found is a mixture of both*

- **Mejora por integración**. En el mundo del software propietario, es común que cuando una empresa quiere incluir nuevas funcionalidades en su programa, decida llegar un acuerdo con otra empresa que ya tenga esa funcionalidad desarrollada, para integrala directamente (quizás tras una adaptación). De hecho, en muchos casos el procedimiento utilizado es comprar la empresa productora de esa funcionalidad. Para el comprador eso tiene la ventaja de eliminar, de paso, a un posible competidor. Pero sólo cuando se llegua a este acuerdo, o cuando se usan programas de la propia empresa, se puede mejorar software por integración. Y en muchos casos las empresas dedicadas al software propietario están motivadas a no llegar a esos acuerdos, pues pueden ver en el "integrador" un competidor. En el mundo del software libre, y en general cuando el código fuente se puede usar libremente, la integración es lo habitual. Esto permite mejorar un programa integrando en él muchos trabajos ya realizados, e incluso beneficiarse de sus mejoras cuando esos otros trabajos se van actualizando.

  > **Comment (jgb):** *As an example fo the proprietary world, we can comment on the current strategy of Microsoft of buying everything which moves. As example in the free world, GNU/Linux is probably the paradigm. Also, some distributions, like Debian. In fact, it could be interesting to make a comparation between how Win* and Debian are including new functionality.*

- **Mejora por competencia "marginal".**

    Comment (jgb): *Joaquín, I don't know what you were refering to here...*

Comment (jgb): *Would be appropriate to put here something like "As a conclussion, we can say that, in many cases, libre software contributes to increase (and improve) software production"?*

Comment (jgb): *Probably all of this should be rewritten, probably with subsection or something like that.*

Comment (jgb): *We should talk somewhere about software as a service. Is this a good place?*

2.5 Licencias, licencias, licencias

Comment (jgb): *Reread the paper about licences [2]), and revise this section (and cite it, of course)*

Con el marco legal actual, la licencia bajo la que se distribuye un programa delimita exactamente los derechos que tienen sobre él sus usuarios. Por ejemplo, en la mayoría de los programas propietarios la licencia priva al usuario de los derechos de copia, modificación, préstamo, alquiler, uso en varias máquinas, etc. De hecho, las licencias suelen especificar que la propietaria del programa es la empresa editora del mismo, que simplemente vende derechos restringidos de uso del mismo.

Comment (jgb): *Extract of one (or some) licenses where the restrictions for users are specified. For instance, that of CorelDraw, o some other. Several of these licenses can be included in an appendix, and cite them here.*

En el mundo del software libre, la licencia bajo la que se disribuye un programa también va a ser de gran importancia. Normalmente, las condiciones de las licencias de software libre son el resultado de un compromiso entre varios objetivos hasta cierto punto contrapuestos. Entre ellos pueden citarse los siguientes:

- Garantizar algunas libertades básicas (de redistribución, de modificación, de uso) a los usuarios.
- Asegurar algunas condiciones impuestas por los autores (cita del autor en trabajos derivados, por ejemplo).
- Procurar que los trabajos derivados sean también software libre.

Los autores pueden elegir proteger su software con distintas licencias según el grado con que quieran cumplir cada uno de estos objetivos, y los detalles que quieran asegurar. De hecho, el autor puede (si así lo desea) distribuir su software con licencias diferentes por canales (y con precios) diferentes[6]. Por lo tanto, el autor de un program suele elegir con mucho cuidado la licencia bajo la que lo distribuye. Y los usuarios, y especialmente quien redistribuya o modifique el software, debe estudiar con cuidado su licencia.

Afortunadamente, aunque cada autor podría utilizar una licencia diferente para sus programas, en realidad casi todo el software libre usa una de las licencias más habituales (GPL, LGPL, Artistic, "estilo" BSD, "estilo" Netscape, etc.), algunas veces con ligeras variaciones. Para simplificar aún más las cosas, en los últimos tiempos están apareciendo organizaciones que promueven "marcas" que garantizan que todo el software que cubren está distribuido bajo licencias que aseguran ciertas condiciones "razonables" y simples de entender. Un ejemplo notable de esto es la marca Open Source[7].

En los siguientes apartados vamos a analizar con algo de detalle alguna de las licencias bajo las que distribuye habitualmente el software libre.

[6]Por ejemplo, hay autores que deciden distribuir su software con una licencia que garantize que los trabajos derivados sean también software libre. Sin embargo, en algunos casos, venden ese mismo software a empresas interesadas en mantener propietarios sus trabajos derivados, bajo una licencia diferente que lo permita.

[7]`http://www.opensource.org`

2.5.1 BSD (Berkeley Software Distribution)

La "licencia BSD" cubre, entre otro software, las entregas de BSD (Berkeley Software Distribution). Estas entregas fueron hechas por el CSRG (Computer Science Research Group) de la Universidad de California en Berkeley. Las entregas de BSD fueron la forma en que el CSRG distribuía su trabajo (y las contribuciones de muchos otros programadores) alrededor del sistema operativo Unix. De hecho, Unix era un sistema operativo propietario, y por elo lurante mucho tiempo los usuarios de las entregas de BSD necesitaban también una licencia Unix. Las entregas de BSD se usaron como la base de muchos sistemas operativos propietarios, como SunOs de Sun Microsystems o Ultrix de DEC. Todos los sistemas derivados de las entregas de BSD componen la "rama BSD" del árbol Unix. La otra gran rama (normalmente llamada Sistema V) deriva más directamente del código hecho en AT&T[8]

> **Comment (jgb):** *The history of BSD should be documented, including some references. The existence of BSD code in the System V release should also be documented, and if possible, some reference should be included.*

Afortunadamente, a principios de los años 1990 el CSRG hizo varias entregas que afirmaban que no incluian código propietario. Después de algunos litigios con los dueños de la licencia Unix se llegó a un acuerdo con al entrega de BSD-Lite, que fue reconocida como completamente libre de código propietario. Esa entrega fue wel origen de NetBSD[9], FreeBSD[10] y OpenBSD[11]. La entrega de BSD-Lite fue el último trabajo hecho por el CSRG antes de desaparecer.

La licencia BSD es un buen ejemplo de una licencia "premisiva', que casi no impone condiciones sobre lo que un usuario puede hacer con el software, incluyendo cobrar a los clientes por distribuciones binarias, sin la obligación de incluir el código fuente. Probablemente esta licencia, además de la excelencia técnica del software, fue una de las razones principales para su uso en tantos sistemas propietarios derivados de Unix durante los años 1980. Los principales puntos que establece la licencia son:

- Se permite la redistribución, uso y modificación del software.
- Las distribuciones deben incluir copias literales de la licencia, anuncio de copyright y una "negación de responsabilidad" (*disclaimer*).
- Debe incluirse reconoonocimeinto del origen del software (la Universidad de California) en cualquier anuncio.

Resuminedo, los redistribuidores pueden hacer casi cualquier cosa con el software, incluyendo usarlo para productos propietarios. Los autores sólo quieren que su trabajo sea reconocido. En cierto sentido, esta restricción asegura un cierto grado de "mercadeo (*marketing*) gratis". Es importante darse cuenta que este tipo de licencia no incluye ninguna restricción orientada a garantizar que los trabajos derivados sigan siendo libres. De hecho, ya se ha mencionado anteriormente cómo muchos sistemas operativos derivados de versiones de BSD han sido distribuidos como software no libre. Puede argumentarse que esta licencia asegura "verdadero software libre", en el sentido que el usuario tiene libertad ilimitada con respecto al software, y puede decidir incluso redistribuirlo como no libre. Otras opiniones están más orientadas a destacar que este tipo de licencia no contribuye al desarrollo de más software libre, incluso cuando es básicamente la versión libre original redustribuida como un producto propietario.

En el anexo **??** se incluye una transcirpción literal de la licencia BSD. Otras licencias similares son las que cubren el Sistema X Window X11R6 (incluida en el anexo C.6), XFree86 (en el anexo C.7) y Tcl/Tk (anexo C.8).

2.5.2 GPL (Licencia Pública de GNU)

La Licencia Pública de GNU (GPL, según las iniciales de su nombre en inglés, *General Public License*) es la licencia bajo la cuaol se distribuye el software del proyecto GNU. Sin embargo, hoy día pueden encontrarse toneladas de software no relacionado con el proyecto GNU pero distribuido bajo la GPL (un ejemplo notable es es el kernel Linux). La GPL se diseñó cuidadosamente para promover la producción de más software libre, y por

[8] La rama de Sistema V también incluye código de BSD, y cuando se estudia en detalle, hay mucha fertilización entre ambas rama.

[9] `http://www.netbsd.org`

[10] `http://www.freebsd.org`

[11] `http://www.openbsd.org`

ello prohibe explícitamente algunas acciones sobre el software que podrían llevar a la integración de software protegido por la GPL en programas propietarios. La GPL usa como base legal la legislación sobre copyright, haciendo de esa forma un uso muy interesante de ella, ya que se usa el copyright para promover la distribución de software que garantiza mucha más libertad a los usuarios que los trabajos habitualmente protegidos por coyright. Por lo tanto, algunas vecesse dice que el software cubierto por la GPL está "copylefted", un interesante juego de palabras en inglés[12].

Las principales características de la GPL son las siguientes:

- Permite la redistribución binaria, pero sólo si se garantiza también la disponibilidad del código fuente. Esta disponibilidad puede garantizarse bien mediante su distribución simultánea, o mediante el compromiso de distribución a solicitud del que recibe la versión binaria.
- Permite la redistribución fuente (y obliga a ella en caso de distribución binaria).
- Permite las modificaciones sin restricciones (siempre que el trabajo derivado quede también cubierto por la GPL).
- La integración completa sólo es posible con software cubierto por la GPL.

Comment (jgb): *This should be completed and probably rewritten. Some mention of the "tainting" characteristics of the license.*

2.5.3 LGPL (GPL para bibliotecas)

La Licencia Pública de GNU para Bibliotecas (LGPL, según las iniciales de su nombre en inglés, *Library General Public License*) fue diseñada por la Free Software Foundation para proteger las bibliotecas que se estaban desarrollando en el proyecto GNU, pero que quería que pudieran ser enlazadas con programas propietarios.

- Bibliotecas del proyecto GNU (y mucho más).
- Diseñada para permitir el uso de bibliotecas libres con software propietario (por ejemplo, en el caso de un compilador libre).
- Como la GPL cuando se redistribuye la biblioteca como tal.
- Permite la integración con cualquier otro software. En este caso, no hay prácticamente limitaciones.

Comment (jgb): *This should be completed and probably rewritten.*

2.5.4 Otras licencias de software libre

- Artistic (Perl, similar a la de BSD).
- Aladdin (Aladdin Ghostscript, en caso de redistribución con ánimo de lucro, las condiciones son similares a las del software propietario).

  Comment (jgb): *Can this be considered a libre license?*
- NPL (Licencia Pública de Netscape, incluye ciertos privilegios para el "primer autor").

Comment (jgb): *This should be completed (more licenses, more detail) and documented, including verbatim licenses as appendix.*

2.5.5 Licencias de software no libre

Comment (jgb): *Study of several proprietary licenses. For instance, CorelDraw. We should "harvest" several of them. "Right of use" (proprietary software) against "code disposal" (free software).*

[12] "Left" en inglés es tanto "dejar" como "izquierda".

Capítulo 3

Historia del software libre

Comment (jgb): *We need to document these paragraphs. What is stated in them could be not the truth. Talk to rms, ask in mailing lists (perhaps foro and fsb). Include teh material about IBM which Carmen sent.*

Comment (pheras,jgb): *Look in the Computer Museum at Cambridge, and in the Museum of Retrocomputing (are both the same thing?)*

Comment (pheras): *History of free software and Unix and Internet: Netizens book*

Aunque todas las historias relacionadas con la informática son forzosamente breves, la del software libre es una de las más largas entre ellas. De hecho, podría decirse que el software nació libre, y permaneció así durante su infancia. Sin embargo, con la llegada de su juventud la situación cambió completamente. Sólo ahora, al llegar a su madurez, está en vías de recuperar la libertad. Esto no deja de ser curioso, pues para gran parte de los profesionales informáticos, el software propietario es el software "en su estado natural". Afortunadamente, la situación es más bien la contraria...

Comment (jgb): *We should talk about Spice and TeX/LaTeX somewhere. Both are interesting examples of communities where collaboration and sharing of code was the rule.*

3.1 Cuando sólo había software libre

3.2 El comienzo: BSD, GNU

3.2.1 Richard Stallman, GNU, FSF

- Fundamentos "legales" (GPL) y filosóficos.
- Infraestructura básica: editor (Emacs), compilador (GCC), depurador (GDB), etc.
- Meta: construir un sistema completo alternativo a Unix.
- Trabajo muy estructurado y con metas claras.

Comment (jgb): *Complete and rewrite the whole subsection*

3.2.2 Desarrollo temprano de Unix

Comment (jgb): *This section should be completely rewritten*

[3], page 139:

> *"Since the Unix programming environment was used by the researchers themselves, they came to learn its weaknesses and problems and were encouraged to make needed improvements. Contributing to the value of Unix during its early development was the fact that the source code was open and available. It could be examined, improved, and customized."*

Comment (jgb): *Unix was, to some extent, an early test of what was seen with GNU and Linux several years later. It was confined to a smaller community, and an ATT license was required, but in other respects its development was similar... (in a world quite less communicated).*

[3], page 142:

"Pioneers such as Henry Spencer agree on how important it was to those in the Unix community to have the source code. He notes how having the sources made it possible to identify and fix the bugs that they discovered. "There is something the UNIX community has always been fairly strong on", he explained during an interview, "admitting things you know don't work about the software". Even in the late 1970s and early 1980s, remembers Spencer, "practically every UNIX site had complete sources" ."

[7]:

"That, by the way, was one of the great things about UNIX in the early days: people actually shared each other's stuff. [...] Not only did we learn a lot in the old days from sharing material, but we also never had to worry about how things really worked because we always could go read the source."

3.2.3 El CSRG de Berkeley

- Importancia de compartir software (cultura Unix "original").
- Coartado por la licencia AT&T (poco desde el punto de vista práctico, todos la tenían).
- Énfasis en el sistema operativo (kernel, utilidades, etc.)
- Utilizado por mucho software propietario (SunOS, Ultrix, etc.)

Comment (jgb): *Complete and rewrite the whole subsection*

3.2.4 Los comienzos de Internet

Comment (jgb): *This section should be completely rewritten*

Desde el comienzo de Internet uno de los aspectos clave de su desarrollo fue la libre disposición de todas las especificaciones y documentos de diseño, incluyendo las normas que definen los protocolos.

[3], page 106:

This open process encouraged and led to the exchange of information. Technical development is only succesful when information is allowed to flow freely and easily between the parties involved. Encouraging participation is the main principle that made the development of the Net possible.

[3], page 118 (NCP is the predecessor of TCP/IP):

"NCP was developed in the field, and versions of it were released early in its development so various programmers could work on implementing and improving the protocol. In addition, all specifications were free and early available for people to examine and comment on. Through this principle of early release, problems and kinks were found and worked out in a timely manner."

[1]

"The content of a NWG note may be any thought, suggestion, etc. related to the HOST software or other aspect of the network. Notes are encouraged to be timely rather than polished. Philosophical positions without examples or other specifics, spoecific suggestions or implementation techniques without introductory or background explication, and explicit questions without any attempted answers are all accpetable."

[6], page 267:

"Since the ARPANET was a public project connecting many major universities and research institutions, the implementation and performance details were widely published"

Comment (jgb): *Other aspects to develop:*

- Implementaciones de referencia, disponibles para todos.
- La Red como herramienta de cooperación (News, ftp, correo-e).
- La comunidad de usuarios proporciona el mejor soporte.

Comment (jgb): *Complete and rewrite the whole subsection*

3.3 Todo en marcha

Durante los años ochenta y primeros noventa, el software libre sigue desarrollándose, primero en varios grupos relativamente aislados. Pero poco a poco, mucho del software desarrollado se va integrando, uniendo de hecho el trabajo de muchos de estos grupos.

- Entornos completos sobre Unix (SunOS, Solaris, etc.):
  - Muchas aplicaciones son las mejores en su campo (utilidades Unix, compiladores, etc.).
  - Especialmente interesante: X Window.
  - Sólo falta el kernel...
- 386BSD, NetBSD, FreeBSD, OpenBSD:
  - Bill Jolitz termina las partes que faltaban para completar un kernel BSD libre.
  - Rápidamente: sistemas completos, similares a SunOS en funcionalidad.
  - Licencia BSD (puede redistribuirse como software propietario).
- GNU/Linux:
  - Linus Torvalds hace su "Minix libre".
  - Cientos de desarrolladores se vuelcan sobre él, e integran todo el software de GNU alrededor.
  - Gran explosión de adaptación de aplicaciones, nuevas aplicaciones, etc.
  - Casi todo bajo GPL: sólo puede redistribuirse con código fuente.
  - Un kernel, muchas distribuciones (Slackware, Debian, RedHat, Suse, etc.).
  - Gran éxito "popular".

Comment (jgb): *It would be good, specially for GNU/Linux and FreeBSd, to have some data about number of users, programmers, companies, presence in market segments, etc.*

3.4 Tiempos excitantes

¿Estamos ante un nuevo modelo de industria software?

Comment (jgb): *References and more details albout all these sections are still missing*

- Anuncio de Netscape.
- GNU/Linux y FreeBSD compiten con Windows-NT.
- GNU/Linux penetra en Universidades (y en casa de los estudiantes).
- En muchos ámbitos, la mejor opción es libre (Apache, infraestructura de Internet, XFree86, GCC, GNAT).
- La prensa comienza a atender al software libre.

3.5 El futuro: ¿la consolidación?

Capítulo 4

Características del software libre

4.1 Coste

- Adquisición gratis, pero hay otros costes:
  - Aprendizaje.
  - Infraestructura.
  - Soporte y mantenimiento.
  - Instalación.

  Coste de distribución muy bajo: Internet, CD-ROM.

- Desarrollo y mantenimiento más baratos.

El hecho de que un software pueda adquirirse gratuitamente no implica que sus autores no perciban ingresos que les permitan dedicarse a desarrollar más software. Habrá que buscar fuentes de financiación que no dependan del precio de la redistribución. Más adelante se discutirán algunas de estas fuentes, y cómo puede conseguirse un modelo de desarrollo software viable basado en ellas.

4.2 Consecuencias de la disposición del código fuente

- Más reutilizabilidad, sin necesidad de genéricos, herencia, etc.
- Más transportabilidad. Especialmente bueno para hardware nuevo, antiguo o minoritario.
- Adaptabilidad a necesidades y entorno locales.
- Más transparente. Bueno para educación, comprensión, y creación de nuevos programas).
- Evolución ágil.
- La calidad es evidente por inspección. Quizás nunca usaríamos algún software propietario si pudiésemos ver su código fuente.
- Depurable por usuario.
- El cliente puede ayudar en el desarrollo.
- Mayor fiabilidad. Por ejemplo, el problema del año 2.000 casi no es un problema si el código fuente está disponible.
- Mayor seguridad. Muchos ojos pueden mirar el código.
- Fijación sencilla de estándares. Las realizaciones de referencia lo hacen más fácil.
- Los programas interpretados sólo pueden ser de código abierto.

- Buen soporte. Por ejemplo, GNU/Linux fue elegido por InfoWorld's en 1997 como "Mejor Producto del Año")

  Comment (jgb): *These characteristics should be revised and completed.*

4.3 El software como servicio

4.4 Reutilización e integración

4.5 Software multiplataforma

4.6 Desarrollo en plataformas hardware minoritarias

4.7 Disponibilidad de software para la comunidad

4.8 Oportunidades de desarrollo para pequeñas empresas

4.9 Innovación: nuevas aplicaciones basadas en código existente

4.10 Internet y el software libre

Comment (jgb): *The role of Internet in the development of free software, and its future*

4.11 Software libre frente a propietario

Comment (jgb): *Advantages of libre software with respect to proprietary software*

4.12 ¿Cómo puede haber software libre de calidad?

Comment (jgb): *Motivation: quality libre software do exist, that is a fact. How can that be achieved without following the "classical" proprietary model?*

4.13 Diversos modelos de soporte

El software libre permite muchos modelos de soporte, cada uno mejor adaptada a un tipo de software. Y muchos de estos modalelos no son posibles (en general) para el software propietario.

4.13.1 Soporte por parte de los propios usuarios

Es posiblemente al única forma viable de soporte en muchos casos. Por ejemplo, cuando el software es relativamente sencillo y de poca "entidad". O cuando el software está destinado a usuarios muy entendidos y acostumbrados a sacarle el máximo partido, incluso modificándolo si es preciso (como suele ser el caso de los administradores de sistemas informáticos).

Comment (jgb): *Importance of user groups*

4.13.2 Soporte por los propios desarrolladores

Es el modelo habitual en muchos proyectos de software libre de cierta entidad, y en general, el más clásico en el mundo del software propietario. Este soporte puede realizarse cobrando por él, o de forma gratuita, con la idea de mejorar el propio software.

Ejemplos:

- Ada Core Technologies[1] (ACT).

  Comment (jgb): *More examples should be included, both of developers charging for support and of those not charging*

4.13.3 Soporte por una empresa

Es el modelo tradicional de soporte para grandes proyectos. Se contrata a una empresa para el mantenimeinto de todo el software que interviene en el proyecto, y que puede ser quien lo diseñó y construyó o no. En el caso del software propietario, esto supone que el cliente haya comprado también el código fuente, o que el mantenimiento lo haga quien lo construyó, ya que el mantenimiento es absolutamente imposible sin él.

En el caso del software libre, este modelo es especialmente sencillo. Como el código fuente está disponible siempre, cualquier empresa puede especializarse en el mantenimeinto de unos cuantos programas, y llegar a conocerlos tan bien como sus autores originales. Al mismo tiempo esto introduce la posibilidad de competencia. De hecho, el permitir que varias empresas puedan competir por el mantenimiento en igualdad de condiciones hace posible que la comptencia actual, que se limita a la construcción de software, se amplíe al mantenimineto.

4.13.4 Pruebas y corrección de erratas

Los mecanismos de detección y corrección de erratas merecen un análisis detallado. Cuando un usuario detecta un problema en un programa propietario, sólo tiene dos posibilidades:

- Aguantarse, e intentar evitar el patrón de comportamiento que produce la errata, si lo encuentra.
- Informar a los fabricantes del software sobre la errata, en la esperanza de que lo arregle en una futura entrega.

El usuario sólo recibirá (en el caso general) una versión nueva con cierta rapidez si tiene un contrato de mantenimiento que lo especifique. Esta situación es muy poco frecuente. En el caso de que no haya contrato de mantenimiento, el fabricante estará muy poco motivado a enviar a sus clientes nuevas versiones gratuitas con la errata corregida. No está obligado legalmente a ello, y en la mayoría de los casos (especialmente en el mercado masivo) sus ingresos vienen de ventas nuevas. Por ello le será mucho más rentable vender a sus usuarios, unos años después una "nueva" entrega, que entre otras mejoras puede que tenga erratas corregidas.

Nótese que con esta situación el usuario tiene poca motivación incluso para informar de las erratas al fabricante. Generalmente no se va a beneficiar directamente de ello, y en muchísimos casos, ni siquiera indirectamente. Si su errata se corrige, quizás va a tener que pagar por un producto nuevo para poder usar una versión sin esa errata.

En el caso del software libre, la situación es muy diferente. En primer lugar, los usuarios están en general más motivados a informar sobre las erratas, e incluso a corregirlas ellos mismos y enviar un parche a quien mantiene el software. En muchos casos, incluso se organizan grupos de usuarios que mantienen archivos de informes de erratas y parches que las solucionen. Además del sentimiento de contribuir a un proyecto común, en la mayoría de los casos tienen la seguridad de que si su errata se corrige, la solución se incluirá en la próxima entrega del programa, que podrán obtener a bajo coste (normalmente, a coste de copia).

Los desarrolaldores y mantenedores también estarán muy motivados a ayudar a los usuarios a informar de erratas. En primer lugar, puede que no sólo encuentren erratas, sino que también sepan (y quieran) corregirlas. En ese caso se obtiene el beneficio bien concreto de que alguien te diga cómo mejorar tu programa. También ayudan en el mecanismo general de detección de erratas, muy importante en cualquier desarrollo de software de calidad. Y en cuanto hagan una nueva entrega, no hay ningún motivo para que no incluyan todos los parches de corrección de erratas que conozcan. Como sus ingresos no proceden de la venta del producto, tampoco hay motivos para que tarden mucho en hacer la nueva versión, y en general estarán bastante motivados a ello.

Comment (jgb): *Revise and complete these last paragraphs. Perhaps a comparation about how bugs are deal with in Debian and Win* could be a good idea?*

4.14 Impacto en la enseñanza

Comment (jgb): *To be written*

[1] http://www.gnat.com

Capítulo 5

Impacto social y económico

5.1 Implicaciones económicas

5.1.1 Competitividad

- Coste cero.
- Libertad.
- Facilidad de prueba.
- Transportes a otros sistemas aumentan el mercado.
- Facilita el desarrollo para nuevo hardware.
- La distribución es gratis. El desarrollo y el mantenimiento son baratos.
- ¿Hay costes de comercialización? El boca a oreja funciona muy bien en la Red. Los grupos de usuarios ayudan mucho.
- La industria informática parece tender al monopolio
  - Años 1980: elevados márgenes, alta demanda.
  - Años 1990: bajada de precios, márgenes escasos.
  - ¿Puede derrotarse al monopolio cooperativamente?

5.1.2 Retribución a los autores

¿Pueden cobrarse los programas que se distribuyen con código fuente?

- Como los programas “normales”, que se distribuyen sólo como binarios.
- La práctica común ha sido distribuirlos a precio desorbitante.
- Pueden copiarse igual que los binarios.
- Es imposible usar protecciones anticopia.
- El código fuente descubre la tecnología usada.
- Es fácil probar judicialmente la copia, pero es difícil iniciar la demanda si el demandado oculta sus fuentes.

5.1.3 Financiación de nuevos desarrollos

- Empresas de hardware: manejadores, sistema operativo, etc.
  - El software es un coste necesario, imprescindible para vender.
  - El software es muy caro de desarrollar.
  - La copia suele ser inútil sin el hardware.
  - El fabricante puede realizar contribuciones propias o pagar por el transporte (ej: GCC, Linux).

  Puede intentarse cobrar por copia.

- Hacer algún programa libre para promocionarse.
- Vivir de la distribución, mantenimiento y/o soporte. Esto podría dar lugar a programas oscuros y mal documentados, para aumentar la motivación de los usuarios a comprar soporte.
- Financiación por empresas con necesidades de cierto software.
- Tener una versión "para uso comercial", que se distribuye de forma similar al software propietario.
- Dinero de impuestos, especialmente interesante para comenzar nuevos proyectos.
  - Práctica habitual en USA (Universidades, FSF, empresas).

    Comment (jgb): *This should be documented or removed*
  - La Unión Europea está financiando programas propietarios con los impuestos de todos.

    Comment (jgb): *This should be documented or removed*
- Asociaciones.
- Voluntarios (retribución en autoestima o prestigio).

También hay otras oportunidades de conseguir dinero para el desarrollo de software libre:

- Distribución con soporte (Caldera, RedHat, Cygnus).
- Venta de máquinas preinstaladas (VA Research).
- Accesorios (libros, hardware, camisetas).

  Comment (jgb): *The classes of open source business, as discussed by Eric S. Raymond, could be included and discussed here, including the critics by some people.*

5.1.4 Empleo

¿Perderán el empleo los programadores?

Podría pensarse que si el fenómeno del software libre se extendiera mucho, habría miles de programadores que podrían perder su trabajo. Al no generarse tantos beneficios marginales como en el software propietario, también habrá menos dinero para pagar a programadores. Esto podría tener no sólo implicaciones sobre el empleo, sino también sobre la propia cantidad de software producido.

Sin embargo, puede argumentarse en contra que si un programa no existe, pero es necesario (alguien siente que es conveniente tenerlo), es muy probable que alguien se decida a pagar para hacerlo, sea libre o no. La propia necesidad de tener nuevos programas debería, por si sola, promover nuevos proyectos, al menos mientras haya clientes que los precisen. Y por último, en general lo más costoso del desarrollo de un programa es su mantenimiento, y eso varía poco según el programa sea libre o no.

> **Comment (jgb):** *This section should be completely rewritten and documented. Something could also be said about the better economical efficiency got from libre software, thanks to competition, and to the diffculty of monopolistic situations*

5.2 El delito informático y el software libre

5.3 Ventajas para los no informáticos

Puede que los no informáticos no perciban muchas de las ventajas del software libre. Pero como mínimo, notarán claramente algunas mejoras:

- Pueden copiar con tranquilidad los programas para su propio uso.
- Pueden regalar o prestar los programas sin ninguna restricción (excepto, qui'zas, el compromiso de proporcionar el código fuente).

5.4 Libertad vs monopolios comerciales

5.5 ¿Hacia un nuevo concepto de propiedad intelectual?

5.6 Posibilidades de desarrollo con pocos recursos

5.7 Comparativa con otros sectores

Comment (jgb): *Economists or economy historians could help a lot here*

5.7.1 Industrias que han cambiado por completo el proceso productivo

Son varias las industrias donde el proceso productivo, y los mecanismos de distribución, han cambiado por completo. Por ejemplo:

- Cadenas de montaje, a principios de siglo: el Ford T y la industria del automóvil.
- La imprenta y el modelo de producción y distribución de libros. Es especialmente interesante su impacto sobre la sociedad.
- Mecanización agrícola.

5.8 Legislación aplicable

5.8.1 Copia ilegal de información

5.8.2 Propiedad intelectual

Comment (jgb): *We could talk here about the problems when a proprietary software uses, without the permission of its author, some piece of libre software, which license forbids it (for instance, GPL). Some cases have been mentioned in fsb (July 1998), could be used for documentation.*

Capítulo 6

Modelos de desarrollo software

No hay un solo modelo de desarrollo de software libre, igual que no hay un solo modelo de desarrollo de software propietario. De hecho, continuamente se están experimentando nuevas ideas, que tratan de aprovechar alguna característica de algún programa en cuestión, de la comunidad a la que va dirigido, o del propio hecho de ser software libre. Pero podríamos hablar de dos modelos extremos. Los describe con bastante detalle Eric S. Raymond en "The Cathedral and the Bazaar".

> **Comment (jgb):** *The Cathedral and... should be cited as bilbliography*

6.1 El modelo "catedral"

En este modelo, el desarrollo está muy centralizado, y hay pocas entregas. Es un modelo muy habitual en el software propietario. Los proyectos GNU y NetBSD, por ejemplo, utilizan en gran medida este modelo (mejorado con contribuciones ajenas).

> **Comment (jgb):** *This section should be carefully improved and completed*

6.2 El modelo "bazar"

Este es un modelo que fomenta muy activamente la colaboración. Habitualmente se hacen entregas muy frecuentemente. El kernel Linux, por ejemplo, utiliza en gran medida este modelo. Sus principales características son las siguientes:

> **Comment (pheras, jgb):** *Does the Linux project really use this model? Perhaps a better example should be used. Perhaps Apache...*

- En muchos, el comienzo del proyecto tiene lugar debido a una necesidad o interés personal del desarrollador. Este fue el caso, por ejemplo, de Unix, Linux, LaTeX, PHP y Apache.
- Mucho énfasis en reutilizar, mejorar o adaptar otro programa, hasta llegar a reescribirlo completamente, con el tiempo. Este ha sido, al menos en parte, el proceso que ha llevado de NCSA httpd a Apache, por ejemplo.
- Es muy importante conseguir una cierta población de usuarios, y tratarlos como codesarrolladores. Estos usuarios estarán intresados en resolver los problemas que aparezcan, pues se beneficiarán directamente de ello. El autor puede discutir con ellos, y como usuarios experimentados que pueden ser, es muy posible que tengan ideas geniales. Para establecer una buena relación con ellos es muy importrante motivarles y reconocer su trabajo. Y ponerles fácil el contribuir al proyecto. Por ejemplo, los requisitos de la FSF (garantizar GPL para todo fragmento de código de más de 20 líneas) puede desmotivar a algunos usuarios a contribuir.

  > **Comment (jgb):** *Document the FSF requisites*

- Se hacen entregas desde etapas muy tempranas del desarrollo, y con mucha frecuencia. De esta forma, los usuarios pueden colaborar en al depuración de programa, y las erratas van encontrándose (y corrigiéndose) continuamente. Hay que tener en cuenta que el trabajo de depuración es uno de los más costosos del desarrollo software. Además, la colaboración de los usuarios permite la "depuración" en paralelo, donde varios usuarios pueden encontrar (y quizás corregir) la misma errata. En este caso, el autor puede elegir la mejor corrección, o en el peor de los casos, disponer de mucha información sobre la errata. Si la información sobre las erratas es pública, puede ocurrir que un usuario descubra una, y otro vea claramente la forma de corregirla y publique una solución. Si los usuarios ven que sus soluciones son incorporadas a las entregas "oficiales", habitualmente quedan motivados para repetir el proceso, y se involucran más en el proyecto. Por otro lado, para evitar que usuarios "normales" se encuentren con versiones que tengan muchas erratas, pueden publicarse dos familias de entregas: las estables (cuidadosamente revisadas) y las de desarrollo (que habitualmente incorpora características más avanzadas, pero sólo para usuarios con ganas de enfrentarse a problemas).

  Comment (jgb): *This could be documented with the number of bug reports and bug corrections to some projects during some period of time*

- Es importante comenzar con algo utilizable.
- La coordinación también es importante. Puede realizarse al menos de dos formas:
  - Una "meritocracia técnica" compone el "núcleo" del proyecto. Se accede a ella mediante el reconocimiento de los pares. Es el caso, por ejemplo, de Apache o NetBSD.
  - El proyecto se estructura alrededor de un líder. En este caso, es muy importante que éste tenga habilidades para coordinar y motivar a los demás desarrolladores, y para darse cuenta de cuándo ha perdido su interés. En ese momento, debería pasar el liderazgo a otro que esté suficientemente interesado en el proyecto. Esta forma de coordinación es también jerarquizable, cuando el líder "principal" delega áreas completas a otros desarrolladores, que actúan como líderes dentro de ellas. Este es, por ejemplo, el modelo de desarrollo del kernel Linux.

Comment (jgb): *This section should be carefully improved and completed*

6.3 ¿Estaba Brooks equivocado?

Comment (jgb): *Discussion about the Brooks' book, "The Mythical Man-month", and its apparent inaplicability in cases like GNU/Linux. Include the book in bibliography*

Capítulo 7

Modelos de negocio basado en el software libre

Comment (jgb): *Maybe this chapter can be merged with the next one, about free soft companies*

Cuando se estudian los modelos de negocio relacionados con el software libre, parecen ser muy diferentes de los modelos de software propietario. De hecho, mucha gente no ve ningún negocio si no se pueden cobrar tarifas por copia, que es lo habitual en modelos propietarios. En este capítulo trataremos de mostrar algunos modelos posibles para empresas que quieran estar en el negocio del software libre. Todos ellos son "puros", en el sentido que hemos tratado de categorizar como modelos diferentes lo que en la realidad se encuentra normalmente como mezclas muy variables. Esperamos que esos modelos puedan ayudar a entender cómo una empresa o un individuo pueden sobrevivir saludablemente alrededor del software libre. Algunos de ellos son sólo modelos tradicionales, casi idénticos a los que se usan con el software propietario, algunos de ellos incluyen notables características nuevas debidas al uso de software libre y algunos otros son completamente nuevos y únicos.

7.1 Consultoría y asesoría

Este tipo de actividad trata de proporcionar soporte a compañías que están tratando de decidir si deberían moverse y usar software libre, o cómo puede usarse software libre con ventaja para sus negocios. Algunos ejemplos son:

- El sistema informático de una empresa dada está basado en SunOS os algún otro sistema Unix propietario, y quieren explorar la posibilidad de moverse parcial o completamente a GNU/Linux-SPARC. El tipo de tareas a realizar en este caso podría ser el siguiente: análisis de la madurez y estabilidad de GNU/Linux-SPARC, estudio de la disponibilidad en la plataforma propuesta de las aplicaciones que use la empresa, evaluación de los costes de transporte para las aplicaciones hechas en al propia empresa, de los costes de aprendizaje para usuarios y adminsitradores de sistemas, ahorros debidos al uso del modelo de software libre, análisis de rendimiento y disponibilidad y calidad de contratos de soporte.
- Una empresa quiere identificar dónde puede usar software libre en su sistema informático con ventajas claras. El tipo de tareas en este caso podrían ser: análisis del software usado por la empresa, identificación de programas libres equivalentes y comparación (en cuanto a coste, calidad, rendimiento, soporte) de ambas opciones.

7.2 Venta de productos hardware/software o software

Comment (jgb): *Needs a lot of ellaboration*

- Venta de hardware con algún valor añadido. Por ejemplo, "cajas de comunicaciones" para conectar pequeñas empresas a Internet o a sus sucursales remotas. Pueden estar basadas en GNU/Linux o FreeBSD.

- Venta de software libre. Por ejemplo, este es (hasta cierto punto) el negocio principal de RedHat, que vende su propia distribución de GNU/Linux. Otro ejemplo puede ser la venta de versiones "reducidas" de Debian a empresas que quieran pagar por un GNU/Linux con aplicaciones ofimáticas. O la venta de CDs con juegos de ordenador libres, basados en un entorno GNU/Linux simple.

7.3 Desarrollo de software

Hay dos modelos principales de desarrollo de software libre, según quién decide qué debe ser desarrollado:

- Desarrollo "interno". La empresa desarrolladora decide qué desarrollar. Normalmente, es un programa que consideran interesante. A veces, ellos no desarrollaron el programa por si mismos, sino que lo "adoptaron".

  Comment (jgb): *This can be written with much more detail. GIMP or Kaffe can be used as examples of "internal" development. Gygnus can be used as an example of "adopting" programs, with egcs. Other example of "adoption" could be getting Apache and "adopting" it, developing glue software for electronic commerce.*

- Desarrollo "externo".

  Comment (jgb): *This has to be elaborated. Specially, the diferences between external and internal are not clear when an "internal" development gets money from a third partie interested in some new feature. Examples of "exeternal" development: making drivers for the Linux kernel, porting applications to a free operating system. These are not that different from "usual", non free development, except for that there can be several developers, since the software is not owned by a company.*

7.4 Soporte, instalación, integración

Comment (jgb): *Is this the Caldera model?*

7.5 Red de conocimiento y experiencia

7.6 Educación y entrenamiento

7.7 Otras actividades

Hay otras actividades que puede mantener una empresa relacionada con el software libre. Algunas de ellas pueden de hecho ser la base de modelos de negocio de software libre, pero preferimos verlas como "actividades colaterales", de las que al compañía puede obtener algún beneficio, aunque usualmente no de forma directa.

7.7.1 Promoción de software libre

Ejemplo: danr dinero a algún proyecto de software libre, como hace RedHat con el Proyecto Gnome.

Ejemplo: dar dinero o dedicar recursos para traducir documentación e internacionalizar programas. Por ejemplo, dar dinero a la gente que está "traduciendo" Linux al castellano.

Comment (jgb): *Complete this subsection, and document the RedHat and Gnome case, and perhaps some others.*

Capítulo 8

Empresas comprometidas con el software libre

Comment (pheras): *Probably better if we split it in cathegories: creators of libre software (ACT, Cygnus), libre software consummers (Sun?, IBM, Corel) and sellers of libre software (Walnut Creek)*

Comment (jgb): *Another point of view could be to identify "patterns", and to illustrate them with a list of companies. For isntance: use of libre software to build proprietary products, illustrated with BSD as the origin of SunOS and Ultrix, etc.*

8.1 Cygnus Support

Comment (jgb): *The first ones. Support to large companies. Development (e.g. porting to MS-Windows, egcc.*

8.2 PTF

Comment (jgb): *First ones in exploring low-priced redistribution of libre software.*

8.3 Aladdin

Comment (jgb): *Versions of Ghostscript.*

8.4 Ada Core Technologies

Comment (jgb): *They are the best ones in their niche, centered around the support to software development in Ada. They obtain their financing mainly from the support to large companies. They maintain probably the best current Ada compiler, which is distributed under GPL*

Comment (jgb): *Comments to their way of "distributing with different releases"*

8.5 Netscape

Comment (jgb): *Solving internal and market problems? Competitive advantage? Only exit?*

8.6 Corel

Comment (jgb): *Network Computers based in Linux. They consider that software is not their business.*

8.7 Caldera

Comment (jgb): *Integration fo GNU/Linux with proprietary software, and sell of support.*

8.8 RedHat

Comment (jgb): *Simplicity and good integration for GNU/Linux*

8.9 Texas Instruments, Motorola

Comment (jgb): *Clients of Cygnus, to port GCC to their processors*

8.10 MetroLink

Comment (jgb): *???*

8.11 Infomagic

8.12 Walnut Creek

8.13 Sun

Comment (jgb): *Support to Tcl/Tk, use of BSD for SunOS, etc.*

Capítulo 9

Proyectos importantes de software libre

9.1 GNU

9.2 La familia BSD

9.3 Linux

9.4 Debian

9.5 KDE

9.6 Tcl/Tk

9.7 PERL

9.8 Apache

9.9 Tex/LaTeX

9.10 Python

9.11 XWindow, XFree86

9.12 Mozilla

`http://www.mozilla.org`

Capítulo 10

Organizaciones relacionadas con el software libre

10.1 Free Software Foundation

Comment (jgb): *First pushes, and they are still at it.*

10.2 Consorcio X

Comment (jgb): *They were thinking about leaving the way of libre software, but currently they seem have committed to go on with libre releases*

10.3 XFree

Comment (jgb): *Development of X Window for i386.*

10.4 FreeBSD

10.5 NetBSD

Comment (jgb): *Promotion and support for their Operating System.*

10.6 Debian

Comment (jgb): *Hundreds of developers integrating applications.*

10.7 Apache

Comment (jgb): *They maintain their HTTP server as the most used one.*

10.8 Opensource

`http://www.opensource.org`.

10.9 LPF

10.10 Organizaciones relacioandas con la información libre en general

- Electronic Frontier Foundation[1].
- Netaction[2].
- CPSR[3].

[1] http://www.eff.org
[2] http://www.netaction.org
[3] http://www.cpsr.org

Capítulo 11

El software libre en el mundo real

Comment (jgb): *Case studies fo use of libre software in the real world*

11.1 Software libre en entornos científicos y universitarios

11.1.1 Spice: análisis de circuitos

Spice es un programa de simulación de circuitos electrónicos. Fue diseñado a principios de los años 1970 en la Universidad de California en Berkeley, y la primera implementación fue hecha (con la colaboración de otros) por Larry Nagel, que en esa época era un estudiante dirigido por Donald Pederson. A finales de los 1990, Spice es reconocido como el programa estándar para la simulación de circuitos analógicos. Aunque Spice es software libre (bajo una licencia similar a la de GNU), varias empresas distribuyen versiones propietarias que incluyen interfaces de usuario gráficas y otras mejoras que hacen del programa una herramienta más práctica.

Desde un punto de vista interesado en el software libre (y en la información libre), Pederson es una persona interesante. Intencionadamente, no ha patentado ninguno de los trabajos que ha realizado durante su carrera. Está a favor de poner los resultados de la investigación en el dominio público, con el argumento de que la puesta en común libre de conocimiento ha sido clave en el desarrollo de su campo de investigación, la microelectrónica.

Los orígenes de Spice pueden trazarse hasta Cancer, un programa propietario hecho por un grupo de estudiantes, encabezados por Nagel, dirigidos por Ronald Rohrer, que en esa época era profesor de ingeniería en Berkeley. Cuando Roher abandonó Berkeley, Pederson (cuyo grupo había escrito ya algunos programas similares) negoció con Rohrer la puesta de Cancer en el común (*public domain*), lo que se hizo en 1972, con el nombre Spice. Pederson también eligió Spice como el programa que se usaría en sus clases de diseño de circuitos integrados. Por lo tanto, a principios de la década de 1970 muchos estudiantes usaron Spice en Berkeley, que era una de las mejores instituciones en la enseñanza de diseño de circuitos integrados. Esos estudiantes, cuando pasaron a la industria, llevaron y adaptaron Spice a los ordenadores disponibles (CDC 6400, IBM 360/370, DEC VAX 11/780, etc.). Ya en 1975, el uso de Spice estaba muy extendido. En 1984 MicroSim Corp. produjo P-Spice, un transporte de Spice a ordenadores de tipo PC, comenzando una saga de productos propietarios basados en el núcleo libre de Spice. Cuando Pederson se retiró, en 1991, unas 100.000 copias de Spice estaban usándose en universidades y en la industria, y unas 20 empresas proporcionaban productos propietarios basados en Spice. Según Pederson, Spice se ha usado para simular al menos aprtes de virtualmente todos los circuitos integrados diseñados en EEUU durante las décadas de 1980 y 1990.

En [5] puede encontrarse una breve biografía de Donald Pederson, y un resumen sobre Spice, con ocasión de su recepción de la Medalla de Honor del IEEE como “padre de Spice”. La información sobre Spice que se ofrece aquí está basada en ese artículo.

> **Comment (jgb):** *We should check exactly which lciense has Spice, and if some URL about it is available. Perhaps in Berkeley?*

11.2 Software libre para el profesional informático

11.3 Software libre para el usuario final

11.4 Software libre para la empresa

Comment (jgb): *Commercial feasibility*

11.5 Internet: un caso interesante

En muchos casos se oye la siguiente frase (especialmente cuando un ingeniero innovador intenta introducir software libre en una empresa que aún no lo ha hecho):

> **"El software libre no es utilizable en entornos de producción..."**

Pero hay un entorno especialmente interesante estos días...:

- Internet está cambiando al economía y la sociedad mundiales.
- El comercio, la educación, las relaciones personales, y una fracción considerable del PIB se mueven sobre Internet, cada vez más.
- ¿Cuántos millones de usuarios?. ¿Cuántos cientos de miles de empresas?

Sin embargo, el software libre está (en grandes cantidades) en el corazón de Internet:

- Las implementaciones de TCP/IP de muchos sistemas operativos están basadas en la de BSD (que posiblemente es la mejor implementación de TCP/IP, from a technical point of view).
- News: prácticamente toda USENET usa CNews, Inews o paquetes similares.
- DNS: En el corazón de los protocolos de Internet, el rey es BIND.
- Correo electrónico: Sendmail es la implementación de referencia para MTAs.
- WWW: Más del 50% de los servidores de WWW son Apache. El Communicator de Netscape ha sido liberado (Mozilla).

> **Comment (jgb):** *Document this (URL, dates, etc.)*

Volviendo a la frase al inicio de esta sección, que el lector decida: ¿Es Internet un "entorno de producción"?

11.6 Otros ejemplos de uso de software libre

- Muchísimos proveedores de Internet están basados en software libre.
- Uso en la NASA: sistemas de exploración espacial.
- Control de sistemas: Trenes de Buenos Aires.
- Encaminamiento de correo (US Mail).
- Uso de GNAT en sistemas embarcados (ESA, militares, etc.).
- Docencia: posiblemente la mejor plataforma es GNU/Linux (y muy usada).
- Apache como servidor WWW utilizado por IBM. Lo usa para venderlo como producto suyo y extenderlo.

  Comment (jgb): *References about this!!!*

- Administración española: más de 200 PCs GNU/Linux para sistema de registros.

  Comment (jgb): *I should check details on this...*

- Granja de PCs para "renderización" del Titanic.

  Comment (jgb): *¿Cómo se dice "render" en castellano?*

  Comment (jgb): *References*

Capítulo 12

¿Hacia un nuevo modelo de desarrollo software?

El software libre ya ha demostrado su viabilidad. Para muchas tareas, un programa libre es reconocido como una de las mejores opciones (y cada vez más veces, como la mejor). También se está avanzando en nuevos modelos que permitan la financiación del desarrollo y la innovación en el área del software libre. Muchas empresas y organizaciones han reconocido ya el software libre como una forma de mejorar su competitividad. Esto es especialmente importante si tenemos en cuenta la creciente importancia del sistema informático en el negocio de casi cualquier empresa.

Capítulo 13

Del software libre a la información libre

13.1 Bibliotecas electrónicas

- Proyecto Gutemberg[1].
- Bitblioteca (Venezuela).
- RFCs como "documentos abiertos".

  Comment (jgb): *The URL of the Bitblioteca is missing*

  Comment (jgb): *Info about RFCs in the book "Netizens" (add to bibliography)*

13.2 Música libre

13.3 Amenazas a la circulación libre de información

Comment (jgb): *This section would probably be better in other chapter. Or not?*

- El caso de los GIFs.
- Patentes de algoritmos.

  Comment (?): *Information about patents on algorithms in* `http://www.lpf.org`

13.4 Un caso interesante: Internet

El desarrollo de Internet debe mucho a la circulación libre de información y al software libre. Pueden destacarse los siguientes aspectos:

- Circulación libre de RFCs.
- Implementaciones de referencia: la importancia del software libre en el desarrollo de normas (estándares).
- Modelo de desarrollo de nuevos sistemas en Internet. Los casos del DNS, PPP, WWW, correo electrónico, News, gopher, MBone, IPv6, etc.
- ¿Podría cambiar este modelo en el futuro?

[1] `http://www.gutemberg.org`

Apéndice A

Algunos enlaces de interés

Comment (jgb): *Many of this URLs are still missing*

- "La catedral y el bazar" ("The Cathedral and the Bazaar"[1], en inglés), por Eric S. Raymond.
- "Por qué el software no debería tener dueños" ("Why Software Should Not Have Owners"[2], en inglés), por Richard Stallman.
- Documentos sobre software libre (Eric Raymond, Stallman, Bernard Lang). Hay varios de ellos (en inglés) en la Red[3].
- Publicaciones periódicas (Linux Focus[4], Linux Journal, GNU Bulletin, Apache Journal).
- Software:
  - Interfaces de usuario.
  - Sistemas operativos completos.
  - Ofimática: hojas de cálculo, procesadores de texto.
  - Compiladores y herramientas para desarrollo de programas.
  - Bases de datos.
  - Juegos.
  - Utilidades.
  - Aplicaciones científicas.
  - Internet: correo electrónico, WWW, News, editores de HTML.
  - Multimedia: GIMP, video, MBone.
- Software libre en la industria.
- Software libre en la Universidad.

[1] `http://www.openresources.com/documents/cathedral-bazaar`
[2] `http://www.openresources.com/documents/rms-why-free`
[3] `http://www.openresources.com/documents/`
[4] `http://www.linuxfocus.org`

Apéndice B

Glosario de términos

Cuando nos adentramos en el mundo del software libre, hay varios términos básicos que es conveniente conocer:

Software propietario (*proprietary software*) Software "con dueño", sobre el que los usuarios no tienen, en general, más derecho que el de uso, y a veces, ni ese es un derecho completo. Suele usarse para contraponerlo al software libre.

Open Source (software de fuente abierto) Término promovido por la Open Source Initiative[1] como una "marca" que pueda ser utilizada por los programas y licencias libres. Algunos miembros de la comunidad del software libre no están de acuerdo con este término porque evita cualquier referencia a la palabra "libertad", y por tanto (en su opinión), no menciona la principal característica del software libre, la libertad que proporciona a sus usuarios y desarrolladores.

> **Comment (jgb):** *Who is the original author of the term Open Source?*
>
> **Comment (jgb):** *The history and current situation of this term should be checked and completed*

Public Domain Software (software del común) El autor cede sus derechos a la comunidad.

Copylefted software (software bajo copyleft) Software distribuido bajo la GPL.

Piratas y otras novelas

Shareware

Freeware

Derechos de autor

Shrink wrap license (licencia de rompe y rasga)

> **Comment (jgb):** *"Pirates" is, obviously, an irony*

[1] `http://www.opensource.org`

Apéndice C

Licencias, definiciones, etc.

Comment (jgb): *Habría que traducir al castellano estas licencias, e incluir la traducción junto con la versión en inglés.*

C.1 Licencias propietarias

Con la idea de ofrecer material para la comparación, parece apropiado incluir aquí el texto de algunas licencias propietarias. En los siguientes apartados el lector puede encontrar algunas licencias que cubren no sólo software, sino también video, CDs de música, etc. Sin embargo, como la información propietaria es ubícua hoy día, animamos al lector a que busque en sus propias cintas de video, CDs, paquetes de software, libros, etc, y estudie las licencias del material que nos rodea en nuestra vida diaria.

> **Comment (jgb):** *Do you know of any interesting proprietary license? Please, let me know. I'd like to include some more illustrative example sin this section, specially of software packages or of new media (documents downloaded from the Internet, MP3, etc.).*

C.1.1 Licencias de un CDs de música

Casi todas las licencias y los anuncios de copyright de los CDs de música son muy similares. Como un ejemplo, esta es la licencia de mi copia de "The Piano", de Michael Nyman. El copyright de esa grabación de sonido es propiedad de Virgin Records Ltd.

> "All rights of the producer and of the owner of the recorded work reserved. Unauthorised copying, public performance, broadcasting, hiring or rental of this recording prohibited."

Ahora, veamos la licencia de mi copia de "Parsley, Sage, Rosemary and Thyme", de Simon and Garfunkel, producido por CBS. Este incluso prohibe la "reproducción pública" (*public performance*). Sólo una idea, pero esto podría significar que no lo puedes poner en tu fiesta de cumpleaños con tus amigos...

> "All rights of the producer and of the owner of the recorded work reserved. Unauthorised copying, public performance, broadcasting, hiring or rental of this recording prohibited."

Aún otra más. "The Commitments, music from the original motion picture soundtrack", producido por MCA. Esta licencia (en español en el original) prohibe el préstamo del CD. No puedes pasar el CD a un amigo durante un par de días...

> "Registrados todos los derechos del productor fonográfico y del propietario de la obra. Salvo autorización queda prohibida la publicación, alquiler, préstamo, ejecución pública o radifónica de este fonograma."

C.1.2 Licencias de cintas de video

Las licencias de las cintas de video son también muy similares. Por ejemplo, la siguiente viene con "Sense and Sensibility", con copyright de Columbia Pictures Industries. En este caso, no puedes prestar la cinta a un amigo, ni intercambiar cintas con él, ni siquiera mostrarla a un grupo de amigos (dependiendo del significado exacto de "shown" en la licencia).

> "The copyright owner has licensed the film contained on this video cassette (including its soundtrack) for domestic use only. All other right reserved. No part of this video cassette may be copied, edited or reduced, transformed, corrected, shown, hired out, exchanged, contracted, lent, broadcast, simultaneously projected in community videos and other communities and/or exhibited and publicly broadcast without express authorization and carrying out any of these activities would incur legal liabilities and may lead to prosecution."

C.1.3 Licencias de software

> ACUERDO DE LICENCIA DE CORELDRAW VERSIÓN 7 EN CASTELLANO
>
> ATENCIÓN: ESTO ES UNA LICENCIA, NO UNA VENTA. ESTE PRODUCTO SE PROVEE BAJO LA LICENCIA SIGUIENTE QUE DEFINE LO QUE USTED PUEDE HACER CON EL PRODUCTO, Y CONTIENE LAS LIMITACIONES SOBRE GARANTÍAS Y O REMEDIOS. ESTA LICENCIA LA OTORGA COREL CORPORATION LIMITED PARA PRODUCTOS COMPRADOS FUERA DEL CANADÁ Y COREL CORPORATION PARA PRODUCTOS COMPRADA EN CANADÁ.

LICENCIA

IMPORTANTE: LEA CUIDADOSAMENTE ESTA LICENCIA ANTES DE USAR EL PRODUCTO. EL USO DE ESTE PRODUCTO IMPLICA EL RECONOCIMIENTO DE QUE USTED HA LEÍDO ESTA LICENCIA Y ACORDADO SUS TÉRMINOS. SI NO ESTÁ DE ACUERDO CON ELLOS, DEVUELVA EL PRODUCTO COMPLETO A COREL CORPORATION DENTRO DE LOS 10 DÍAS SIGUIENTES A LA FECHA DE ADQUISICIÓN, PARA UN REEMBOLSO COMPLETO. ESTE ACUERDO DE LICENCIA ES SU PRUEBA DE LA LICENCIA. POR FAVOR TRÁTELO COMO PROPIEDAD VALIOSA.

A. LICENCIA: COREL (”nosotros”), le provee con medios de almacenamiento que contienen un programa informático (el ”Programa”), el manual de usuario, Licencia (la ”Licencia”), la tarjeta de registro y documentos adjuntos (llamados en conjunto el ”Producto”) y le otorga a usted una licencia para usar el Producto según los términos de esta Licencia. La propiedad y todos otros derechos sobre el Producto permanecerán nuestros o de nuestros proveedores. Usted debe reproducir cualquier nota de propiedad o de otro tipo marcadas en el Producto en todas las copias que haga de él.

B. USTED PUEDE:

1. usar el Programa sólo en una red o computadora única, y sólo un usuario único a la vez sin considerar el número de copias originales del Programa incluidas con el Producto. Si usted desea usar el Programa para más usuarios, necesitará una licencia adicional para cada uno de ellos;

2. hacer una de copia del Programa para archivo o copia de seguridad, y

3. transferir el Programa a otra persona, únicamente si usted le transfiere todos sus derechos bajo esta Licencia, cesa todo uso del Programa, borra o destruye cualquier copia (incluyendo la copia en disco duro) hecha para su uso del Programa, y además la otra persona acuerda los términos de esta Licencia.

4. si el producto es una actualización desde otro producto, bien nuestro o de otro proveedor, usted puede usar o transferir el producto sólo conjuntamente con el producto actualizado, a menos que usted lo destruya. Si el producto es una actualización de un producto de Corel, usted puede usar únicamente ese producto actualizado según esta Licencia.

C. USTED NO PUEDE:

1. usar el Producto o hacer copias de él excepto como permite en esta Licencia;

2. crear trabajos escandalosos, obscenos o inmorales usando las imágenes informáticas incluidas con el Programa;

3. publicar o distribuir las imágenes informáticas, las fuentes o archivos de sonido incluidos con el Programa como imágenes informáticas, fuentes o archivos de sonido;

4. usar cualquiera de las imágenes informáticas relativas a entidades o individuos identificables en una manera que sugiera su asociación con ellas o ellos o el respaldo de cualquier producto o servicio;

5. traducir, realizar ingeniería inversa, descompilar o desensamblar el Programa, excepto en los términos en que la restricción anterior está prohibida expresamente por las leyes competentes;

6. alquilar, arrendar, ceder o transferir el Producto excepto como se ha explicado anteriormente; o

7. modificar el Programa o combinar todo o cualquier parte del Programa en otro programa.

D. VIGENCIA: Esta Licencia continuará vigente mientras usted use el Producto. Sin embargo, terminará si usted incumple cualquiera de sus términos o condiciones. Usted acuerda, tras la terminación, destruir todas las copias del Producto. Las Limitaciones de Garantías y la Responsabilidad establecidas más adelante continuarán en vigor incluso después de cualquier terminación.

E. GARANTÍA: Nosotros garantizamos que el medio de almacenamiento de este Producto estará libre de defectos de materiales y fabricación 90 días a partir de la fecha en que usted lo adquiera. Si se produjera tal defecto, devuélvanoslo a la dirección especificada más adelante y nosotros lo reemplazaremos gratuitamente. Este remedio es el único de que dispone para esta garantía; le otorga ciertos derechos y usted puede tener otros derechos legislados que varían según la jurisdicción competente.

F. LIMITACIÓN DE GARANTÍAS Y RESPONSABILIDAD: A excepción de la garantía expresada arriba, el Producto se provee "tal cual", sin otras garantías, o condiciones, expresas o implícitas, incluidas pero no limitadas a: garantías de calidad comercial, comerciabilidad o aptitud para un propósito particular, ni las provistas por ley, estatuto, norma de comercio o curso de acuerdo. Todo el riesgo con respecto a resultados y prestaciones del Producto los asume usted. Ni nosotros ni nuestros distribuidores o proveedores asumimos responsabilidad alguna ante usted o cualquier otra persona o entidad por daños indirectos, incidentales, especiales o consecuentes cualesquiera, incluidos pero no limitados a la pérdida de ingresos o ganancia, pérdida o daños a datos u otra pérdida comercial o económica, aún cuando nosotros hayamos advertido de la posibilidad de tales daños o éstos sean previsibles; ni ante reclamaciones de terceras partes. Nuestra responsabilidad total máxima ante usted, y la de nuestros distribuidores o proveedores, no excederá la cantidad que usted pagara por el Producto. Las limitaciones en esta sección se aplicarán ante incumplimientos que sean o no de un término o condición fundamental, o un incumplimiento fundamental. Algunos estados/países no permiten la exclusión o limitación de responsabilidad por daños consecuentes o incidentales, de modo que la limitación anterior puede no aplicarse a usted.

G. RESTRICCIÓN DE DERECHOS: Si este Producto es adquirido por o para el Gobierno de EE.UU. entonces se provee con Derechos Restringidos. El uso, duplicación, o divulgación por el Gobierno DE EE.UU. está sujeto a las restricciones que se establecen en el subpárrafo (c) (1) (ii) de la cláusula The Rights in Technical Data and Computer Software en DFARS 252.227-7013, o los subpárrafos (c) (1) y (2) del documento Commerical Computer Softwate - Restricted Rights en 48 CFR 52.227-19, o la cláusula 18-52.227-86 (d) del NASA Supplement to the FAR, según sean aplicables. El contratista/ fabricante es Corel Corporation Limited, Europa House, 3rd Floor, Harcourt St., Dublín 2, Irlanda.

H. GENERAL: Esta Licencia es el acuerdo completo entre nosotros, sobresee cualquier otro acuerdo o discusiones, orales o escritas y no puede cambiarse excepto mediante un acuerdo firmado y por escrito. Esta Licencia se regirá e interpretará según las leyes de la Provincia de Ontario, Canadá para productos comprados en Canadá y según las leyes de la República de Irlanda para productos comprados fuera de Canadá. Si una Corte de jurisdicción competente declarara que cualquier provisión de esta Licencia es inválida, ilegal, o inaplicable, dicha provisión se separará de esta Licencia y las demás mantendrán efecto y fuerza plenas."

Ahora, algunos extractos de una licencia de Microsoft Windows 95 (la que vino con mi portátil Toshiba, incluso aunque yo no la quería).

"[...] This End-User License Agreement ("EULA") is a legal agreement between you [...]

The SOFTWARE PRODUCT is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. the SOFTWARE PRODUCT is licensed, not sold.

1. GRANT OF LICENSE. This EULA grants you the following rights:

- Software. You may install and use one copy of the SOFTWARE PRODUCT on the COMPUTER.
  [...]

[...]

2. DESCIRPTION OF OTHER RIGHTS AND LIMITATIONS

- Limitations on Reverse Engineering, Decompilation and Disassembly. You may not reverse engineer, decompile, or disassemble the SOFTWARE PRODUCT, expcept and only to the extent that such activity is expressly permited by applicable law notwithstanding this limitation.
- Separation of components. The SOFTWARE PRODUCT is licensed as a simgle product. Its component parts may not be separated foruse on more than one computer.
- Single COMPUTER. The SOFTWARE PRODUCT is licensed with the COMPUTER as a single integrated product. The SOFTWARE PRODUCT may only be used with the COMPUTER.

- Rental. You may not rent or lease the SOFTWARE PRODUCT.
- Software transfer. You may permanently transfer all of your rights under this EULA only as part of a sale or transfer of the COMPUTER, provided youretain no copies, you transfer all of the SOFTWARE PRODUCT (including all component parts, the media and printed materials, any upgrades, this EULA and, if applicable, the Certificate(s) of Authenticity), and the recipient agrees tothe terms of this EULA. If the SOFTWARE PRODUCT is an upgrade, any transfer must include all prior versions of the SOFTWARE PRODUCT.
  [...]

[...]
6. PRODUCT SUPPORT. Product support for the SOFTWARE PRODUCT is not provided by Microsoft Corporation or its subsidiaries. For product support, please refer to the PC Manufacturer's support number provided in the documentation for the COMPUTER. Should you have any question concerning this EULA, or if you desire to contact PC Manufacturer for any other reason, please refer to the address provided in the documentation [of] the COMPUTER.

[...]

Warranty and Special Provisions for the United States of America [...]

LIMITED WARRANTY. PC Manufacturer warrants that (a) the SOFTWARE will perform substantially in accordance to with the accompanying written materials for a period of ninety (90) days from the date of recepit, and (b) any Microsoft hardware accompanying the SOFTWARE will be free from defects in materials and workmanship under normal use and service for a period of one (1) year from the day of recepit. Any implied warranties on the SOFTWARE and Microsoft hardware are limited to ninety (90) days and one (1) year, respectively. [...]

[...]

NO OTHER WARRANTIES. To the maximun extent permitted by applicable law, PC Manufacturer and its suppliers disclaim all other warranties, either express or implied, including, but not limited to implied warranties of merchantability and fitness for a particular purpose, with regard to the SOFTWARE, the accompanying written materials, and any accompanying hardware. This limited warranty gives you specific legal rights. You may have others which vary from state/jurisdiction to state/jurisdiction.

[...]

C.2 The Debian Free Software Guidelines

La siguiente definición de software libre está tomada del Manual de Política de Debian (Debian Policy Manual[1]), versión 2.4.1.0 del 14 de abril de 1998. Sus autores son Ian Jackson y Christian Schwarz, y está revisado por David A. Morris. Esta definición se usa en ese momento para decidir qué se incluye en la distribución Debian del sistema GNU/Linux.

The Debian Free Software Guidelines (DFSG) is our definition of 'free' software.

1. Free Redistribution. The license of a Debian component may not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license may not require a royalty or other fee for such sale.

2. Source Code. The program must include source code, and must allow distribution in source code as well as compiled form.

3. Derived Works. The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

4. Integrity of The Author's Source Code. The license may restrict source-code from being distributed in modified form *only* if the license allows the distribution of ''patch files'' with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software. (This is a compromise. The Debian group encourages all authors to not restrict any files, source or binary, from being modified.)

5. No Discrimination Against Persons or Groups. The license must not discriminate against any person or group of persons.

6. No Discrimination Against Fields of Endeavor. The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

7. Distribution of License. The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

8. License Must Not Be Specific to Debian. The rights attached to the program must not depend on the program's being part of a Debian system. If the program is extracted from Debian and used or distributed without Debian but otherwise within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the Debian system.

9. License Must Not Contaminate Other Software. The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be free software.

10. Example Licenses. The ''GPL,'' ''BSD,'' and ''Artistic'' licenses are examples of licenses that we consider free.

[1] http://www.debian.org/doc/manuals/debian-policy/

C.3 Licencia Pública GNU

Esta es la conocida *GNU Public License* (GPL), versión 2 (de junio de 1.991), que cubre la mayor parte del software de la Free Software Foundation, y muchos más programas. La traducción al castellano ha sido revisada por Richard Stallman, pero no tienen ningún valor legal, ni ha sido comprobada de acuerdo a la legislación de ningún país en particular.

Preámbulo

Las licencias que cubren la mayor parte del software están diseñadas para quitarle a usted la libertad de compartirlo y modificarlo. Por el contrario, la Licencia Pública General de GNU pretende garantizarle la libertad de compartir y modificar software libre, para asegurar que el software es libre para todos sus usuarios. Esta Licencia Pública General se aplica a la mayor parte del software del la Free Software Foundation y a cualquier otro programa si sus autores se comprometen a utilizarla. (Existe otro software de la Free Software Foundation que está cubierto por la Licencia Pública General de GNU para Bibliotecas). Si quiere, también puede aplicarla a sus propios programas.

Cuando hablamos de software libre, estamos refiriéndonos a libertad, no a precio. Nuestras Licencias Públicas Generales están diseñadas para asegurarnos de que tenga la libertad de distribuir copias de software libre (y cobrar por ese servicio si quiere), de que reciba el código fuente o que pueda conseguirlo si lo quiere, de que pueda modificar el software o usar fragmentos de él en nuevos programas libres, y de que sepa que puede hacer todas estas cosas.

Para proteger sus derechos necesitamos algunas restricciones que prohiban a cualquiera negarle a usted estos derechos o pedirle que renuncie a ellos. Estas restricciones se traducen en ciertas obligaciones que le afectan si distribuye copias del software, o si lo modifica.

Por ejemplo, si distribuye copias de uno de estos programas, sea gratuitamente, o a cambio de una contraprestación, debe dar a los receptores todos los derechos que tiene. Debe asegurarse de que ellos también reciben, o pueden conseguir, el código fuente. Y debe mostrarles estas condiciones de forma que conozcan sus derechos.

Protegemos sus derechos con la combinación de dos medidas:

1. Ponemos el software bajo copyright y

2. le ofrecemos esta licencia, que le da permiso legal para copiar, distribuir y/o modificar el software.

También, para la protección de cada autor y la nuestra propia, queremos asegurarnos de que todo el mundo comprende que no se proporciona ninguna garantía para este software libre. Si el software se modifica por cualquiera y éste a su vez lo distribuye, queremos que sus receptores sepan que lo que tienen no es el original, de forma que cualquier problema introducido por otros no afecte a la reputación de los autores originales.

Por último, cualquier programa libre está constantemente amenazado por patentes sobre el software. Queremos evitar el peligro de que los redistribuidores de un programa libre obtengan patentes por su cuenta, convirtiendo de facto el programa en propietario. Para evitar esto, hemos dejado claro que cualquier patente debe ser pedida para el uso libre de cualquiera, o no ser pedida.

Los términos exactos y las condiciones para la copia, distribución y modificación se exponen a continuación.

Licencia Pública General de GNU
Términos y condiciones para la copia, distribución y modificación

0. Esta Licencia se aplica a cualquier programa u otro tipo de trabajo que contenga una nota colocada por el tenedor del copyright diciendo que puede ser distribuido bajo los términos de esta Licencia Pública General. En adelante, “Programa” se referirá a cualquier programa o trabajo que cumpla esa condición y “trabajo basado en el Programa” se referirá bien al Programa o a cualquier trabajo derivado de él según la ley de copyright. Esto es, un trabajo que contenga el programa o una proción de él, bien en forma literal o con modificaciones y/o traducido en otro lenguaje. Por lo tanto, la traducción está incluida sin limitaciones en el término “modificación”. Cada concesionario (licenciatario) será denominado “usted”.

Cualquier otra actividad que no sea la copia, distribución o modificación no está cubierta por esta Licencia, está fuera de su ámbito. El acto de ejecutar el Programa no está restringido, y los resultados del Programa están cubiertos únicamente si sus contenidos constituyen un trabajo basado en el Programa, independientemente de haberlo producido mediante la ejecución del programa. El que esto se cumpla, depende de lo que haga el programa.

1. Usted puede copiar y distribuir copias literales del código fuente del Programa, según lo has recibido, en cualquier medio, supuesto que de forma adecuada y bien visible publique en cada copia un anuncio de copyright adecuado y un repudio de garantía, mantenga intactos todos los anuncios que se refieran a esta Licencia y a la ausencia de garantía, y proporcione a cualquier otro receptor del programa una copia de esta Licencia junto con el Programa.

Puede cobrar un precio por el acto físico de transferir una copia, y puede, según su libre albedrío, ofrecer garantía a cambio de unos honorarios.

2. Puede modificar su copia o copias del Programa o de cualquier porción de él, formando de esta manera un trabajo basado en el Programa, y copiar y distribuir esa modificación o trabajo bajo los términos del apartado 1, antedicho, supuesto que además cumpla las siguientes condiciones:

a. Debe hacer que los ficheros modificados lleven anuncios prominentes indicando que los ha cambiado y la fecha de cualquier cambio.

b. Debe hacer que cualquier trabajo que distribuya o publique y que en todo o en parte contenga o sea derivado del Programa o de cualquier parte de él sea licenciada como un todo, sin carga alguna, a todas las terceras partes y bajo los términos de esta Licencia.

c. Si el programa modificado lee normalmente órdenes interactivamente cuando es ejecutado, debe hacer que, cuando comience su ejecución para ese uso interactivo de la forma más habitual, muestre o escriba un mensaje que incluya un anuncio de copyright y un anuncio de que no se ofrece ninguna garantía (o por el contrario que sí se ofrece garantía) y que los usuarios pueden redistribuir el programa bajo estas condiciones, e indicando al usuario cómo ver una copia de esta licencia. (Excepción: si el propio programa es interactivo pero normalmente no muestra ese anuncio, no se requiere que su trabajo basado en el Programa muestre ningún anuncio).

Estos requisitos se aplican al trabajo modificado como un todo. Si partes identificables de ese trabajo no son derivadas del Programa, y pueden, razonablemente, ser consideradas trabajos independientes y separados por ellos mismos, entonces esta Licencia y sus términos no se aplican a esas partes cuando sean distribuidas como trabajos separados. Pero cuando distribuya esas mismas secciones como partes de un todo que es un trabajo basado en el Programa, la distribución del todo debe ser según los términos de esta licencia, cuyos permisos para otros licenciatarios se extienden al todo completo, y por lo tanto a todas y cada una de sus partes, con independencia de quién la escribió.

Por lo tanto, no es la intención de este apartado reclamar derechos o desafiar sus derechos sobre trabajos escritos totalmente por usted mismo. El intento es ejercer el derecho a controlar la distribución de trabajos derivados o colectivos basados en el Programa.

Además, el simple hecho de reunir un trabajo no basado en el Programa con el Programa (o con un trabajo basado en el Programa) en un volumen de almacenamiento o en un medio de distribución no hace que dicho trabajo entre dentro del ámbito cubierto por esta Licencia.

3. Puede copiar y distribuir el Programa (o un trabajo basado en él, según se especifica en el apartado 2, como código objeto o en formato ejecutable según los términos de los apartados 1 y 2, supuesto que además cumpla una de las siguientes condiciones:

a. Acompañarlo con el código fuente completo correspondiente, en formato electrónico, que debe ser distribuido según se especifica en los apartados 1 y 2 de esta Licencia en un medio habitualmente utilizado para el intercambio de programas, o

b. Acompañarlo con una oferta por escrito, válida durante al menos tres años, de proporcionar a cualquier tercera parte una copia completa en formato electrónico del código fuente correspondiente, a un coste no mayor que el de realizar físicamente la distribución del fuente, que será distribuido bajo las condiciones descritas en los apartados 1 y 2 anteriores, en un medio habitualmente utilizado para el intercambio de programas, o

c. Acompañarlo con la información que recibiste ofreciendo distribuir el código fuente correspondiente. (Esta opción se permite sólo para distribución no comercial y sólo si usted recibió el programa como código objeto o en formato ejecutable con tal oferta, de acuerdo con el apartado b anterior).

Por código fuente de un trabajo se entiende la forma preferida del trabajo cuando se le hacen modificaciones. Para un trabajo ejecutable, se entiende por código fuente completo todo el código fuente para todos los módulos que contiene, más cualquier fichero asociado de definición de interfaces, más los guiones utilizados para controlar la compilación e instalación del ejecutable. Como excepción especial el código fuente distribuido no necesita incluir nada que sea distribuido normalmente (bien como fuente, bien en forma binaria) con los componentes principales (compilador, kernel y similares) del sistema operativo en el cual funciona el ejecutable, a no ser que el propio componente acompañe al ejecutable.

Si la distribución del ejecutable o del código objeto se hace mediante la oferta acceso para copiarlo de un cierto lugar, entonces se considera la oferta de acceso para copiar el código fuente del mismo lugar como distribución del código fuente, incluso aunque terceras partes no estén forzadas a copiar el fuente junto con el código objeto.

4. No puede copiar, modificar, sublicenciar o distribuir el Programa excepto como prevé expresamente esta Licencia. Cualquier intento de copiar, modificar sublicenciar o distribuir el Programa de otra forma es inválida, y hará que cesen automáticamente los derechos que te proporciona esta Licencia. En cualquier caso, las partes que hayan recibido copias o derechos de usted bajo esta Licencia no cesarán en sus derechos mientras esas partes continúen cumpliéndola.

5. No está obligado a aceptar esta licencia, ya que no la ha firmado. Sin embargo, no hay hada más que le proporcione permiso para modificar o distribuir el Programa o sus trabajos derivados. Estas acciones están prohibidas por la ley si no acepta esta Licencia. Por lo tanto, si modifica o distribuye el Programa (o cualquier trabajo basado en el Programa), está indicando que acepta esta Licencia para poder hacerlo, y todos sus términos y condiciones para copiar, distribuir o modificar el Programa o trabajos basados en él.

6. Cada vez que redistribuya el Programa (o cualquier trabajo basado en el Programa), el receptor recibe automáticamente una licencia del licenciatario original para copiar, distribuir o modificar el Programa, de forma sujeta a estos términos y condiciones. No puede imponer al receptor ninguna restricción más sobre el ejercicio de los derechos aquí garantizados. No es usted responsable de hacer cumplir esta licencia por terceras partes.

7. Si como consecuencia de una resolución judicial o de una alegación de infracción de patente o por cualquier otra razón (no limitada a asuntos relacionados con patentes) se le imponen condiciones (ya sea por mandato judicial, por acuerdo o por cualquier otra causa) que contradigan las condiciones de esta Licencia, ello no le exime de cumplir las condiciones de esta Licencia. Si no puede realizar distribuciones de forma que se satisfagan simultáneamente sus obligaciones bajo esta licencia y cualquier otra obligación pertinente entonces, como consecuencia, no puede distribuir el Programa de ninguna forma. Por ejemplo, si una patente no permite la redistribución libre de derechos de autor del Programa por parte de todos aquellos que reciban copias directa o indirectamente a través de usted, entonces la única forma en que podría satisfacer tanto esa condición como esta Licencia sería evitar completamente la distribución del Programa.

Si cualquier porción de este apartado se considera inválida o imposible de cumplir bajo cualquier circunstancia particular ha de cumplirse el resto y la sección por entero ha de cumplirse en cualquier otra circunstancia.

No es el propósito de este apartado inducirle a infringir ninguna reivindicación de patente ni de ningún otro derecho de propiedad o impugnar la validez de ninguna de dichas reivindicaciones. Este apartado tiene el único propósito de proteger la integridad del sistema de distribución de software libre, que se realiza mediante prácticas de licencia pública. Mucha gente ha hecho contribuciones generosas a la gran variedad de software distribuido mediante ese sistema con la confianza de que el sistema se aplicará consistentemente. Será el autor/donante quien decida si quiere distribuir software mediante cualquier otro sistema y una licencia no puede imponer esa elección.

Este apartado pretende dejar completamente claro lo que se cree que es una consecuencia del resto de esta Licencia.

8. Si la distribución y/o uso de el Programa está restringida en ciertos países, bien por patentes o por interfaces bajo copyright, el tenedor del copyright que coloca este Programa bajo esta Licencia puede añadir una limitación explícita de distribución geográfica excluyendo esos países, de forma que la distribución se permita sólo en o entre los países no excluidos de esta manera. En ese caso, esta Licencia incorporará la limitación como si estuviese escrita en el cuerpo de esta Licencia.

9. La Free Software Foundation puede publicar versiones revisadas y/o nuevas de la Licencia Pública General de tiempo en tiempo. Dichas nuevas versiones serán similares en espíritu a la presente versión, pero pueden ser diferentes en detalles para considerar nuevos problemas o situaciones.

Cada versión recibe un número de versión que la distingue de otras. Si el Programa especifica un número de versión de esta Licencia que se refiere a ella y a "cualquier versión posterior", tienes la opción de seguir los términos y condiciones, bien de esa versión, bien de cualquier versión posterior publicada por la Free Software Foundation. Si el Programa no especifica un número de versión de esta Licencia, puedes escoger cualquier versión publicada por la Free Software Foundation.

10. Si quiere incorporar partes del Programa en otros programas libres cuyas condiciones de distribución son diferentes, escribe al autor para pedirle permiso. Si el software tiene copyright de la Free Software Foundation, escribe a la Free Software Foundation: algunas veces hacemos excepciones en estos casos. Nuestra decisión estará guiada por el doble objetivo de de preservar la libertad de todos los derivados de nuestro software libre y promover el que se comparta y reutilice el software en general.

AUSENCIA DE GARANTÍA

11. Como el programa se licencia libre de cargas, no se ofrece ninguna garantía sobre el programa, en todas la extensión permitida por la legislación aplicable. Excepto cuando se indique de otra forma por escrito, los tenedores del copyright y/u otras partes proporcionan el programa "tal cual", sin garantía de ninguna clase, bien expresa o implícita, con inclusión, pero sin limitación a las garantías mercantiles implícitas o a la conveniencia para un propósito particular. Cualquier riesgo referente a la calidad y prestaciones del programa es asumido por usted. Si se probase que el Programa es defectuoso, asume el coste de cualquier servicio, reparación o corrección.

12. En ningún caso, salvo que lo requiera la legislación aplicable o haya sido acordado por escrito, ningún tenedor del copyright ni ninguna otra parte que modifique y/o redistribuya el Programa según se permite en esta Licencia será responsable ante usted por daños, incluyendo cualquier daño general, especial, incidental o resultante producido por el uso o la imposibilidad de uso del Programa (con inclusión, pero sin limitación a la pérdida de datos o a la generación incorrecta de datos o a pérdidas sufridas por usted o por terceras partes o a un fallo del Programa al funcionar en combinación con cualquier otro programa), incluso si dicho tenedor u otra parte ha sido advertido de la posibilidad de dichos daños.

FIN DE TÉRMINOS Y CONDICIONES

Cómo aplicar estos términos a sus nuevos programas.

Si usted desarrolla un nuevo Programa, y quiere que sea del mayor uso posible para el público en general, la mejor forma de conseguirlo es convirtiéndolo en software libre que cualquiera pueda redistribuir y cambiar bajo estos términos.

Para hacerlo, añada los siguientes anuncios al programa. Lo más seguro es añadirlos al principio de cada fichero fuente para transmitir lo más efectivamente posible la ausencia de garantía. Además cada fichero debería tener al menos la línea de "copyright" y un indicador a dónde puede encontrarse el anuncio completo.

```
   <una línea para indicar el nombre del programa y una rápida idea de qué hace.>
   Copyright (C) 19aa <nombre del autor>
   Este programa es software libre.  Puede redistribuirlo y/o modificarlo bajo los términos de
la Licencia Pública General de GNU según es publicada por la Free Software Foundation, bien de la
versión 2 de dicha Licencia o bien (según su elección) de cualquier versión posterior.
   Este programa se distribuye con la esperanza de que sea útil, pero SIN NINGUNA GARANTÍA, incluso
sin la garantía MERCANTIL implícita o sin garantizar la CONVENIENCIA PARA UN PROPÓSITO PARTICULAR.
Véase la Licencia Pública General de GNU para más detalles.
   Debería haber recibido una copia de la Licencia Pública General junto con este programa.  Si
no ha sido así, escriba a la Free Software Foundation, Inc., en 675 Mass Ave, Cambridge, MA 02139,
EEUU.
```

Añada también información sobre cómo contactar con usted mediante correo electrónico y postal.

Si el programa es interactivo, haga que muestre un pequeño anuncio como el siguiente, cuando comienza a funcionar en modo interactivo:

`Gnomovision versión 69, Copyright (C) 19aa nombre del autor`

`Gnomovision no ofrece ABSOLUTAMENTE NINGUNA GARANTÍA. Para más detalles escriba ''show w''.`

`Esto es software libre, y se le invita a redistribuirlo bajo ciertas condiciones. Escriba ''show c'' para más detalles.`

Los comandos hipotéticos "show w" y "show c" deberían mostrar las partes adecuadas de la Licencia Pública General. Por supuesto, los comandos que use pueden llamarse de cualquier otra manera. Podrían incluso ser pulsaciones del ratón o elementos de un menú (lo que sea apropiado para su programa).

También deberías conseguir que su empleador (si trabaja como programador) o tu Universidad (si es el caso) firme un "renuncia de copyright" para el programa, si es necesario. A continuación se ofrece un ejemplo, altere los nombres según sea conveniente:

`Yoyodyne, Inc. mediante este documento renuncia a cualquier interés de derechos de copyright con respecto al programa Gnomovision (que hace pasadas a compiladores) escrito por Pepe Programador.`

`<firma de Pepito Grillo>, 20 de diciembre de 1996> Pepito Grillo, Presidente de Asuntillos Varios.`

Esta Licencia Pública General no permite que incluya sus programas en programas propietarios. Si su programa es una biblioteca de subrutinas, puede considerar más útil el permitir el enlazado de aplicaciones propietarias con la biblioteca. Si este es el caso, use la Licencia Pública General de GNU para Bibliotecas en lugar de esta Licencia.

C.4 Licencia GNU para bibliotecas

Esta es la GNU Library General Public License (LGPL), versión 2 (de junio de 1.991). Está diseñada para proteger a las bibliotecas que desarrolla la Free Software Foundation, y se utiliza también para muchas otras bibliotecas.

[This is the first released version of the library GPL. It is numbered 2 because it goes with version 2 of the ordinary GPL.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users.

This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

GNU LIBRARY GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

> a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)
>
> b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
>
> c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
>
> d) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

> a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
>
> b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or

otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY

YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Library General Public
License as published by the Free Software Foundation; either
version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
Library General Public License for more details.

You should have received a copy of the GNU Library General Public
License along with this library; if not, write to the Free
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
```

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice
```

That's all there is to it!

C.5 Licencia BSD

Esta es la licencia aplicada a las distribuciones de software del Computer Science Research Group, de la Universidad de California en Berkeley.

```
Copyright (c) The Regents of the University of California.
All rights reserved.
```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgement: ''This product includes software developed by the University of California, Berkeley and its contributors''.

4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ''AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

C.6 X Window System (X Consortium) License

Esta es la licencia de distribución del Sistema X Window, X11R6 (versión 11, entrega 6).

C.7 XFree86 Project License

Esta es la licencia de XFree86 3.3.2, distribuido por el Proyecto XFree86:

C.8 Licencia de Tcl/Tk

Esta es la licencia bajo la que se distribuyen la versión 7.6 de Tcl y la 4.2 de Tk.

C.9 Licencia Artística

Esta es la *Artistic license*, usada entre otros por Larry Wall como una de las licencias de distribución de Perl (la otra es la GPL).

Comment (jgb): *Is Larry Wall the original author of this license?*

Preamble:

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the Package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

It also grants you the rights to reuse parts of a Package in your own programs without transferring this License to those programs, provided that you meet some reasonable requirements.

Definitions:

‘‘Package’’ refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.

‘‘Standard Version’’ refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder as specified below.

‘‘Copyright Holder’’ is whoever is named in the copyright or copyrights for the package.

‘‘You’’ is you, if you’re thinking about copying or distributing this Package.

‘‘Reasonable copying fee’’ is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)

‘‘Freely Available’’ means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.

2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.

3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:

   (a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as uunet.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.

   (b) use the modified Package only within your corporation or organization.

   (c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.

   (d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

(a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

(b) accompany the distribution with the machine-readable source of the Package with your modifications.

(c) give non-standard executables non-standard names, and clearly document the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

(d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this Package. If such scripts or library files are aggregated with this Package via the so-called ‘‘undump’’ or ‘‘unexec’’ methods of producing a binary executable image, then distribution of such an image shall neither be construed as a distribution of this Package nor shall it fall under the restrictions of Paragraphs 3 and 4, provided that you do not represent such an executable image as a Standard Version of this Package.

7. You may reuse parts of this Package in your own programs, provided that you explicitly state where you got them from, in the source code (and, left to your courtesy, in the documentation), duplicating all the associated copyright notices and disclaimers. Besides your changes, if any, must be clearly marked as such. Parts reused that way will no longer fall under this license if, and only if, the name of your program(s) have no immediate connection with the name of the Package itself or its associated programs. You may then apply whatever restrictions you wish on the reused parts or choose to place them in the Public Domain---this will apply only within the context of your package.

8. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

9. THIS PACKAGE IS PROVIDED ‘‘AS IS’’ AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Apéndice D

Fechas relacionadas con el software libre

Comment (jgb): *SPICE should be included*

Comment (jgb): *More info about TeX/LaTeX should be included*

Comment (jgb): *More milestones of BSD and GNU should be included*

Comment (jgb): *Tcl/Tk should be included*

Comment (jgb): *PTF CDs should be included (1992?)*

D.1 Fechas históricas

1950s y 1960s: El software se distribuye con c'odigo fuente y sin restricciones en foros como los grupo de ususarios SHARE de IBM o DECUS de DEC, o en la sección "Algorithms" de la revista "Communications of the ACM".

1969 Abr: Se publica la RFC número 1, que describe el primer software pare Internet (entonces ARPANET). La disponibilidad libre de las RFCs y específicamente de las especificaciones de los protocolos fue un factor clave en el desarrollo de Internet.

1972: El código fuente se distribuye libremente en círculos académicos. Ejemplo: MIT, hacia 1972, HACKMEM (lenguaje ensamblador para PDP-6/10).

1978: El profesor Donald E. Knuth de la Universidad de Stanford comienza a trabajar en TeX, un sistema de composición de textos que se distribuye como software libre.

1983:] Richard Stallman escribe el Manifiesto GNU, en el que clama por que se vuelva a compartir públicamente el software y el código fuente.

1983: Comienza el Proyecto GNU. Los desarrolladores comienzan a crear una gran variedad de herramientas estilo Unix, incluyendo por ejemplo un editor (Emacs) y un compilador (GCC).

1985: El Consorcio X, basado en el MIT, distribuye el Sistema de Ventanas X (X Window) como software libre cubierto por una de las licencias libres menos restrictivas en uso.

1989: Se funda Cygnus, la primera empresa comercial dedicada a proporcionar mantenimiento para el software de GNU, y para el software libre en general.

1990: La Free Software Foundation anuncia su intención de construir un kernel tipo Unix llamado GNU Hurd. Su meta es llenar el último hueco en el software de GNU para crear un sistema de desarrollo completamente libre.

1991: William and Lynne Jolitz escriben una serie en Dr. Dobbs Journal sobre cómo portar Unix BSD a PCs basados en el i386. Como resultado del software libre desarrollado y recopilado en el CSRG de la Universidad de California en Berkeley, era posible tener un sistema operativo BSD completo (y libre). Este es el punto de partida para la familia BSD de sistemas operativos libres (NetBSD, FreeBSD, OpenBSD).

1991 Ago: El estudiante de grado finlandés Linus Torvalds anuncia que ha estado trabajando en un kernel libre estilo Unix desde abril de ese año, usando herramientas de GNU, como GCC.

1991 Oct: Torvalds libera públicamente el código fuente de su kernel estilo Unix, que se llama Linux.

1991 Dic: Torvalds anuncia la primera entrega de Linux que se autosoporta, la versión 0.11. Los desarrolladores pueden ahora trabajar en Linux sin usar herramientas o sistemas operativos propietarios.

1992 Ene: Tanenbaum critica públicamente Linxu por ser técnicamente obsoleto, y demasiado específico para una arquitectura. En el diálogo posterior que tuvo lugar en Usenet Torvalds adopta una actitud favorable al software libre que ayuda a atraer a nuevos desarrolladores Linux.

1992 Jul: 386BSD 0.1 es liberado por William y Lynne Jolitz.

> **Comment (jgb):** *Something should be explained about the legal battle about Net/2. Somehtinkg like: A legal battle started to determine if there was some proprietary code in Berkeley Net Release/2, in which 386BSD was based. Some months later, a settlement was achieved with the release of Berkeley Net Release/3, in which all modern free *BSD operating systems are based.*

1992: El Ejército del Aire de EEUU concede a la Universidad de Nueva York (NYU) un contrato para construir un compilador libre para lo que ahora se llama Ada95. El equipo de la NYU elige GNU GCC para al generación de código y llama GNAT a su compilador (*GNU NYU Ada Translator*, o Traductor de Ada GNU NYU).

1993 Ago: Ian Murdock crea una nueva distribución de Linux que se llama Debian GNU/Linux, desarrollada por un grupo de voluntarios repartidos por el mundo.

1993 Dic: FreeBSD 1.0, uno de los primeros descendientes del 386BSD de los Jolitz, está disponible en la red.

1994: GNAT recibe un gran impulso comercial con la fundación de Ada Core Technologies (ACT) por sus creadores originales. ACT decide ganar dinero mediante la evolución de GNAT y la venta de servicios de mantenimento, en lugar de mediante la venta del propio GNAT (que no era posible, en cualquier caso, ya que estaba cubierto por la GPL). Con el tiempo y con la ayuda de ACT, GNAT se ha convertido en el compilador dominante de Ada 95.

1994 Ene: Debian GNU/Linux (versión 0.91), desarrollado por 12 voluntarios, es liberado.

1994: Marc Ewing comienza la distribución Red Hat de GNU/Linux. Como la distribución Debian, se intenta mejorar la entonces dominante distribución Softlanding Linux System (SLS).

> **Comment (jgb):** *And what about Slackware?*

1994 Mar: Se publica el primer ejemplar de Linux Journal.

1994 Oct: Liberación de NetBSD 1.0.

1995 Ene: Liberación de FreeBSD 2.0.

1995 Abr: Se distribuye la primera entrega oficial (0.6.2) de Apache.

> **Comment (jgb):** *Dates for CERNÑCSA*

1996: Primera Conferencia sobre Software Redistribuible Libremente (*First Conference on Freely Redistributable Software*). Cambridge, Massachusetts, EEUU.

1997 Jun: Eric S. Raymond presenta su artículo “La catedral y el bazar” (*“The Cathedral and the Bazaar”* sobre porqué funciona el modelo de desarrollo de software de Linux.

1998 Ene: Netscape declara su intención de liberar el código fuente de su navegador (Navigator), persuadidos en parte por el artículo de Raymond.

1998 Feb: Chris Peterson y otros acuñan el término “fuente abierto” (*“open source”*) y lo registran, para que funcione como una marca registrada para los productos de software libre.

1998 Abr: El código fuente de Netscape es liberado, y en pocas horas empiezan a llegar mejoras y parches preliminares.

1998 Jun: FREENIX, la parte de software libremente redistribuible (*Freely Redistributable Software Track*) de la Conferencia Técnica de USENIX (*USENIX Technical Conference*) se dedica a los desarrolladores y usuarios de software libre.

1998 Jul: Debian 2.0 es liberada por más de 300 desarrolladores trabajando en más de 1.500 paquetes.

1998 Ago 10: Linux Torvalds y Linux aparecen en la portada de la revista Forbes.

1998 Oct: IBM decide probar el fuente abierto al usar Apache en sus servidores AS/400.

1998 Oct: Intel y Netscape invierten en Red Hat.

1998 Nov: Los documentos “Halloween” (atribuidos a Microsoft) son filtrados al público por Eric S. Raymond. Los documentos analizan los puntos fuertes y débiles del software de fuente abierto y de Linux.

D.2 Conferencias y exhibiciones

1999, Mar 13-15: The Bazaar Exhibition[1], New York City, NY, USA.

1999, Jun 6-11: USENIX/Freenix[2], Monterey, CA, USA.

1998, Jun 3-5: Internationaler Linux Kongress[3], Koln, Germany.

1999, Aug 21: O’Reilly Open Source Conference[4], Monterey, CA, USA.

1999, Mar 1-4: LinuxWorld Conference & Expo[5], San Jose, CA, USA.

1999, May 18-22: Linux Expo[6], Raleigh, NC, USA.

[1] `http://www.thebazaar.org`
[2] `http://www.usenix.org/events/usenix99/cfp/cfp_5a.html`
[3] `http://www.linux-kongress.de/`
[4] `http://conferences.ora.com`
[5] `http://www.linuxworld.com`
[6] `http://www.linuxexpo.org`

Apéndice E

Créditos

Debo agradecer su colaboración a las siguientes personas, que han hecho posible este trabajo:

Javier Macias-Guarasa (macias@die.upm.es). Contributed the Corel 7.0 license in Spanish.

> **Comment (jgb):** *If I have forgotten some of you, who have helped someway to make this happen, please let me know. I'll be very happy of correcting my error!*

Apéndice F

Bibliografía comentada

Comment (jgb): *Detail them...*

- Actas del taller software libre, UCAR 1995.
- Proceedings 1st Conf. on Freely Redistributable Software.
- GNU Bulletin.
- Linux Journal.
- Netizens.
- Monografía de Novática sobre software libre.
- "Una breve historia de Internet", March 97, Communications of the ACM, publicado en castellano "Cuadernos de Novática", feb 1988.

Bibliografía

[1] Steve Crocker. Documentation conventions. Technical Report RFC 3, Network Working Group, April 1969.

[2] Peter L. Deutsch. Licenses for freely redistributable software. In *Proceedings of the First Conference on Freely Redistributable Software*, Cambridge, Massachusetts, USA, February 1996. Also available in Spanish in `http://www.gsyc.inf.uc3m.es/sobre/novatica-mono/deutsch/`.

[3] Michael Hauben and Ronda Hauben. *Netizens. On the History and Impact of Usenet and the Internet.* IEEE Computer Society Press, 1997.

[4] Ted Lewis. *Friction-Free Economy.* Harper Collins, 1997.

[5] Tekla S. Perry. Donald o. pederson. *IEEE Spectrum*, 35(6):22–27, June 1998.

[6] Lawrence Roberts. The evolution of packet switching. *Proceedings of the IEEE*, (66), November 1978.

[7] Marc Rochkind. Interview with dick haight. *Unix Review*, May 1986.