

ULACIT
UNIVERSIDAD LATINOAMERICANA DE CIENCIA Y TECNOLOGÍA

LICENCIATURA EN INGENIERIA INFORMÁTICA CON ÉNFASIS EN
DESARROLLO DE SISTEMAS.

“SUITE OFIMATICA LIBRE: ASPECTOS QUE SE DEBEN TOMAR EN
CUENTA PARA DETERMINAR SI UNA HERRAMIENTA DE ESTE TIPO ES
VIABLE”

SUSTENTANTE
Carlos Enrique Badilla Sánchez
Cédula 1-0668-0185

PROYECTO DE GRADUACIÓN PARA OPTAR POR EL GRADO
LICENCIATURA EN INGENIERÍA INFORMÁTICA CON ÉNFASIS
EN DESARROLLO DE SOFTWARE.

PROF. Guillermo Oviedo Blanco.

San José – Costa Rica
Diciembre 2005.

Tabla de contenido

Resumen.....	3
Abstract.....	3
Palabras claves.....	4
Introducción.....	5
Objetivos y características del “Free Software”	6
El proyecto GNU.....	7
Definición de “Free Software”	8
Evolución del Software libre.....	9
Iniciativas de Software libre	11
Clásicos del Software libre.....	13
Las características del software libre.....	14
Otras razones del software libre.....	16
Regulación legal de la producción y distribución de software.....	17
Definición de CopyRight.....	19
Definición de CopyLeft.....	23
Definición licencia de software.....	24
Licencias propietarias.....	25
Licencias de software libre.....	27
Cláusulas de no responsabilidad.....	31
Definición de ofimática.....	32
Definición formato de almacenamiento.....	34

Formato cerrado.....	34
Formato publicado.....	34
Formato abierto.....	35
El formato OpenDocument.....	36
Implicaciones de orden público.....	37
Otras implicaciones.....	38
Costos licencias de ofimática.....	39
Mitos y realidades acerca del software libre.....	41
Sólo existe software libre para Linux.....	42
No hay soporte para las empresas.....	43
Nadie escribe software gratis.....	43
El software libre destruye la industria.....	43
El software libre destruye la innovación.....	44
La disponibilidad del código fuente crea versiones incompatibles.....	44
Las empresas competidoras copiarán tu código	44
El software libre es más inseguro que el propietario.....	44
Recomendaciones finales.....	45
Bibliografía.....	48

Resumen

Primeramente se define ¿Qué es "Free Software"? ¿Cuáles son sus objetivos y características fundamentales?. En segundo lugar se hace un recorrido histórico de los últimos años del movimiento de "Free Software", identificando las principales iniciativas, sus tendencias, razones económicas, políticas, sociales, y psicológicas del movimiento. En tercer lugar se tratan aspectos legales sobre la producción y distribución de software, enumerando los tipos de licencias de software y las características particulares. En cuarto lugar se define ¿Qué es ofimática?, y se describen los formatos de almacenamiento: cerrado, publicado, abierto y las implicaciones de cada uno. En quinto lugar, se muestra un ejemplo de costos por licenciamiento de herramientas de ofimática, para una empresa del sector comercial de Costa Rica. Finalmente se mencionan los aspectos a tomar en cuenta, para determinar si una herramienta de ofimática multiplataforma, distribuida bajo alguna licencia de software libre, representa una oportunidad real para las empresas y usuarios en general.

Abstract

Firstly it is defined :What is "Free Software"? Which are their characteristic objectives and fundamental. Secondly it is made historical route of the last years of movement of "Free Software", identifying main initiatives, its tendencies, reasons economic, policies, social, and psychological of movement. Thirdly aspects treat legal on the production and distribution of software, enumerating the types of licenses of software and the particular characteristics. In quarter place is defined What is "ofimática"?, and they are described the storage formats: closed, published, opened and the implications of each one. In fifth place, is an example of costs by licensing of "ofimática" tools, for a company of the commercial sector of Costa Rica. Finally the aspects are mentioned that to consider, in order to determine if a tool of ofimática multiplatform, distributed under some license of free software, it represents one real opportunity, by the users of this type of product.

Palabras claves

Software libre, libertades del software, ofimática, licencias de software, derechos de autor, copyleft.

Introducción

El software libre se inicio en pequeños grupos de idealistas que deseaban cambiar la industria del software. Después de algunos años la comunidad ha crecido e incorporado voluntarios en todo el mundo, además de atraer la atención y colaboración de algunas empresas claves.

Linux, -el estandarte de este movimiento-, en sus inicios, fue popular en servidores y con el paso del tiempo y gracias al esfuerzo de miles de colaboradores, ahora es un sistema que es usado por miles de usuarios en todo el mundo en computadoras personales, teléfonos y organizadores personales.

El uso de software libre es similar al software no libre, sin embargo, el primero es un legado que le pertenece a toda la humanidad, no tiene propietario, de la misma manera que las leyes básicas de la física o las matemáticas. No existe un monopolio y no es necesario pagar para utilizarlo. En los últimos años, ha sido adoptado por algunos gobiernos y otros han iniciado importantes iniciativas a fin de adoptar herramientas que implementen estándares abiertos.

Con respecto a las herramientas de ofimática, existen productos distribuidos bajo licencias de software libre, con características similares, e incluso superiores al líder indiscutible, -Microsoft Office- lo que plantea la necesidad de identificar que aspectos deben considerarse para identificar las ventajas y oportunidades que puede ofrecer el modelo de software libre, tanto para las empresas y organizaciones, como para los individuos en general.

Este trabajo brinda información sobre temas estrechamente relacionados al software libre: Las libertades del software, los modelos de distribución, las características del movimiento, las razones del modelo, y principalmente sus ventajas con respecto al modelo de software propietario.

Objetivos y características del “Free Software”

Para hablar del movimiento "Free Software", es necesario mencionar al artífice de esta iniciativa, Richard Mathew Stallman.

En 1971, siendo estudiante de primer año de Física en la Universidad de Harvard, Stallman se convirtió en un “hacker” del laboratorio de inteligencia artificial del Massachusetts Institute of Technology (MIT). Comenzó a interesarse al respecto cuando trabajaba en el laboratorio de inteligencia artificial del MIT. Tenían una impresora que se averiaba con alguna regularidad, pero gracias a que podían acceder al código del programa controlador de la impresora, resolvían favorablemente las fallas. En algún momento, se decidió cambiar la impresora, pero el nuevo proveedor se negó a dar los fuentes. Es entonces cuando Stallman toma conciencia del peligro de la lógica propietaria.

En los años 80, la cultura “hacker” que constituía la vida de Stallman, empezó a disolverse bajo la presión de la comercialización en la industria del software. En particular, otros “hackers” del laboratorio de Inteligencia Artificial fundaron la compañía Symbolics, la cual intentaba activamente reemplazar el software libre del laboratorio con su propio software propietario. Desde 1983 a 1985, Stallman duplicó los esfuerzos de los programadores de Symbolics para impedir que adquirieran un monopolio sobre los ordenadores del laboratorio.

Se le pidió que firmara un acuerdo de no divulgación ("non-disclosure agreement") y llevara a cabo otras acciones que él consideró traiciones a sus principios. En 1984, Stallman publicó “El Manifiesto GNU”, en el cual declaraba sus intenciones y motivaciones para crear una alternativa libre al sistema operativo Unix, el cual nombró GNU. Poco tiempo después, se incorporó a la organización no lucrativa "Free Software" Foundation (FSF) para coordinar el esfuerzo, proveer soportes logísticos, legales y financieros al proyecto GNU.

El proyecto GNU

Es considerado el estandarte del movimiento del “Free Software”, el objetivo de

este proyecto fue crear un sistema operativo completo libre: el sistema GNU. En septiembre de 1983 se anunció el proyecto públicamente por primera vez. Al anuncio original, siguieron otros ensayos escritos por Richard Stallman, en Free Software Foundation (1985) se establecen las motivaciones para realizar el proyecto, entre las que destaca:

"Retornar al espíritu de cooperación que prevaleció en los tiempos iniciales de la comunidad de usuarios de computadoras".

GNU es un acrónimo recursivo que significa "GNU No es Unix".

UNIX es un sistema operativo no libre, muy popular, porque está basado en una arquitectura que ha demostrado ser técnicamente estable. Diseñado para ser totalmente compatible con UNIX implica que GNU esté compuesto de pequeñas piezas individuales de software, muchas de las cuales ya estaban disponibles, como el sistema de edición de textos TeX y el sistema gráfico XWindow, que fueron adaptadas y reutilizadas; otras en cambio fueron reescritas.

A medida que GNU ganaba renombre, negocios interesados comenzaron a contribuir al desarrollo o comercialización de productos GNU y a brindar el correspondiente soporte técnico. El más prominente y exitoso de ellos fue Cygnus Solutions, ahora parte de Red Hat.

En 1990, el sistema GNU contaba con un editor de texto llamado Emacs, el exitoso compilador GCC, y la mayor parte de las librerías y utilidades que componen un sistema operativo UNIX típico; pero faltaba el componente más importante, el núcleo o "kernel".

Inicialmente, se trabajó con el núcleo TRIX, desarrollado por el MIT y cuyos autores decidieron que fuera distribuido libremente, trix era totalmente compatible con UNIX versión 7. Poco tiempo después, los programadores decidieron que no era inicialmente utilizable, debido a que solamente funcionaba en "algunos equipos sumamente complicados y caros" razón por la cual debería ser portado a otras arquitecturas antes de que se pudiera utilizar. Finalmente, en 1988, se decidió utilizar como base el núcleo Mach desarrollado en la Carnegie-Mellon

University (CMU). Este nuevo núcleo recibió el nombre de Hurd; pero debido a razones técnicas el desarrollo acabó por estancarse.

En 1991, Linus Torvalds empezó a escribir el núcleo Linux. Múltiples programadores se unieron a Linus en el desarrollo a través de Internet. Paulatinamente se consiguió que Linux llegase a ser un núcleo compatible con UNIX.

En 1992, el núcleo Linux fue combinado con el sistema GNU. Esto resultó en un sistema operativo libre y completamente funcional. El sistema operativo formado por esta combinación es usualmente conocido como "GNU/Linux" o como una "distribución Linux". Es frecuente encontrar componentes de GNU instalados en un sistema UNIX no libre, en lugar de los programas originales para UNIX. Debido a que muchos de los programas escritos por el proyecto GNU han demostrado ser de mayor calidad que sus versiones equivalentes de UNIX. A menudo, estos componentes se conocen como "herramientas GNU". Muchos de los programas GNU han sido también portados a otras plataformas como Microsoft Windows y MacOSX.

Definición de "Free Software"

La palabra anglosajona 'free' traducida al español tiene dos acepciones: libre y gratis. En algunos círculos se utilizan los términos "software libre" y "software gratis" para evitar la ambigüedad de esta palabra. Sin embargo, estos términos alternativos son usados únicamente dentro del movimiento del "Free Software", aunque están extendiéndose lentamente hacia el resto del mundo.

Según Richard Stallman, miembro activo de la FSF, y precursor del movimiento, el término "free software" ha sido mal interpretado, pues no tiene nada que ver con precio, tiene que ver con libertad.

De acuerdo con esto, se considera software libre al programa de software, sistema operativo o programa aplicativo, que garantice las siguientes libertades:

- La libertad de usar el programa, con cualquier propósito (libertad 0).

- La libertad de estudiar cómo funciona el programa, y adaptarlo a sus necesidades (libertad 1). El acceso al código fuente es una condición previa para esto.
- La libertad de distribuir copias del programa (libertad 2).
- La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. (libertad 3). El acceso al código fuente es un requisito previo para esto.

Es importante rescatar que "Free Software" no significa "no comercial". Un programa libre debe estar disponible para uso comercial, desarrollo comercial y distribución comercial.

El software libre se refiere a una nueva filosofía respecto al software, donde priman aspectos como especificaciones abiertas y bien común, sobre software cerrado y ánimo de lucro. En el movimiento "Free Software", se evita utilizar términos como: "regalar" o "gratis", porque implican que lo importante es el precio, y no la libertad.

Evolución del Software libre

Existe una gran cantidad de software libre, los observadores y adeptos describen este fenómeno como el movimiento del software libre. Algunos proyectos notables de software libre incluyen los "kernel" de los sistemas operativos GNU-Linux y BSD, los compiladores GCC, el depurador GDB y las bibliotecas de lenguaje de programación C, el servidor de nombres BIND, el servidor de transporte de correo Sendmail, el servidor web Apache, los sistemas de base de datos relacional MySQL y PostgreSQL, los lenguajes de programación Perl, Python, Tcl y PHP, el sistema XWindow, los entornos de escritorio GNOME y KDE, la suite de ofimática OpenOffice.org, el navegador Mozilla, el servidor de ficheros Samba, y el editor de gráficos GIMP.

Los paquetes de software libre constituyen un ecosistema software donde diferentes piezas de programas pueden proporcionar servicios a otras, lo que

permite la co-evolución de características. Por mostrar un ejemplo sencillo, el lenguaje de programación Python proporciona soporte para el protocolo HTTP, y el servidor web Apache que proporciona el protocolo HTTP puede llamar al lenguaje de programación Python para servir contenido dinámico.

En la década de los años 90 ocurrió la explosión de Internet, la “e-década” (e-commerce, e-business, e-publishing, eBay, ETrade, etc.), pero también surgió la necesidad de contar con estructuras abiertas que permitan integrar los sistemas, este acontecimiento produce la aparición, con mucha fuerza en los últimos años, de la “o-década” (open source, open systems, open standard, open access, open archives, – todo abierto-). Esta tendencia se ha revelado con especial intensidad y fuerza en la educación superior y en las herramientas de ofimática.

Esta situación es fuertemente apoyada por el movimiento de “Free Software” y por la iniciativa “Open Source”, por lo cual los productos de software libre, tienen una característica que no se encuentra en los productos propietarios, y es el soporte e implementación de formatos abiertos para almacenar datos, es decir el uso de estándares abiertos.

Iniciativas de Software libre

El software libre en la actualidad es promovido por dos organizaciones sin fines de lucro, que en términos generales comparten objetivos comunes, aunque por diferencias de pensamiento se han separado creando polémicas filosóficas un tanto extremistas. Estas agrupaciones son: La Free Software Foundation, y la Open Source Initiative.

La Free Software Foundation (FSF): Es una organización creada a partir del esfuerzo de Richard Stallman y otros entusiastas del software libre, con el propósito de difundir el movimiento. Está dedicada a eliminar las restricciones sobre el copiado, redistribución, entendimiento, y modificación de programas de computadoras. Hace esto, promocionando el desarrollo y uso del software libre en todas las áreas de la computación, pero muy particularmente, ayudando a desarrollar el sistema operativo GNU/Linux.

Muchas organizaciones distribuyen cualquier software libre que esté disponible. En cambio, la FSF se concentra en desarrollar nuevo software libre, y en hacer de este software, un sistema coherente, para eliminar la necesidad de uso del software no libre.

La Open Source Initiative (OSI): Fue creada en 1998 por un grupo de personas entre ellos Eric S. Raymond y Bruce Perens, colaboradores de la FSF e interesados en el software libre y en el proyecto GNU, quienes decidieron introducir un término de mercadotecnia para el software libre, su objetivo era posicionarlo como amigable para negocios y con menos carga ideológica en su competencia con el software propietario.

La OSI buscaba principalmente lo siguiente:

- Darle mayor relevancia a los beneficios prácticos de compartir el código fuente
- Interesar a las principales casas de software y otras empresas de la industria de la alta tecnología en el concepto.

El significado obvio del término "open source" es "se puede mirar el código fuente", lo cual es un criterio más débil y flexible que el del "Free Software"; un programa de código abierto puede ser software libre, pero también puede serlo un programa no libre.

Los defensores del movimiento consideran que el término "open source" evita la ambigüedad del término "free" en "free software". Sin embargo, aunque se elimina la ambigüedad de Libertad frente a Precio (en el caso del Inglés), introduce una nueva entre los programas que se ajustan a la Open Source Definition (OSD), que dan a los usuarios la libertad de mejorar su código fuente, y programas con código fuente disponible, que posiblemente tienen fuertes restricciones sobre el uso del código fuente.

Se cree que cualquier software con código fuente disponible es "open source", pues se puede manipular (un ejemplo de este tipo de software sería el popular

paquete de software gratuito Graphviz, inicialmente no libre, pero que incluía el código fuente, aunque luego AT&T le cambió la licencia). Sin embargo, mucho de este software no da a sus usuarios la libertad de distribuir sus modificaciones, restringe el uso comercial, y los derechos de los usuarios.

El movimiento "free software" de la FSF enfatiza en los aspectos morales o éticos del software, viendo la excelencia técnica como un producto secundario deseable de su estándar ético. El movimiento "open source" de la OSI ve la excelencia técnica como el objetivo prioritario, la compartición del código fuente es el medio para dicho fin. Por dicho motivo, la FSF se distancia tanto de la OSI, como del término "Open Source".

Aunque en la práctica el software "open source" y el "free software" comparten las mismas licencias, la FSF opina que el movimiento "open source" es filosóficamente diferente del movimiento "free software".

Puesto que la OSI sólo aprueba las licencias que se ajustan a la OSD, la mayoría de la gente lo interpreta como un esquema de distribución, e intercambia libremente "open source" con "free software". Aun cuando existen importantes diferencias filosóficas entre ambos términos; especialmente en cuanto a las motivaciones para el desarrollo y el uso de tal software, raramente tienen impacto en el proceso de colaboración.

El Open Source Software (OSS) es software para el que su código fuente está disponible públicamente, aunque los términos de licenciamiento específicos varían respecto de lo que se puede hacer con el código.

"Open source" y "free software" son esencialmente lo mismo, la diferencia radica en que los defensores del "free software" argumentan que el mercado corporativo antepone la utilidad a la libertad, a la comunidad y a los principios y por ende no va de la mano con la filosofía pura detrás del "free software".

Por otra parte, los seguidores del software "open source" sostienen que el proceso normal de crecimiento de la tendencia debe llegar al mercado corporativo y no seguir escondida bajo el manto de la oposición. Para lograrlo, creen en la

necesidad de un software “open source” más confiable que el software propietario.

Clásicos del Software libre

El software libre, cubre la mayoría de los dominios de las aplicaciones informáticas. Gracias a su calidad y apertura, muchos programas libres son líderes en su mercado. A continuación se detallan los más populares.

- Linux, sistema operativo
- Apache, el servidor web más usado a nivel mundial
- PostgreSQL, base de datos relacional con características avanzadas
- MySQL : base de datos relacional muy difundida en aplicaciones de software libre
- El JBoss es un J2EE application server.

Las características del software libre

Raymond (2001) describe los aspectos técnicos y psicológicos presentes en el software libre y sus protagonistas. Se observa que en el mundo del desarrollo de sistemas hay dos estilos fundamentalmente opuestos: el modelo catedral, que es utilizado por los fabricantes de software propietario y el modelo bazar del mundo Linux (software libre).

Raymond (2001) se refiere al software libre en los siguientes términos:

¿Quién hubiera pensado hace apenas cinco años que un sistema operativo de talla mundial surgiría, como por arte de magia, gracias a la actividad hacker desplegada en ratos libres por varios miles de programadores diseminados en todo el planeta, conectados solamente por los tenues hilos de la Internet?

Se pensaba que el software de mayor envergadura (sistemas operativos y

herramientas realmente grandes) requería construirse como las catedrales, es decir, que debía ser cuidadosamente elaborado por genios o pequeñas bandas de magos trabajando encerrados a piedra y lodo, sin liberar versiones beta antes de tiempo.

El estilo de desarrollo de Linus Torvalds ("libere rápido y a menudo, delegue todo lo que pueda, sea abierto hasta el punto de la promiscuidad") No se trata de ninguna forma reverente de construir la catedral. Al contrario, la comunidad Linux se asemejaba más a un bullicioso bazar de Babel, colmado de individuos con propósitos y enfoques dispares, de donde surgiría un sistema estable y coherente únicamente a partir de una serie de artilugios.

Los puntos más importantes que caracterizan al desarrollo del software libre, según Raymond(2001), son los siguientes:

1. Todo buen trabajo de software comienza a partir de las necesidades personales del programador.
2. Tratar a los usuarios como colaboradores es la forma más apropiada de mejorar el código, y la más efectiva de depurarlo.
3. Libere rápido y a menudo, y escuche a sus clientes.
4. Dada una base suficiente de desarrolladores asistentes y beta-testers, casi cualquier problema puede ser caracterizado rápidamente, y su solución ser obvia al menos para alguien.
5. La perfección (en diseño) se alcanza no cuando ya no hay nada que agregar, sino cuando ya no hay algo que quitar.

Weinberg (1998) en su discusión de la "programación sin ego", señala que en los lugares donde los desarrolladores no tienen propiedad sobre su código, y estimulan a otras personas a buscar errores y posibles mejoras, son los lugares donde el avance es dramáticamente más rápido que en cualquier otro lado.

A largo plazo posiblemente triunfe la cultura del software libre, no porque la cooperación es moralmente correcta o porque la "apropiación" del software es moralmente incorrecta, sino simplemente porque el mundo comercial no puede ganar una carrera de armamentos evolutiva a las comunidades de software libre.

Otras razones del software libre

Según Raymond(2005) el desarrollo de programas informáticos de software libre no tiene nada que ver con el idealismo de algunos iluminados, se debe principalmente a varias razones entre las que destaca las siguientes:

Razones económicas: Se trata de un nuevo modelo de producción económico. El desarrollo es preferible para el consumidor que se beneficia de programas informáticos baratos, estables, evolutivos, asegurados y respetuosos de las normas de intercambio. Además ofrece características favorables para los productores de software, con las cuales se logra lo siguiente:

- Compartir sus costos con otros (reciclaje)
- Aprovechar la mano de obra voluntaria
- Mejorar sus normas de calidad

El razonamiento económico es reforzados por motivos políticos.

Razones políticas:

- Conflictos entre Estados (China, Estados Unidos, América Latina y la Union Europea)
- Preocupación de autonomía en relación con un monopolizador (Microsoft)
- Preocupación de fluidez de la comunicación con los ciudadanos (normas abiertas)

- Necesidad de perpetuidad de la información (protocolos y formatos no propietarios)

Razones sociales

- Voluntad de desarrollar la mano de obra local (servicios, soporte)
- Deseo de transferir hacia una sociedad de servicios
- Preocupación por garantizar el acceso de todos los ciudadanos a la información
- Proyectos de ayuda a los países en vías de desarrollo.

Razones psicológicas: El software libre contempla también factores psicológicos de motivación para los productores.

- Reputación, esto favorece la calidad de los productos libres
- Ofrece a los productores oportunidades de reconocimiento
- El trabajo se expone en el gigantesco espacio público de la Internet

Regulación legal de la producción y distribución de software

El término propiedad intelectual tiene varias acepciones según el contexto y quién lo utiliza. Hoy se utiliza en muchos foros para agrupar distintos privilegios que se otorgan sobre bienes intocables con valor económico. Uno de los más importantes es el copyright (derechos de autor) y similares, que protegen de la copia no autorizada los trabajos literarios o artísticos, programas de ordenador, recopilaciones de datos, diseños industriales, entre otros: Las marcas, que protegen símbolos; las indicaciones geográficas, que protegen denominaciones de origen; los secretos industriales, los cuales respaldan la ocultación de información; y las patentes, que otorgan monopolios temporales sobre invenciones a cambio de desvelarlas.

Sin embargo, en muchas tradiciones legales como la hispana, se distingue

entre la propiedad intelectual (derechos de autor) y la propiedad industrial (las figuras restantes). En cualquier caso, la legislación que se aplica en todos estos aspectos es la más coordinada en prácticamente todo el mundo. Por un lado la Organización Mundial de la Propiedad Intelectual (OMPI, WIPO según sus siglas en inglés) promueve ambos tipos de propiedad en todos sus aspectos. Por el otro, el acuerdo de aspectos comerciales de la propiedad intelectual (ACPI) establece unos mínimos de protección y obliga a todos los países miembros de la Organización Mundial del Comercio, (OMC) a desarrollarlos en ciertos plazos que dependen del nivel de desarrollo del país.

En ONU (1998) se reconoce en su artículo 27 el derecho a que se proteja los intereses morales y materiales que correspondan a cualquier persona por razón de las producciones científicas, literarias o artísticas de que sean autores. Sin embargo, en muchos casos (y de forma habitual en el caso del software), este derecho suele ser transferido en la práctica a las empresas que emplean a los creadores o que comercializan sus creaciones. No obstante, la propiedad intelectual se justifica no sólo por razones morales, sino también prácticas, para dar cumplimiento a otro derecho: el de la sociedad a beneficiarse de las creaciones, se incentivan con beneficios y protegiendo las inversiones para la creación, investigación y desarrollo. Para armonizar ambos derechos, la propiedad intelectual es temporal, caduca cuando ha cumplido su función de promoción.

Pero la caducidad no es la única característica que diferencia la propiedad intelectual de la ordinaria. Los objetos de esta pueden copiarse fácil y económicamente, sin pérdida de calidad. La copia no perjudica a quien ya disfruta de lo copiado, al contrario del robo, que sí priva del objeto al poseedor original.

El control de la copia de intocables es mucho más complicado que del robo de bienes tocables y puede llevarnos a una sociedad policial, que necesite controlar todas las copias de información, y a una gran inseguridad jurídica, porque aumentan las posibilidades de violación accidental de derechos. Además la creatividad es incremental: al crear siempre se copia algo, y la línea divisoria entre la copia burda y la inspiración es sutil.

Tradicionalmente en el mundo del software propietario se ha utilizado este marco legal para garantizar que el fabricante del software – que tiene derechos exclusivos sobre el programa informático - pueda bloquear el acceso al código fuente de sus programas, impedir a la competencia desarrollar productos similares, y limitar los derechos que los usuarios tienen sobre el programa, como copia o modificación.

A la inversa, en el mundo del software libre, se utiliza el marco legal de los derechos de autor para colectivizar los derechos que el autor tiene sobre el software, incluyendo los derechos a copia, uso, modificación y redistribución.

Sin entrar en detalles que quedan fuera del objeto de este artículo, el software libre propone un punto de equilibrio nuevo en este ámbito, primando los beneficios de la copia y la innovación incremental frente al control exclusivo de una obra por su autor.

Definición de CopyRight

Los derechos de autor (copyright) protegen la expresión de un contenido, no el contenido en sí mismo. Se desarrollaron para recompensar a los autores de libros o de arte. Las obras protegidas pueden expresar ideas, conocimientos o métodos libremente utilizables, pero se prohíbe reproducirlas sin permiso, total o parcial, con o sin modificaciones.

Esta protección es muy sencilla, ya que entra automáticamente en vigor en el momento de publicación de la obra con ámbito casi universal. Modernamente se ha extendido a los programas de ordenador y a recopilaciones de datos.

La ley de la propiedad intelectual (LPI) en Costa Rica, y leyes similares en otros países, desarrolladas con base en el Convenio de Berna para la protección de trabajos literarios y artísticos de 1886, regulan los derechos de autor.

Estos derechos se dividen en derechos morales y patrimoniales. Los primeros garantizan al autor el control sobre la divulgación de su obra, con nombre o seudónimo, el reconocimiento de autoría, el respeto a la integridad de la obra y el derecho de modificación y retirada. Los segundos dan derecho a explotar económicamente la obra y pueden ser cedidos total o parcialmente, de forma exclusiva o no, a un tercero. Los derechos morales son vitalicios o indefinidos, mientras que los patrimoniales tienen una duración bastante larga (30 años después de la muerte del autor, si es una persona física). La cesión de derechos se especifica por un contrato denominado licencia. En el caso de programas propietarios, éstos generalmente se distribuyen por medio de licencias de uso no exclusivo, que se entiende que se aceptan automáticamente al abrir o instalar el producto. Estas licencias estipulan lo que los usuarios pueden hacer con el programa en cuanto a: uso, modificación, redistribución, y en qué condiciones. Es la legislación sobre propiedad intelectual la que impide que legalmente se pueda, por ejemplo, vender copias de un programa que se tenga (aunque se haya comprado). Por ello, el modelo de software propietario, es posible porque los estados han puesto en vigor una legislación que está diseñada para ello.

Estrictamente hablando, la situación con respecto de los programas libres no es muy diferente: también se distribuyen bajo licencia. La diferencia es precisamente lo que permite esa licencia. En el caso de las licencias de programas libres, no restringen lo relacionado al uso, la modificación y la redistribución; lo que imponen son condiciones a satisfacer precisamente en caso de que se quiera redistribuir el programa. Por ejemplo: pueden exigir que se respeten las indicaciones de autoría, o que se incluya el código fuente si se quiere redistribuir el programa listo para ejecutar.

Aunque en esencia software libre y software propietario se diferencian en la

licencia con la que los autores distribuyen sus programas, es importante hacer hincapié que esta diferencia se refleja en condiciones de uso y redistribución totalmente diferentes. Como se ha visto en el transcurso de los últimos años, esto ha originado no sólo métodos de desarrollo totalmente diferentes, sino formas prácticamente opuestas (en muchos sentidos) de entender la informática.

Las condiciones y/o restricciones que imponen las licencias sólo pueden ser precisadas por los propios autores, según la normativa de propiedad intelectual son los propietarios de la obra. En cualquier caso, la propiedad de la obra será de los autores, ya que la licencia no supone transferencia de propiedad, sino solamente derecho de uso y, en algunos casos, de distribución. Por ello, los autores de software libre en ningún caso renuncian a la propiedad de su creación, sino que permiten a terceros, el uso, la modificación y la redistribución.

Un tema todavía relativamente abierto es la licencia que cubre a las contribuciones externas. Generalmente se supone que una persona la cual contribuya a un proyecto acepta de facto que su contribución se ajuste a las condiciones especificadas por la licencia del proyecto, aunque esto podría tener poco fundamento jurídico. La iniciativa de la FSF de solicitar mediante carta (física) la cesión de todos los derechos de copyright a cualquiera que contribuya con más de diez líneas de código a un subproyecto de GNU es una buena muestra de que en el mundo del software libre hay políticas más estrictas respecto a las contribuciones.

El símbolo del "copyright ©" es usado para indicar que una obra está protegida por la ley de derecho de autor. Generalmente le da al dueño de la propiedad intelectual de la obra el derecho exclusivo para hacer y para autorizar a otros a:

- Reproducir la obra en copias o fonogramas.
- Preparar obras derivadas basadas en la obra.
- Distribuir copias o fonogramas de la obra al público, vender o hacer otro tipo de transferencias de propiedad tales como: alquilar, arrendar o prestar dichas copias.

- Presentar la obra públicamente: en el caso de obras literarias, musicales, dramáticas y coreográficas, pantomimas, películas y otras producciones audiovisuales.
- Mostrar la obra públicamente, en el caso de obras literarias, musicales, dramáticas coreográficas, pantomimas, obras pictóricas, gráficas y esculturales, incluyendo imágenes individuales de películas u otras producciones audiovisuales.
- En el caso de grabaciones sonoras: interpretar la obra públicamente a través de la transmisión audio digital.

El "CopyRight" sobre una obra creada se convierte inmediatamente en propiedad del autor que creó dicha obra. Sólo él o aquellos cuyo derechos derivan del autor pueden reclamar propiedad.

En diciembre de 1996 fue concluido en Ginebra, El tratado Tratado de la OMPI sobre Derecho de Autor, el cual entró en vigencia en marzo del 2002.

En OMPI (1996) el tratado menciona los temas que son protegidos por los derechos de autor:

1. Programas de computadora: Cualquiera que sea la modalidad o la forma de su expresión.
2. Compilaciones de datos u otros materiales "bases de datos": En cualquier forma, que en virtud de la selección o arreglo de su contenido constituyan creaciones intelectuales.

En lo referente a los derechos de autor, el tratado se ocupa de los siguientes:

1. El derecho de distribución
2. El derecho de alquiler
3. El derecho de comunicación al público.

Cada uno de ellos es un derecho exclusivo, sujeto a ciertas limitaciones y

excepciones.

El tratado obliga a cada una de las partes contratantes a adoptar las medidas necesarias, de acuerdo con su sistema legal, para garantizar la aplicación de dicho tratado. En particular, la parte contratante deberá asegurarse de que su ley incluya procedimientos que garanticen el cumplimiento, de modo que pueda instruirse una acción legal eficaz contra cualquier infracción de los derechos cubiertos por el tratado. Esa acción debe incluir medios expeditos para prevenir la infracción y “castigos” que sean un factor disuasorio contra futuras transgresiones.

Definición de CopyLeft.

CopyLeft describe un movimiento y un ideario que promueve la flexibilización del copyright y los derechos de autor. El término CopyLeft se asocia a un conjunto de licencias, aplicadas a creaciones como el software y obras artísticas, para permitir que esas obras sean utilizadas, modificadas, copiadas y redistribuidas libremente.

El término CopyLeft nace en el mundo de la programación, concretamente fue creado por la FSF y el movimiento "Free Software", ámbito en el cual constituye un método para garantizar que un programa de software libre se mantenga siempre libre. Esto obliga que todas las modificaciones y versiones extendidas del programa mantengan las libertades otorgadas a los usuarios. De forma análoga este concepto se aplica también a todo tipo de conocimiento libre (textos, fotos, vídeos, etc.).

Es un término reciente acuñado por el Proyecto GNU. Nace de un juego de palabras en inglés, en oposición a “copyright”. No se restringe la redistribución, con o sin cambios, sino por el contrario se permite mientras se garanticen las libertades promovidas por la FSF.

Definición licencia de software

En Wikipedia (2004) la definen:

“Una Licencia de software es la autorización o permiso concedida por el titular del derecho de autor, en cualquier forma contractual, al usuario de un programa informático, ya sea sistema operativo o un programa aplicativo específico, para utilizar éste en una forma determinada y de conformidad con las condiciones convenidas”.

La licencia, puede ser gratuita o tener un precio. Precisa los derechos de uso, modificación y/o redistribución concedidos a la persona autorizada y sus límites. Además, puede señalar el plazo de duración, el territorio de aplicación y todas las cláusulas que el titular del derecho de autor establezca.

Cuando la licencia no es concretamente negociada entre proveedor y el usuario de un programa informático, para asegurar que el usuario tenga conocimiento y acepte las condiciones de la misma, los proveedores obligan a aceptar la licencia al momento de instalar el programa - licencia click-wrap - o, en algunos programas propietarios, por el simple hecho de abrir el sobre o paquete que lo contiene - licencia shrink-wrap - , en ambos casos se discute la validez legal aún hoy en día.

Básicamente podemos hablar de dos grandes grupos de licencias de software: las licencias de software propietario y las licencias de software libre.

Licencias propietarias

En este tipo de licencia el dueño de los derechos del software insta al usuario final a que reconozca tener conocimiento de las restricciones de derechos de autor, patentes, etc. y acepte un contrato general de uso.

Las compañías utilizan este tipo de contratos para tener un mayor control del uso de sus productos y como un documento único de aceptación cuando los componentes del producto están regulados por más de un contrato o licencia.

A pesar de que este tipo de contratos se encuentran regulados en la mayor parte de las veces al menos por la legislación del país de origen del producto y a veces también por las legislaciones locales, estas licencias han sido muy atacados por diversas razones entre las que destacan:

- El conocimiento del contenido de los contratos es difícil antes de realizar la compra, pues las cajas de los productos raramente contienen una copia completa. El comprador, en la mayor parte de los casos, conoce su contenido después de realizar la compra.
- En ocasiones se exige al usuario renunciar a realizar reclamos o demandas legales por diversos motivos tales como posibles daños producidos por el producto (o fallos en el) o aceptar la restitución de estos en la forma y monto que la compañía decida.
- Este tipo de acuerdo expresa que tipos de usos se pueden dar y cuales no al producto ya que quien lo compra no es legalmente en ninguna forma dueño del producto sino de una licencia para su uso. Esto último limita los derechos del consumidor
- Este tipo de acuerdos son unilaterales pues el usuario no tiene más opción que aceptar o rechazar el contenido del mismo (en algunos países existen organizaciones de protección al consumidor que previamente los autorizan).

En el software bajo licencias de este tipo el usuario básicamente tiene derecho de utilizar el programa de la manera que lo estipula la licencia. La redistribución y modificación están prohibidas. Un ejemplo de este tipo de licencia es la End User License Agreement (EULA por sus siglas en inglés), que es utilizada para distribuir algunos de los productos de software desarrollado por la empresa Microsoft.

Dentro del software propietario existen algunas formas de distribuir software, que a menudo causan confusión y son relacionados incorrectamente con el software libre, seguidamente se mencionan los más conocidos y significativos.

FreeWare: Es software de computadora que se distribuye sin costo para el usuario. A veces incluye el código fuente, pero no es usual.

El freeware suele incluir una licencia de uso que permite su redistribución; pero

con algunas restricciones: como no modificar la aplicación en sí, ni venderla, y dar cuenta a su autor. También puede desautorizar el uso en una compañía con fines comerciales o en una entidad gubernamental.

El término fue acuñado en 1982 por Andrew Fluegelman, que quería distribuir un programa que había escrito, llamado PC-Talk, pero con el que no deseaba usar métodos tradicionales de distribución de software. Fluegelman registró el término freeware, pero esos derechos de autor ya han sido abandonados. De hecho, el método que usó para distribuir PC-Talk hoy se denomina shareware.

Algunos desarrolladores liberan versiones freeware de sus productos para obtener un número de seguidores que puedan estar dispuestos a pagar por una versión más completa, o porque es una edición obsoleta de un programa.

Otros lo hacen porque no consideran que ese programa pueda generar una ganancia económica, o porque creen en el valor de ofrecer algo gratis.

ShareWare: Es software protegido por leyes de copyright, que se encuentra disponible gratuitamente durante cierto tiempo para su evaluación por el usuario. Una vez transcurrido este plazo, el programa expira. La forma de poder utilizarlo es comprando una licencia del producto.

Licencias de software libre

La variedad de licencias de software libre es grande; sin embargo, es posible dividir las en tres grandes categorías. La primera está compuesta por licencias del tipo GNU|GPL, que son referenciadas como licencias robustas o licencias con CopyLeft. Estas licencias contienen condiciones que exigen que las versiones modificadas del programa o programas resultantes de la combinación de programas amparados a este tipo de licencia, mantengan las mismas libertades de la licencia original, no permite restringir ningún derecho a terceros. Esto evita que la industria del software propietario pueda crear software no libre a partir del software libre.

Algunas de las licencias de software libre que incluyen esta condición son las

siguientes:

- General Public License(GNU|GPL)
- Lesser General Public License (GNU|LGPL)
- Artistic License
- Common Public License v.1.0.

La segunda categoría está compuesta por las licencias que no imponen condiciones especiales en la segunda redistribución (esto es, que sólo especifican que el software se puede redistribuir o modificar, pero no imponen condiciones especiales si se hace. Permite por ejemplo, que alguien quien reciba el programa pueda después redistribuirlo como software propietario). Este tipo de licencias son generalmente referenciadas como licencias permisivas, licencias sin CopyLeft o licencias con CopyLeft débil. Estas licencias presentan el problema que algunas copias o versiones modificadas pueden no ser libres completamente.

Puede argumentarse que estas licencias aseguran “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 orientadas a destacar este tipo de licencia debido principalmente a que no contribuye al desarrollo de más software libre.

Dentro de las licencias más conocidas están las siguientes:

Licencias estilo BSD: llamadas así porque se utilizan en gran cantidad de software distribuido junto a los sistemas operativos BSD. El autor, bajo tales licencias, mantiene la protección de copyright únicamente para la renuncia de garantía y para requerir la adecuada atribución de la autoría en trabajos derivados, pero permite la libre redistribución y modificación, incluso si dichos trabajos tienen propietario. Son muy permisivas, tanto que son fácilmente absorbidas al ser mezcladas con la licencia GNU|GPL con quienes son compatibles.

Licencias estilo MPL y derivadas: Esta licencia tiene un gran valor para el movimiento "Free Software", porque fue el instrumento que empleó Netscape Communications Corp. para liberar Netscape Communicator 4.0 y empezar ese proyecto tan importante para el mundo del Software Libre: Mozilla. Se utilizan en gran cantidad de productos de software libre de uso cotidiano en todo tipo de sistemas operativos.

La Mozilla Public License (MPL) promueve eficazmente la colaboración evitando el efecto "viral" de la GPL. No obstante la MPL no es tan excesivamente permisiva como las licencias tipo BSD.

La Netscape Public License (NPL luego MPL) fue la primera licencia nueva después de muchos años, que se encargaba de algunos puntos que no fueron tenidos en cuenta por las licencias BSD y GNU|GPL. En el espectro de las licencias de software libre se considera adyacente a la licencia estilo BSD; pero perfeccionada.

Algunas de las licencias que pertenecen a esta tipo son:

- Academic Free License v.1.2.
- Apache Software License v.1.1.
- Artistic.
- Attribution Assurance License.
- BSD License.
- MIT License.
- University of Illinois/NCSA Open Source License;
- W3C Software Notice and License.
- Mozilla Public License.

Mientras que las licencias sin CopyLeft enfatizan en la libertad de quien recibe

un programa, ya que le permite hacer casi lo que quiera con él (en términos de condiciones de futuras redistribuciones), Las licencias con CopyLeft lo hacen en la libertad de cualquiera que potencialmente pueda recibir algún día un trabajo derivado del software original (ya que obliga a que las sucesivas modificaciones y redistribuciones respeten los términos de la licencia original).

La tercera categoría de licencias de software libre son referenciadas generalmente como Licencias Dual. Con este tipo de licencias el autor cede su creación bajo dos licencias diferentes según el uso que se vaya a hacer del software y las libertades y obligaciones que se desean. Bajo este modelo, una empresa puede ofrecer una versión libre de su programa y otra con condiciones más óptimas para las necesidades de sus clientes que quieran trabajar fuera del modelo del software libre. Este modelo es utilizado por muchas empresas ya que permite ofrecer los productos de forma libre y no libre, según las necesidades de cada cliente.

Esta técnica ha sido usada como un modelo de negocio por una serie de empresas que desarrollan software libre (por ejemplo MySQL); esta práctica no restringe ninguno de los derechos otorgados a los usuarios.

La diferencia entre las licencias con CopyLeft y las licencias sin CopyLeft es tema de debate en la comunidad del software libre, de cualquier forma, es conveniente recordar que todas ellas son licencias de software libre.

El software de dominio público es un caso especial de software libre no protegido con CopyLeft. Algunas veces se utiliza el término “dominio público” de una manera imprecisa para decir “libre”, “disponible, gratis”. Sin embargo, “dominio público” es un término legal y significa de manera precisa “sin copyright”. En este tipo de licencias el autor ha abandonado sus derechos de autor. El software bajo el dominio público, puesto que carece de protección de copyright alguno, puede ser incorporado libremente tanto en software libre como no libre.

Cláusulas de no responsabilidad

Todas las licencias de software, libres o propietarias, incluyen una limitación de

garantía que en realidad es una negación de garantía, necesaria para evitar demandas legales por garantías implícitas. Aunque se ha criticado mucho esta técnica en el software libre, es práctica habitual en el software propietario. Generalmente sólo garantiza que el soporte es correcto y el programa en cuestión se ejecuta.

Es importante destacar dos características especiales de estas cláusulas en el caso del software libre:

- En la mayoría de los casos el usuario no recibe un programa libre como resultado de una transacción comercial directa o indirecta con el productor. Por ello, parece razonable que el autor no tenga responsabilidad sobre el uso que pueda hacer cualquier persona del programa, si no ha tenido ningún tipo de relación comercial con él, ni contraprestación económica.
- Quien quiera puede, si lo desea, proporcionar servicios con cualquier tipo de garantía sobre un programa de software libre. Es factible hacerlo no sólo porque el código fuente esté disponible para su redistribución, sino también porque permite inspeccionar y modificar el código, por ejemplo: para resolver un error que presente. Una empresa especializada podría decidir, después de estudiar y estabilizar el programa a su gusto, ofrecerlo de forma garantizada a sus clientes (que quizás estarían dispuestos a pagar por ello una cantidad que permita a esa empresa sufragar el estudio y estabilización).

Seguidamente se define el concepto de ofimática, para brindar la información requerida sobre el tema en estudio del presente trabajo.

Definición de ofimática

En la Universidad Nacional de Colombia (1992), se destacan las siguientes definiciones:

“Emplear tecnología para manejar información usada en la oficina con el objeto de mejorar el contenido, formato y cantidad de trabajo realizado”.

"La ofimática es la utilización de ordenadores en la oficina como soporte a los trabajadores de la información que no son especialistas en ordenadores".

"La utilización de tecnología para mejorar la realización de funciones de oficina".

"La automatización de oficinas se refiere a la utilización de sistemas integrados de ordenadores y comunicaciones, como soporte a los procedimientos administrativos en un entorno de oficina"

Ampliando las definiciones anteriores una suite ofimática, es un paquete integrado de programas informáticos diseñados para el trabajo de oficina, distribuido en forma conjunta bajo los términos de determinado tipo de licencia, posee una interfaz homogénea, y cuyas aplicaciones son capaces de interactuar mutuamente en un nivel superior al que permitiría el sistema operativo.

No existe una norma estricta sobre los programas a incluir en una suite ofimática, pero en la mayoría encontramos un procesador de texto, una hoja de cálculo, un programa de presentaciones. De forma añadida, puede contener un sistema gestor de base de datos, un programa de organización (agenda), un cliente de correo, un navegador web, y programas menores de gráficos y comunicaciones.

El software de ofimática se encuentra disponible bajo diversas presentaciones, versiones y productores en el mercado. Puede ser usado no sólo en empresas y oficinas, sino que también resulta útil e incluso indispensable en ambientes académicos y caseros, donde puede potenciar y facilitar muchas actividades cotidianas (solución de problemas matemáticos y estadísticos, presentación de documentos y trabajos escritos, diseño y presentación de exposiciones mediante diapositivas, almacenamiento de información en bases de datos, planeación de actividades, administración de correo, etc.).

Para ilustrar la gran variedad de suites ofimáticas en el mercado se incluye la siguiente tabla, separándolas de acuerdo con la plataforma soportada:

Multiplataforma (Windows, Macintosh, Unix/Linux)	
OpenOffice.org	(Writer, Calc, Impress, Draw). Suite ofimática de código abierto y

	gratuita que utiliza el estándar XML.
Gnome Office	(Abiword, Gnumeric, GNOME-DB). Suite ofimática de código abierto y gratuita.
StarOffice	Suite basada en OpenOffice.org.
CorelWordPerfect Office	(WordPerfect, Quattro Pro, etc)
Lotus SmartSuite	(Word Pro, Lotus 1-2-3, etc.)
Papyrus Office	(Papyrus WORD, Papyrus BASE)
Macintosh y Windows	
Microsoft Office	(Word, Excel, PowerPoint, Access, Outlook, etc.)
ThinkFree Office	
Macintosh	
NeoOffice	Porte de OpenOffice.org para Mac OS X.
MarinerPack	
iWork	
AppleWorks.	Suite para uso doméstico.
Windows	
EasyOffice	
Works	Suite para uso doméstico.
Unix/Linux	
Siag Office	

Fuente:"http://es.wikipedia.org/wiki/Suite_ofimática"

Un aspecto que será desarrollado más adelante, es la forma en que los datos producidos por las aplicaciones de ofimática son almacenados en los ficheros o archivos, y la interoperabilidad que ofrecen estos formatos con otros productos de software, esto a fin de garantizar aspectos funcionales que son obviados, los cuales perjudican directamente a terceros, pues imponen el uso de herramientas específicas.

Definición formato de almacenamiento

En Wikipedia (1998) es descrito de la siguiente manera:

“En informática, un formato de almacenamiento es una secuencia finita de operaciones realizables, no ambiguas, cuya ejecución da solución del problema de almacenar datos en memoria”.

Existen tres categorías de formatos de almacenamiento, Los formatos

cerrados, los formatos publicados y los formatos abiertos.

Formato cerrado

Los derechos del formato pertenecen a una o varias empresas, su especificación no está publicada, o lo está parcialmente, sólo el software producido por las compañías dueñas de la especificación, podrá ser capaz de leer completa y en forma correcta la información contenida en los archivos. Los formatos cerrados son generalmente protegidos por patentes y los dueños exigen regalías por el uso o implementación de los formatos en software de terceros. Habitualmente se requiere de técnicas de ingeniería reversa para que otras herramientas lo utilicen plenamente. Dentro de los formatos cerrados más conocidos podemos citar los formatos doc, xls, ppt que son utilizados en la suite ofimática Office de Microsoft.

Formato publicado

Los derechos pertenecen a una o varias empresas, su especificación está publicada totalmente, y existe la posibilidad de utilizarlo por parte de una aplicación no perteneciente a la empresa propietaria. El uso del formato no está restringido. El futuro del formato, al igual que en el caso de los formatos cerrados, depende de los avatares y decisiones de la empresa propietaria. Dentro de los formatos publicados más conocidos están: pdf, ps, rtf.

Formato abierto

Pertenece a alguna organización sin fines de lucro, cuyo propósito es establecer normas para la interoperabilidad del software. Dicha organización fomenta su uso por parte del público en general, sin esperar retribución por ello. Publicada completamente la especificación, y habitualmente existe una implementación de referencia que ayuda a desarrollar software totalmente compatible. Los formatos abiertos más conocidos son: html, sxw, sxc.

Estos formatos pueden ser implementados por programas propietarios o libres, usando las licencias típicas de cada uno. En contraste, los formatos propietarios

son controlados y definidos por intereses privados.

La relación entre los formatos abiertos y el software libre es con frecuencia mal comprendida. Algunos programas propietarios usan formatos abiertos, y algunos programas abiertos utilizan formatos propietarios. HTML por ejemplo, el lenguaje abierto de formato de texto de Internet, es la base en que se apoyan navegadores propietarios como el Internet Explorer de Microsoft, así como navegadores libres como Mozilla Firefox. Análogamente, OpenOffice, una suite ofimática libre, puede manipular los formatos propietarios DOC, XLS y PPT de Microsoft, así como los correspondientes formatos abiertos ODT, ODS y ODP de OpenDocument.

Algunos fabricantes comerciales publican sus especificaciones para que puedan ser implementadas en otros programas, incluyendo desarrolladores de software libre, como hizo Adobe Systems con su formato Portable Document Format (PDF), o Microsoft con su formato Rich Text Format (RTF).

La meta fundamental de los formatos abiertos es garantizar el acceso a largo plazo de los datos almacenados sin la incertidumbre actual o futura respecto a los derechos legales de uso de la tecnología de acceso, y la disponibilidad de esta tecnología, o a la especificación técnica del formato de almacenamiento de los datos. Una meta secundaria común de los formatos abiertos es fomentar la competencia, en vez de permitir que el control de un vendedor sobre un formato propietario inhiba el uso de los productos de la competencia.

El formato OpenDocument

El formato "OpenDocument" para aplicaciones ofimáticas de OASIS, es un formato de fichero abierto estándar para el almacenamiento de documentos ofimáticos tales como hojas de cálculo, memorandos, gráficas y presentaciones. Está basado en el esquema Extensible Markup Language (XML).

Este formato pretende ofrecer una alternativa abierta a los formatos de documentos cerrados como los DOC, XLS y PPT que son utilizados por Microsoft Office, o los formatos publicados, como los PDF, PS y RTF.

La motivación principal para usar formatos estándar, es evitar que los usuarios se conviertan en cautivos de un único proveedor de software, permitiéndoles cambiar de entorno informático si su proveedor actual cambia su modelo de licencia en términos menos favorables.

Ya existen versiones de programas que usan el formato OpenDocument, siendo el más conocidos KOffice (a partir de su versión 1.4, liberada en junio de 2005). También la versión 2.0 del conocido OpenOffice.org lo utiliza como formato de ficheros por defecto y la próxima versión 1.1.5 lo podrá usar como filtro. Por su parte, el proyecto AbiWord ha anunciado un filtro de exportación e importación hacia el formato utilizado por éste producto.

Implicaciones de orden público

Puesto que dentro de los objetivos de los formatos abiertos, como OpenDocument, están:

- Garantizar la perdurabilidad de los datos producidos
- Garantizar que los documentos generados pueda ser utilizado en el futuro con el contenido y la calidad original.
- Eliminar las barreras técnicas
- Eliminar las barreras legales

Algunas administraciones públicas y gobiernos han empezado a considerarlo un asunto de política de interés público.

En Massachusetts Government (2005), Eric Kriss, secretario de Administraciones Públicas y Hacienda del Estado de Massachusetts, estableció como uno de los principios de su administración el compromiso de utilizar formatos abiertos en la siguiente declaración:

"Es absolutamente imperativo para el sistema democrático de los EEUU que descontinúemos la práctica de tener nuestros documentos públicos cautivos en

un formato exclusivo, sea éste el que sea, arriesgándonos a que en el futuro el documento sea quizás ilegible o esté sujeto a un sistema de licencias exclusivo que restrinja su acceso."

El 21 de septiembre de 2005, Massachusetts se convirtió en el primer estado norteamericano en aprobar formalmente los diferentes formatos OpenDocument para su uso en los registros públicos, a la vez que se rechazaba el formato basado en XML propuesto por Microsoft, su principal proveedor actual, por no ser considerado abierto. Si Microsoft decide no dar soporte a OpenDocument para el año 2007, fecha límite definida por el Estado, se descalificará de consideración futura por el Estado de Massachusetts.

En mayo del 2003 el Ayuntamiento de Munich (Alemania) aprobó la migración a GNU/Linux y aplicaciones de ofimática libres de todos los ordenadores usados como escritorios, unos 14.000. La decisión de hacerlo no fue sólo económica, también se tomaron en cuenta aspectos estratégicos y cualitativos, según declararon sus responsables. En el estudio que se realizó previamente a la decisión, la solución finalmente elegida (GNU/Linux más OpenOffice, fundamentalmente) obtuvo 6.218 puntos (de un máximo de 10.000) frente a los 5.000 que obtuvo la solución "tradicional" basada en software de Microsoft.

En el año 2003 el gobierno brasileño anunció que utilizaría software libre en prácticamente todas las áreas de su Gobierno, por lo que se convertirá en el primer ejecutivo a escala mundial en llevar a cabo un despliegue masivo de este sistema en la Administración Pública.

La industria informática brasileña mueve más de 3.000 millones de dólares al año, de los que aproximadamente 1.000 millones van a parar a Estados Unidos por concepto de licencias de software. El Gobierno apuesta por la creación de un tejido local de empresas que ofrezcan soluciones y servicios en software libre, para fomentar así, el desarrollo local y desvincular la inversión pública de las grandes multinacionales.

La importancia relacionada a las aplicaciones de ofimática en las iniciativas del

software libre, se da porque estas aplicaciones son el punto de extensión del software libre a un gran número de usuarios.

Otras implicaciones

En la actualidad, la gran difusión de la suite ofimática Microsoft Office ha causado que los formatos de almacenamiento utilizados por esta herramienta, se hayan convertido en la opción utilizada por la mayoría de los usuarios en todo el mundo, lo cual ha provocado un estándar de facto. Sin embargo, se han presentado situaciones de incompatibilidad entre documentos cifrados con distintas versiones del mismo producto, lo que genera la necesidad de invertir reiteradamente en estos productos.

En otros casos, el proceso de evolución ha provocado la desaparición de algunos formatos, y aplicaciones de software que los implementen. Al no disponer de herramientas para el manejo adecuado y completo de un conjunto de documentos, la migración, de ser posible, genera mucho trabajo y se obtienen resultados limitados. Esto ocurre especialmente cuando la especificación del formato no está disponible, o cuando simplemente no existen implementaciones que los contemplen adecuadamente.

Costos licencias de ofimática

Como se ha indicado anteriormente, las razones de costo no son el punto principal del movimiento de software libre, y tampoco debe prevalecer para determinar si una suite ofimática multiplataforma distribuida bajo los términos de alguna licencia de software libre, es conveniente o no. Sin embargo, es necesario contar con datos reales sobre el tema para cuantificar el impacto que representan sobre los costos totales por licenciamiento en una empresa. A manera de ilustración, seguidamente se muestran datos de una compañía local. Es necesario indicar que debido a la existencia de un convenio de confidencialidad con el proveedor de software, se utiliza un nombre ficticio y no se especifica la actividad económica de la empresa.

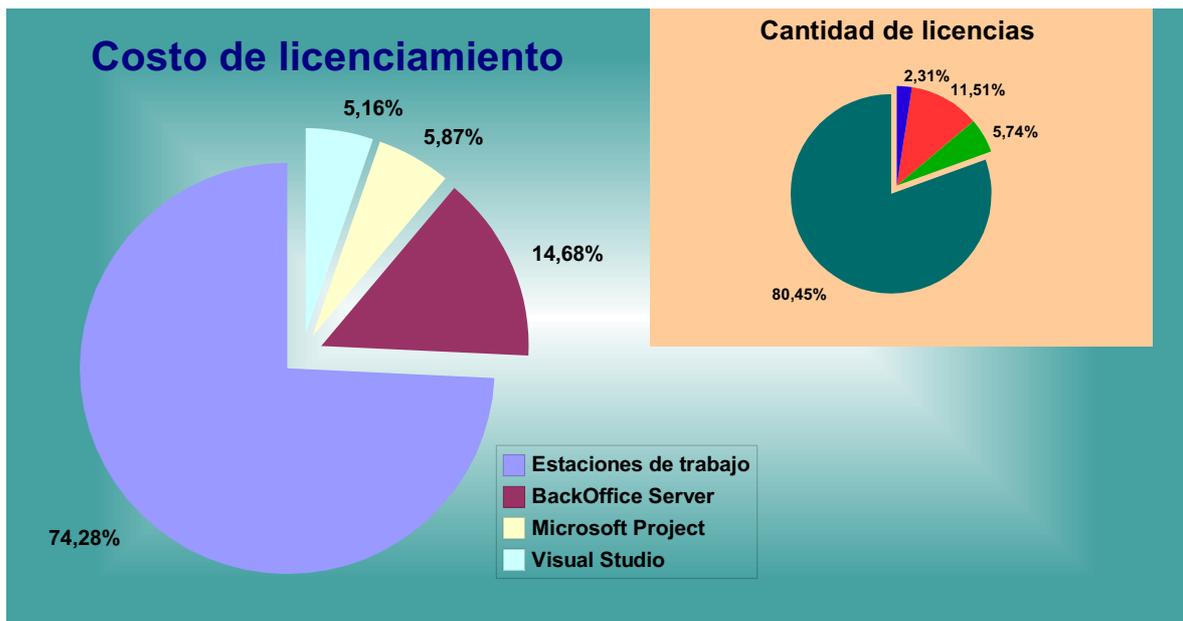
La empresa XYZ S.A suscribió un contrato con la Corporación Microsoft para adquirir por un período de 3 años las licencias de software requeridas para sus labores normales. La empresa invierte la suma de \$761,662.00 al año por concepto de licenciamiento de productos Microsoft, como se muestra en la siguiente tabla:

Producto	Cantidad	Precio unitario	Total
Estaciones de trabajo	2,300	\$246.00	\$565,800.00
BackOffice Server	164	\$682,00	\$111,848.00
Microsoft Project	329	\$136,00	\$44,744.00
Visual Studio	66	\$595,00	\$39,270.00
Total			\$761,662.00

Fuente: Trabajo actual.

El primer paquete de ofimática (Microsoft office 2003) lo constituían las siguientes herramientas: El editor de texto microsoft word, la hoja de cálculo excel, la herramienta de presentaciones powerpoint, la base de datos access, el cliente de correo outlook y la herramienta publisher.

Se proporciona un gráfico sencillo que detalla el porcentaje que representan las licencias de productos de ofimática con respecto a la inversión total por licenciamiento anual. Así como la relación entre la cantidad total de licencias y las que corresponden a productos de ofimática.



Estos datos demuestran que un porcentaje considerable del licenciamiento de la empresa corresponden al software de ofimática.

Los que defienden el uso de software propietario sostienen que el respaldo, la garantía y el soporte ofrecido por compañías como Microsoft, hacen la diferencia; sin embargo, hay que recordar que tanto las licencias de software libre como las licencias propietarias contienen cláusulas de no responsabilidad, las cuales representan una negación de garantía.

Mitos y realidades acerca del software libre

En el mundo de las tecnologías de información, existe una estrategia que consiste en confundir al público dando información incompleta, una visión distorsionada de la realidad, y dejar el resto a la imaginación del receptor del mensaje, con el objetivo de crear miedo, incertidumbre y duda en el momento de tener que evaluar soluciones de la competencia. Esta técnica se denomina Fear, Uncertainty and Doubt (FUD miedo, incertidumbre y duda).

Los objetivos principales del FUD son los usuarios, analistas de mercado, y responsables de informática.

Microsoft y otros fabricantes de la industria han utilizado esta técnica

ampliamente. Por ejemplo, la empresa intentó hacer creer a la industria que se debían usar sus productos Windows NT, en el cliente y servidor juntos, creando incertidumbre, miedo, y duda sobre el hecho de si NT sería capaz de funcionar con otros sistemas operativos.

El software libre ha sido objetivo de ataques de este tipo desde distintos frentes, teniendo como efecto el retraso -en algunos casos- de la adopción de soluciones basadas en software libre.

En octubre de 1998 se filtraron varios documentos internos de Microsoft que describían y analizaban la amenaza que representaba el software libre para la compañía y recomendaba algunas estrategias en la línea FUD.

Microsoft a los pocos días confirmó la autenticidad de los documentos y reconoció que habían sido escritos por algunos de sus empleados; pero indicó, que no representaban la posición oficial de la empresa.

Estos documentos se conocen con el nombre de documentos Halloween, ya que coincidió con esta festividad el día que el primer documento se filtró.

Los documentos Halloween reconocen ya en el año 1998, que el software libre había alcanzado niveles de credibilidad importantes, que tiene una calidad similar o superior al software propietario, y que representa una amenaza para las ventas de Microsoft. Además, hacen constar que según la opinión de Microsoft las técnicas FUD no han tenido gran efecto sobre el software libre y que la mejor estrategia es extender los estándares actuales de forma que sólo Microsoft pueda proporcionar el software que los soporte correctamente. Popularmente se conoce esta técnica como contaminación de estándares.

Para combatir el FUD y sus efectos negativos la mejor estrategia es dar la máxima información sobre el software libre. A continuación se enumeran algunos de los mitos y errores comunes sobre el software libre.

Sólo existe software libre para Linux

Linux es uno de los barcos insignia del movimiento del software libre; pero la

realidad es que hay muchos más programas libres, como por ejemplo: OpenOffice.org, Mozilla, Abiword, GIMP, Apache, MySQL, y otros, que se encuentran disponibles para diferentes sistemas operativos, incluyendo Microsoft Windows y MacOS.

El concepto de software libre no está atado a ningún sistema operativo, de hecho, también es aplicable a la documentación y a toda clase de creaciones artísticas digitales.

No hay soporte para las empresas

En la actualidad, empresas como IBM, Sun, Compaq, Dell, RedHat, Suse, Mandrake, Novell y muchas otras apoyan y dan servicios profesionales a sistemas basados en software libre a usuarios, empresas e instituciones. Por otra parte, el apoyo técnico de empresas de software propietario se acostumbra adquirir como un servicio adicional.

Nadie escribe software gratis

Las motivaciones de los programadores de software libre son diversas. Existen proyectos impulsados solamente por voluntarios, otros vinculados a universidades, vinculados a empresas, y muchos que son combinaciones de los anteriores.

Cada día se escriben líneas de código nuevo y existen más proyectos libres. Por otro lado, son comunes los casos donde un cliente contrata un programador para desarrollar una aplicación que luego libera bajo los términos de licencias libres.

El software libre destruye la industria

En todas las industrias hay factores disruptivos que ocasionan que algunos fabricantes salgan del mercado y aparezcan nuevos. El software libre no destruye la industria; representa un modelo de producción diferente por el cual cada vez más a menudo empresas, usuarios y administraciones tienen interés. Según Libro

Blanco (2005), una iniciativa que hace seguimiento del software libre en el ámbito del sector público, a Noviembre de 2005 habían 454 empresas en España contabilizadas que trabajan y dan servicios basados en software libre.

El software libre destruye la innovación

La comunidad del software libre crea constantemente nuevos programas, mejora los existentes, y crea nueva documentación y material de soporte que son patrimonio de la humanidad y que centros de investigación, universidades, usuarios, y empresas usan en todo el mundo.

Con el software propietario, solamente la empresa dueña del código lo puede mejorar, dejando la innovación en sus manos y dependientes de intereses comerciales. El software propietario crea monopolios, y en tal situación la innovación es menor que en el mundo del software libre, principalmente porque existen una gran variedad de interesados en realizar ajustes innovadores a un producto determinado.

La disponibilidad del código fuente crea versiones incompatibles

El mundo del software libre ha aprendido mucho de los Unix propietarios donde esto fue realmente un problema durante los años 80 y 90. El ejemplo más claro, que ocurre lo contrario, lo demuestra que las diferentes distribuciones de Linux son compatibles entre ellas.

Las empresas competidoras copiarán tu código

El software libre está protegido por los derechos de autor. Si una empresa copia parte del código de una aplicación debe hacerlo respetando la licencia en que el código copiado fue liberado. Si esta licencia es del tipo copyleft, como por ejemplo la GPL, deberá hacer su código también GPL y por lo tanto deberá convertirse en un proveedor de soluciones libres.

El software libre es más inseguro que el propietario

Uno de los informes que han analizado con más profundidad el tema de la seguridad del software libre en comparación al propietario es el presentado por Ross Anderson(2002). El informe concluye que el hecho que el código fuente este disponible, a largo plazo no supone una diferencia por si mismo.

Recomendaciones finales

En (González, Robles, Seone 2003), se detallan las ventajas del software libre con respecto al software propietario; más que ventajas, son condiciones ha evaluar para determinar si una herramienta de este tipo en general, es adecuada para un fin determinado, por lo tanto aplicables en el proceso de selección de una suite ofimática libre y multiplataforma.

Cada uno de estos aspectos pueden ser ampliados y desarrollados, sin embargo, el interés de este trabajo, es promover una adecuada información sobre las herramientas de ofimática distribuidas bajo licencias de software libre y sus características:

- Es necesario documentarse adecuadamente sobre el software libre: las tendencias del modelo, las oportunidades que ofrece, las características, y su filosofía.
- Es conveniente seleccionar productos multiplataforma, esto reduce los impactos del cambio, únicamente aplican los relacionados al producto seleccionado y no a todo el entorno operativo de hardware instalado.
- Realizar evaluaciones reales de los productos antes de adoptarlos.
- Seleccionar herramientas que promuevan verdadera competencia, evitando mercados con tendencia al monopolio, los cuales rigen las condiciones de acuerdo a sus estrategias comerciales.
- No depender del soporte del fabricante del software, permitiendo que múltiples empresas, quizá pequeñas, que dispongan del conocimiento

necesario y del código fuente, puedan crear modelos de negocio manteniendo el software, generando oportunidades de crecimiento para las economías locales .

- No depender de la fiabilidad del fabricante del software para intentar deducir la calidad del producto.
- Minimizar el impacto que ocasiona que el proveedor unilateralmente descontinúe un producto determinado de software.
- Las herramientas de software escogidas permitan, teniendo los conocimientos necesarios, realizar adecuaciones para usos particulares y especiales
- Que existan modelos adecuados para detectar errores en los programas
- Evitar el pago de licencias para productos que son utilizados con poca regularidad.
- Evitar el pago de licencias para contar con la última versión del producto.
- Que las herramientas seleccionadas permitan fácilmente y de manera natural, ser integradas con otro tipo de productos.
- Que sea factible auditar la calidad, la seguridad y la privacidad que ofrece la herramienta seleccionada.
- Que el modelo de negocio de las herramientas seleccionada, proporcionen y promuevan servicios accesibles neutrales para la sociedad y los individuos.
- Que la herramienta seleccionada garantice la integridad, utilidad, privacidad y seguridad de los datos.
- Garantizar la perpetuidad de los datos almacenados.
- Que las herramientas seleccionadas promuevan la colaboración como

mecanismo para desarrollar nuevas soluciones del producto.

- Que las herramientas seleccionadas respeten y utilicen estándares abiertos.
- Aprovechar las oportunidades que brindan los diferentes modelos de negocio relacionadas con la industria del software.
- Seleccionar software que fomente aspectos de responsabilidad cívica y ciudadana, para evitar imponer herramientas a terceros, obligándolos a adquirir licencias de software específicas.

Bibliografía

- Cusumano, Michael A (2004). *The business of Software*, Free press.
- Free Software Foundation (1985). *Philosophy of the GNU Project*. Recuperado el 20 octubre 2005. <http://www.gnu.org/philosophy>
- Free Software Foundation (1991). *GNU general public license, version 2*.
<http://www.fsf.org/licenses/gpl.html>
- Free Software Foundation (1999). *GNU lesser general public license, version 2,1*.
<http://www.fsf.org/licenses/lgpl.html>
- Free Software Foundation (1996). *Free software definition*.
<http://www.gnu.org/philosophy/free-sw.html>
- Free Software Foundation. *Licencias libres*. <http://www.gnu.org/licenses/license-list.html>
- Free Software Foundation (1994). *The GNU Project*. Recuperado el 23 Setiembre 2005 <http://www.fsf.org>
- Free Software Foundation. *Free Software*. Recuperado el 18 Setiembre 2005
<http://www.gnu.org/philosophy/philosophy.es.html#AboutFreeSoftware>
- Free Software Foundation (1998). *What is CopyLeft*. Recuperado el 12 octubre 2005 <http://www.gnu.org/copyleft/copyleft.html>
- García, Juan Tomas; Romero, Alfredo. *La pastilla roja: Software libre y revolución digital*, EditLin.
- González Barahona Jesús M. (2000). *Software libre, monopolios y otras yerbas*.
Todo Linux (núm. 3). <http://sinetgy.org/~jgb/articulos/soft-libre-monopolios>
- González Barahona Jesús M. (2002). *¿Qué se hace con mi dinero?* Todo Linux
(vol. 17). <http://sinetgy.org/~jgb/articulos/sobre-administracion>
- González Barahona Jesús M.; Robles, Gregorio; Seone Joaquín. (2003).
Introducción al software libre, Barcelona: Editorial Dykinson.
- Libro Blanco. *Software Libre*. Recuperado el 24 de octubre 2005
<http://www.libroblanco.com/html/index.php>
- Massachusetts Government. *Informal comments on Open Formats*. Recuperado el 23 octubre 2005 http://www.mass.gov/eoaf/open_formats_comments.html
- OMPI 1996. *Tratado de la OMPI sobre derechos de autor*. Recuperado el 29

octubre 2005.

ONU 1998. *Declaración Universal de los derechos humanos*. Recuperado el 15 noviembre 2005. <http://www.un.org/spanish/aboutun/hrights.htm>

Open Source Initiative (1998). *Open Source*. Recuperado el 03 noviembre 2005 <http://www.opensource.org>

Raymond, Eric. (2001). *The Cathedral & the Bazaar*. New York: Editorial O'Reilly.

Ross Anderson. *Security in Open versus Closed Systems*. Recuperado el 01 noviembre 2005 <http://www.ftl.cl.cam.ac.uk/ftp/users/rja14/toulouse.pdf>

Universidad Nacional de Colombia. Ofimática. Recuperado el 18 de octubre 2005 http://www.virtual.unal.edu.co/cursos/economicas/92211/lecciones/unidad3/of_definicion.htm

W3C (2002). *Intellectual Rights Notice and Legal Disclaimers*. Recuperado el 18 octubre 2005 <http://www.w3.org/Consortium/Legal/2002/ipr-notice-20021231>

Weinberg, Gerald M. (1998). *The Psychology of Computer Programming*. New York: Editorial Dorset House

Williams, Sam (2002), *Free as in Freedom*. O'Reilly.

Wikipedia. *Software Libre*. Recuperado el 20 setiembre 2005 http://es.wikipedia.org/wiki/Software_Libre

Wikipedia. *Licencias de software*. Recuperado el 27 octubre 2005 http://es.wikipedia.org/wiki/Licencias_Software