

Artículos docentes

Software libre: una alternativa para las bibliotecas

[Lic. María Laura Porcel Iturralde1 y Lic. Mabel Rodríguez Mederos2](#)

RESUMEN

Con el objetivo de explicar los fundamentos y elementos que caracterizan al movimiento del software libre, se realiza una breve historia del origen y evolución de este movimiento: el proyecto GNU, la aparición de Linux, los programas de código fuente abierto, el copyleft y las licencias para esta clase de software. Finalmente, se presentan algunos de los principales software libres orientados a la esfera bibliotecaria.

Palabras clave: Software libre, software de código fuente abierto, software bibliotecarios.

ABSTRACT

In order to explain the foundations and elements that characterize the free software movement, a brief history of the origin and evolution of this movement is made: the GNU project, the appearance of Linux, the programs of open source code, the copyleft and the licenses for this kind of software. Finally, some of the main free software oriented to the library sphere are presented,.

Key words: Free software, open source code software, library software.

Copyright: © ECIMED. Contribución de acceso abierto, distribuida bajo los términos de la Licencia Creative Commons Reconocimiento-No Comercial-Compartir Igual 2.0, que permite consultar, reproducir, distribuir, comunicar públicamente y utilizar los resultados del trabajo en la práctica, así como todos sus derivados, sin propósitos comerciales y con licencia idéntica, siempre que se cite adecuadamente el autor o los autores y su fuente original.

Cita (Vancouver): Porcel Iturralde ML, Rodríguez Mederos M. Software libre: una alternativa para las bibliotecas. Acimed 2005;13(6). Disponible en: http://bvs.sld.cu/revistas/aci/vol13_6_05/aci09605.htm

Consultado: día/mes/año.

La introducción y uso apropiado, tanto de las computadoras como de diversos productos informáticos en distintas esferas profesionales, facilitaron la gestión de la información, o al menos, de ciertas facetas de ella, debido a su alta capacidad de procesamiento, velocidad, flexibilidad y precisión.

En los años 80, una tendencia en la automatización de las bibliotecas llevó a la aparición de los llamados sistemas integrados de gestión de bibliotecas (SIGB) y los catálogos en línea de acceso público (OPAC). Su implantación benefició la integración y la colaboración de las diferentes áreas en el interior de la organización.

Más tarde con el surgimiento de Internet y el acceso, cada vez más común, por parte de las personas a las computadoras, condicionó que estos productos experimentaran adaptaciones para buscar trascender las fronteras institucionales, facilitar la comunicación y el servicio a los usuarios.

El problema realmente se erigió en las dificultades para acceder a esta clase de software que ofrecía el mercado. Para muchas instituciones, los costos de adquisición, aplicación y mantenimiento eran muy elevados con relación a sus presupuestos. En muchos casos, el mero pago de licencias o costos de arrendamiento imposibilitan su adquisición o mantenimiento. La problemática para las bibliotecas se agudizaría aun más, porque durante años han experimentado continuos recortes y afectaciones en materia de subsidios.

Hoy esta problemática sigue vigente. Sin embargo, hoy también, es la propia comunidad de creadores y usuarios la que crea mecanismos de igualdad y equidad para aquellos que no alcanzan el tren tecnológico. Así nació la llamada comunidad del software libre con nuevas propuestas para transformar la industria del software; sus esfuerzos están dirigidos a la creación, divulgación, uso, distribución y acceso al código fuente de los paquetes informáticos, así como a la divulgación de soluciones informáticas específicas.

Por tanto, puede decirse, que en estos momentos, a nivel mundial, se perciben dos grandes grupos de desarrollo de software: los de software libre y los de software propietario. La motivación esencial para desarrollar un programa o herramienta libre es que no exista ninguna herramienta para realizar una tarea determinada o que exista pero sea propietaria. El deseo de ser libre motiva a los programadores y otros profesionales a escribir sustitutos libres de programas privativos.

Esta motivación, convertida en principio, coincide con los fundamentos más altruistas enarbolados por los bibliotecarios durante siglos. La extensión del movimiento del software libre al ámbito bibliotecario ha posibilitado el desarrollo de aplicaciones particulares para esta esfera, como son los SIGB. La aparición de algunos SIGB de elevada calidad es una alternativa viable para muchas bibliotecas situadas ante un mercado informático de difícil acceso.

El propósito del presente trabajo es explicar los fundamentos y elementos que caracterizan al movimiento del software libre, así como identificar algunas herramientas disponibles, útiles a la esfera bibliotecaria.

EL SOFTWARE LIBRE, EL SOFTWARE DE CÓDIGO FUENTE ABIERTO

El hardware comprende la parte física de la computadora: placas electrónicas, cables, así como todos los periféricos que se conectan a ella; mientras que el software abarca los componentes lógicos o programas que posibilitan su funcionamiento.

Para desarrollar un software, el programador se vale de un lenguaje de programación que hace de intermediario entre el lenguaje humano y el lenguaje binario de la máquina. El compilado de instrucciones escritas en un lenguaje de programación, conforman lo que se denomina código fuente del programa.

Muchas empresas creadoras de tecnología computacional trabajan bajo condiciones de software cerrado con el objetivo de vender y distribuir el programa compilado sin temor a que alguien pueda modificarlo o reutilizarlo en otro programa, esta práctica se sustenta en los términos del llamado “licenciamiento” que habilita o no la utilización del programa. El concepto de venta de licencias no se ajusta a los programas libres.

Hace algunos años y de la mano del desarrollo de Internet, surgieron nuevos enfoques respecto al modo de hacer y de comprender estas acciones.

El software libre

Para lograr entender los orígenes del movimiento prosoftware libre, debemos situarnos en la primera mitad de los años 70, cuando las computadoras no resultaban de fácil acceso y la industria del licenciamiento de software aún no había desarrollado. Las aplicaciones de entonces, casi en su totalidad, se creaban a la medida; eran específicos para una determinada arquitectura, y cuando era necesario realizar alguna transformación que involucrase cambiar la arquitectura, los sistemas debían ser reescritos.

Esta situación generó un ambiente de cooperación entre los usuarios de las máquinas, que posibilitó intercambiar información y compartir los códigos fuentes. Los miembros de esta comunidad se hacían llamar hackers, ellos tenían la facilidad de estudiar, modificar y desarrollar los códigos de los programas, compartir el conocimiento y evitar la duplicación de esfuerzos.

Los ambientes propicios para este tipo de intercambio fueron las universidades, las instituciones de investigación y las empresas; en ellas, no percibía problema alguno en la distribución y el compartir códigos. Las ganancias de las empresas de informática de la época provenían de la venta de hardware, el software era considerado sólo parte integrante del equipo. Pero con la evolución de la industria informática, las computadoras se hicieron más potentes y asequibles.

Con la disociación entre software y hardware, a finales de los años 70 y principios de los 80, se generó un movimiento progresivo hacia el cerrado de códigos. Para los años 80, salvo raras excepciones, todo el software era cerrado o propietario. Las grandes compañías de software, al amparo de la legislación sobre la propiedad intelectual, firmaron acuerdos para desarrollar máquinas con su propio sistema operativo (cerrado o propietario). De esta manera, la comercialización de licencias comenzó a tomar cuerpo.

Estos acontecimientos causaron disgustos entre los hackers. Entre ellos, se destaca la figura de *Richard Stallman*, quien a propósito del tema, comentara: "una de las presunciones es que las compañías de software tienen un derecho natural incuestionable que los habilita para ser dueños de un software, y por lo tanto a disponer de poder sobre todos sus usuarios; si este fuera un derecho natural, entonces sin importar cuanto daño cause al público, no podríamos objetarlo. Sin embargo, y es muy interesante, la Constitución de los Estados Unidos de América y la tradición legal rechazan esta visión; el copyright no es un derecho natural sino un monopolio artificial impuesto por el gobierno que limita el natural derecho a copia de los usuarios." Y añadió, esta práctica "obstaculiza el desarrollo tecnológico y favorece el enriquecimiento acelerado de una minoría".¹ Por estas razones, *Stallman*, en 1984, abandonó el Laboratorio de Inteligencia Artificial del Massachusetts Institute of Technology (MIT) y comienza prácticamente desde cero a desarrollar un sistema operativo compatible con Unix, que denominó GNU.

Proyecto GNU

En 1985, con el progreso de GNU (GUN's Not Unix), surgió La Fundación para el Software Libre (Free Software Foundation-FSF). Paralelo al desarrollo de la fundación, se recibieron donaciones de máquinas con sistema operativo Unix y ello generó una problemática, porque el proyecto GNU prohibía el uso de software propietario, como es el caso de Unix; a su vez, sin donaciones no se podía continuar con el proyecto, así que se trabajó hasta terminar el desarrollo de uno libre. 1-2

El sistema que se pensó desarrollar tenía toda la intención de ser un clon de Unix, pero totalmente libre de código propietario. El programa seguía todas las especificaciones y parámetros de Unix, pero los programas fueron totalmente reescritos, sin utilizar cualquier código previamente disponible que pudiese ser objeto de disputa relativa a los derechos de autor. El hecho de que fuera compatible con este sistema era conveniente, porque las aplicaciones desarrolladas para Unix serían automáticamente compatibles con GNU, y los usuarios de Unix podrían fácilmente cambiar sus sistemas por GNU.

Aparición de Linux

En el año 1990, el sistema GNU poseía un grupo de herramientas pero su núcleo, Kernel, necesitaba trabajarse aún más. El núcleo es la parte central del sistema operativo, es el responsable de la ejecución de los programas, el escalamiento de tareas y el acceso a los recursos del hardware. No es hasta el año 1991, que el estudiante finlandés *Linus Torvalds*, desarrolla un núcleo compatible con UNIX al que denominó Linux. Y es finalmente en 1992, cuando se combina este núcleo con el sistema GNU, para lograr definitivamente un sistema operativo libre, que se denominó “GNU/Linux”. 1

Aunque es muy importante aclarar que el Linux no fue el primer software libre con éxito, porque, desde mucho antes, existía en forma estable, en muchos servidores UNIX, Apache como servidor web, Sendmail como servidor de correo y Bind como servidor DNS. Estas aplicaciones, sin duda alguna, formaban parte de los pilares en que se sostiene Internet hoy, pero fue Linux el que popularizó las ideas del movimiento Free Software y Open Source.

Con el desarrollo de productos libres, varios individuos, grupos de usuarios y compañías, recopilaron y compilaron dichas aplicaciones en formatos particulares; a estas compilaciones, se les llamó distribuciones Linux, así aparecieron Red Hat, Suse, Debian, Mandrake, entre otras. Algunas empresas comenzaron a ver al software libre como un competidor en el mercado, que posibilitaba otra alternativa para los clientes.

Esto generó que en el año 1998, las principales empresas de software comenzaran a soportar la plataforma Linux, entre ellas, sobresalen IBM, Oracle, HP, Compaq, Dell y Sun Microsystems. Sucedió además que, en este año, la empresa norteamericana IBM System desarrolló productos para la plataforma GNU. Las empresas informáticas desarrollaron nuevas formas de elaborar software, a partir del uso de programas libres pero que no se liberaban en forma libre y, en muchas ocasiones, competían con estos (*Rodríguez Mederos M.* Gestión de información y el software libre: herramientas para automatizar un centro de información. Tesis para optar por el título de Licenciada en Ciencias de la Información y Bibliotecología. La Habana: Facultad de Comunicación, 2003).

Programas de código fuente abierto

El movimiento de los programas de código fuente abierto nació a finales del año 1997 e inicios de 1998, cuando parte de la comunidad que trabajaba con software libre, decidió abandonar el término “software libre” y lo sustituyó por “software de código abierto” u “open source software”. La motivación para la adopción de este término se dio por la necesidad de evitar la confusión entre “libre” y “gratis” porque, en inglés, “free” presenta las dos acepciones. Bajo estas circunstancias, este grupo de creadores fundó la Open Source Initiative (OSI).

Para este movimiento, el software no debe ser abierto por cuestiones simples de libertad, sino porque constituye un modelo de desarrollo más eficiente, tanto técnica como económicamente, los costes de desarrollo se dividen entre los integrantes de la comunidad. La idea es que, cuando el código de un programa que se puede leer, modificar y distribuir, éste tiene mayores probabilidades de que se realicen mejoras, adaptaciones y correcciones, y, por tanto, se desarrolle.

Las motivaciones de la OSI se centraban puramente en cuestiones prácticas y no éticas a diferencia del Free Software; algunos de sus miembros procuraron atraer a los ejecutivos y a los usuarios de empresas. El movimiento pro código abierto, procuró eliminar las características filosóficas del movimiento y enfocarse a sus características utilitarias, la eficiencia del modelo de desarrollo y el desarrollo de productos con fines de uso corporativo. Actualmente, la OSI lleva adelante un programa de certificación de licencias en el que las compañías de software certifican sus licencias, las compatibilizan con los requerimientos de la OSI y adquieren el derecho de decir que sus productos son Open Source, entre estas compañías, se encuentran: IBM, Sun Microsystem, Apple Computer y Novel.

Aunque los seguidores de la OSI han intentado registrar la marca Open Source, no han tenido éxito hasta el momento.

Diferencias entre software libre y software de código fuente abierto.

En cuestiones de base, ambos movimientos coinciden en sus principios: el software debe ser abierto, libre para revisarlo, estudiarlo e implementarlo, pero difieren en la manera de implementar esos principios.

Tanto el OSS como el FS, se manejarán bajo una serie de preceptos para el desarrollo del software, que establecen una serie de diferencias en la manera de implementar dichos principios.

Para la OSS, el software considerado como software de código abierto cumple los siguientes criterios: 3

1. Redistribución libre.

La licencia no deberá impedir la venta o el suministro del software como un componente de una distribución de software agregado que contenga programas con códigos fuente distintos. La licencia no deberá requerir el pago de los derechos de autor u otra tasa por dicha venta.

2. Código fuente

El programa debe incluir el código fuente y permitir la distribución, tanto en código fuente como en forma compilada. Si alguna forma de un producto no se distribuye con el código fuente, debe existir un

medio público adecuado para obtener el código fuente por un costo razonable, sólo de reproducción preferentemente, o por medio de una descarga desde Internet sin cargo. El código fuente es la forma preferida a partir de la cuál un programador modifica el programa. El código fuente deliberadamente ofuscado no está permitido. No se permiten formas intermedias como la salida de un procesador o traductor.

3. Trabajos derivados

La licencia debe permitir modificaciones y trabajos derivados, así como que ellos se distribuyan bajo los mismos términos de la licencia del software original.

4. Integridad del código fuente del autor

La licencia puede impedir que el código fuente se distribuya en forma modificada. Esto puede realizarse sólo si la licencia permite la distribución de "archivos parches" junto al código fuente con el objetivo de poder modificar el programa en el tiempo de construcción. La licencia debe permitir explícitamente la distribución del software construido, a partir del código fuente modificado. La licencia puede requerir que los trabajos derivados tengan un nombre distinto o un número de versión distinto al del software original”.

5. No a la discriminación de personas o grupos

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

6. No a la discriminación de campos laborales

La licencia no debe restringir a nadie el uso del programa en un campo laboral específico. Por ejemplo, no puede impedir que el programa se emplee en un negocio, o que se utilice en una investigación genética.

7. Distribución de la licencia

Los derechos adjuntos al programa deben aplicarse a todos aquellos que reciben el programa sin necesidad de ejecutar una licencia adicional para estas partes.

8. La licencia no tiene que ser específica de un producto

Los derechos adjuntos al programa no deben depender de que el programa forme parte de una distribución particular de software. Si el programa se extrae de esa distribución y se utiliza o distribuye según los términos de la licencia del programa, todas las partes a las que el programa se redistribuya deben tener los mismos derechos que se garantizan en conjunto con la distribución original del software.

9. La licencia no debe restringir a otros softwares

La licencia no debe presentar restricciones a otros softwares que se distribuyan junto con el software licenciado. Por ejemplo, la licencia no debe insistir en que todos los otros programas distribuidos en el mismo medio deban suministrar su código en forma abierta.

A diferencia del anterior, el movimiento del Free Software, se rige por una serie de libertades: 4

- Ejecutar. La libertad de usar el programa, con cualquier propósito -nótese que existen muchas formas de utilizar un programa y ejecutarlo es una de ellas.
- Inspeccionar. La libertad de estudiar como funciona el programa y adaptarlo a sus necesidades. El acceso al código fuente es una condición previa para esto.
- Redistribuir. La libertad de distribuir copias, con lo que se puede ayudar a otros.
- Modificar y redistribuir la modificación. La libertad de mejorar el programa y hacer públicas las mejoras de modo que toda la comunidad se beneficie. El acceso al código fuente es un requisito previo para esto.

Si se puede conocer, utilizar, crear, programar, desarrollar, comunicar y compartir sin restricciones un software, entonces es software libre.

Si usted modifica un programa no tiene obligatoriamente que publicar dicha modificación, pero no tiene derecho a establecer restricciones para el software.

Libre no significa no comercial

El software comercial es el software que desarrolla una entidad con la intención de obtener utilidades, a partir de su uso. ``Comercial" y ``propietario" ¡no son la misma cosa! La mayoría del software comercial es propietario, pero existen software libres comerciales y software no libres no comerciales. 1

El software propietario es software que no es libre ni semilibre. Su uso, redistribución o modificación está prohibida, o se requiere que usted solicite una autorización previa que a menudo es tan restringida que no puede hacerse nunca libre de un modo efectivo.

Copyleft

Según las regulaciones legislativas de algunos países los derechos de autor pueden dividirse en patrimoniales (copyright) y morales. Los derechos patrimoniales pueden involucrar cuestiones de licenciamiento a terceros, comercialización, alquiler, transferencia de propiedad, acceso al código fuente y establecimiento de condiciones para su uso y alteración. Estos derechos son, por tanto, negociables. Los derechos morales en cambio comprenden los derechos del autor a que su nombre aparezca siempre vinculado a su obra con independencia de a quien pertenezca o por quien se utilice, son inalienables e irrenunciables.

Todos los softwares en la legislación de los Estados Unidos, como sucede en muchos otros países, protegen su “propiedad intelectual” con un copyright que se refleja en la licencia con que ellos se venden u ofertan. En el caso del software libre, para garantizar su libertad y lograr que los programas sean libres o abiertos, se utilizó la propia estructura legal empleada para la protección de los derechos patrimoniales que posee el autor. Basado en la ley del copyright, se utilizó el término “copyleft”, que en lugar de privatizarlo lo hizo de uso libre; así se garantizaron las libertades a los usuarios y se impidió que el software abierto se convirtiera en propietario. 1 Cada programa debe registrarse convenientemente bajo este término. Por medio de estas leyes, el autor toma los derechos sobre su obra, protege su autoría y sus derechos patrimoniales e impone condiciones para su licenciamiento con el propósito de hacerlo público en lugar de restringir su acceso.

“La idea fundamental del copyleft es que se autoriza la ejecución del programa, su copia, modificación y distribución de las versiones modificadas, siempre que no se añada ninguna clase de restricción a posteriori”. 1,5 Esta regla no entra en conflicto con las libertades centrales, sino que las protege. Con el trabajo de la FSF, se conceptúan varias ideas y temas, como es el caso del copyleft, el free software, el software comercial y el software cerrado. Existen licencias para software libres aun sin copyleft, aquellos que no poseen su correspondiente licencia pueden verse amenazados por las empresas, que pueden hacer desaparecer su libertad.

Algunos especialistas argumentan que un software es verdaderamente libre si no posee ni copyleft ni copyright, es decir, si es de dominio público, pero esto, debido a las razones expuestas es una posición insostenible e a la larga. Las personas que desean hacer software propietarios no pueden emplear códigos protegidos por copyleft, cuando alguien viola estos términos, suprime o añade nuevas condiciones que limiten en algo la disponibilidad de un programa con copyleft pierde el derecho de servirse del software. Otra restricción importante que fue adicionada al término del copyleft, radica en que toda obra derivada de la obra original tiene necesariamente que favorecer su licenciamiento exactamente con los mismos derechos.

El legado de la FSF fue precisamente el desarrollo del concepto de copyleft en el mundo del software. Su representación en una licencia, dio lugar a la Licencia GPL (General Public License GNU/GPL) .

La GPL perpetúa las libertades que se ofrecen, garantiza que todas las versiones subsiguientes y derivaciones del software conserven esas libertades, con independencia de quien las realice, el autor original o algún otro programador.

La GPL es una manera de plasmar jurídicamente el concepto de copyleft, ella contiene los términos de distribución específicos que se utilizan por excelencia en el proyecto GNU, por lo que se ha convertido en la base del FS (*Rodríguez Mederos M . Gestión de información y el software libre: herramientas para automatizar un centro de información. Tesis para optar por el título de Licenciada en Ciencias de la Información y Bibliotecología. La Habana: Facultad de Comunicación, 2003*).

Licencias y tipos de softwares

Licencias y modos de uso

Como se ha dicho anteriormente, el hecho de añadir una licencia permite asegurar que el uso que se haga del código y de sus derivados esté en correspondencia con los principios con los cuales se publicó el código. La licencia no es nada más que un texto que especifica cómo podrá utilizarse el código o el programa. Si no se aceptan las condiciones que impone la licencia, no puede utilizarse o modificarse el código o el programa.

Existen múltiples licencias, cada una con algunas diferencias con respecto a las otras; la más pura, según la filosofía del software libre es la GNU/GPL; a partir de ella, se añaden o eliminan restricciones.

Se utilizará indistintamente el calificativo “copyleft” o “vívica” para referirnos a las licencias que obligan a extender la licencia a todos sus derivados y compuestos.

A continuación, se citarán algunas de las más conocidas, así como algunas de características distintivas. Asimismo, se indicarán las aprobadas por la OSI, es decir, aquellas que siguen la filosofía del código abierto: 6-7

- GNU GPL: Filosofía del software libre, asegura las cuatro libertades y por eso es muy vívica. La idea es que la persona que recibe un programa tiene los mismos derechos que tenía el distribuidor cuando él obtuvo el programa, es decir, que código que toca, código que tiene que ser GNU GPL. (Aprobada por la OSI).

- GNU LGPL: (Lesser General Public License) Sigue la filosofía GNU GPL, pero no es tan vívica, porque está pensada para bibliotecas de funciones de software. La idea que subyace tras esta licencia es facilitar el acceso al código libre al gran público, para hacerlo, permiten que ciertas bibliotecas de software puedan utilizarse desde programas no libres, es menos restrictiva que la GPL. Antes se denominaba GNU Library General Public License; más tarde cambió su nombre por el actual. (Aprobada por la OSI).

- Licencia X11 y Xfree86: Filosofía GNU GPL pero no es vívica. Permite hacer modificaciones en el software y comercializarlo , pero sin publicar los cambios.

- BSD modificada. Versión modificada de la BSD (Berkeley System Distribution) con el fin de hacerla compatible con la GPL, es muy similar a X11, es decir, una licencia no vírica para software libre. (Aprobada por la OSI). 8

- Artistic License: Utilizada por Perl. parecida a GPL, pero no es vírica. Tiene una redacción bastante laxa, y por tanto, los defensores de GNU no la aconsejan y prefieren que si alguien desea una licencia no vírica utilice Xfree86 (o similares). (Aprobada por la OSI).

A partir de estas, todas las licencias siguen parte de la filosofía del software libre, no son compatibles con la GNU GPL, y por tanto, se enumeran separadamente:6

- BSD. Más permisiva y no vírica (Aprobada por la OSI).

- Apache: Más permisiva y no vírica.

- IBM Public License Versión 1.0: Muy parecida a la GNU/GPL, pero tiene diferencias en el tratamiento de las patentes. (Aprobada por la OSI).

- Mozilla Public License (MPL). Es muy parecida a la X11, es decir, no es vírica, pero no permite enlazar códigos bajo la licencia GNU/GPL. (Aprobada por la OSI).

- Netscape Public License (NPL). Casi idéntica a la Mozilla, pero con el inconveniente añadido que permite a Netscape utilizar un código creado bajo esta licencia para las versiones propietarias del programa.

- QPL. No vírica y obliga a que las modificaciones se distribuyan como correcciones añadidas al original. (Aprobada por la OSI).

- Sum Microsystem .

Patentes de software

El impacto de las patentes sobre el software libre es muy negativo. Más aún que para otros tipos de software (como el propietario). Existen tres características del software libre que explican este efecto negativo específico: 1

- Disponibilidad del código fuente.

El código fuente siempre está disponible para su estudio y escrutinio. Eso significa que todas las

tecnologías de software que se utilizan, están expuestas completamente a un análisis de patentes. Si una empresa tiene que considerar la posibilidad de luchar en un juicio por infracción de patente, la exposición del código fuente no es la mejor estrategia posible. Las empresas dificultan las querellas por infracción de patentes por medio de la protección del público del código fuente de sus aplicaciones.

- Imposibilidad de negociar licencias.

El software libre puede copiarse y redistribuirse sin restricciones. Puede ser modificado e incorporado en otros productos libres. Por tanto, no hay ningún punto único de distribución. Esto hace que sea realmente difícil encontrar un esquema para negociar licencias para el uso de patentes en programas libres, y es poco probable que se concedan licencias de muchas patentes para el uso en programas libres.

- El impacto en pequeños desarrolladores.

El software libre se desarrolla en muchos casos por empresas muy pequeñas y creadores individuales, con mucha frecuencia en su tiempo libre. Se reciben contribuciones de mucha gente de alrededor del mundo. El trabajo de examinar todo el código producido, y todas las contribuciones recibidas para buscar posibles usos de tecnologías patentadas está fuera de las posibilidades de estos individuos. 5

Tipos de software

Existen diferentes categorías de software: 1,4

- Software libre (Free Software).

El software libre es un software que posee una autorización para que cualquiera pueda usarlo, copiarlo y distribuirlo, sea en forma literal o con modificaciones, gratis o mediante una gratificación. En particular, esto significa que el código fuente debe estar disponible.

- Software de código fuente abierto (Open Source).

El término software de “código fuente abierto” se emplea por algunas personas para dar a entender que es software libre.

- Software de dominio público.

El software de dominio público es aquel software que no está protegido con copyright. Dominio público, es un término legal que quiere decir precisamente “sin copyright.

- Software con copyleft.

El software protegido con copyleft es un software libre cuyos términos de distribución no permiten a los redistribuidores agregar ninguna restricción adicional cuando éstos redistribuyen o modifican el software. Esto significa que cada copia del software, aun si se ha modificado, debe ser software libre.

- Software libre no protegido con copyleft

El software libre no protegido con copyleft viene desde el autor con autorización para redistribuir y modificar, así como para añadirle restricciones adicionales.

Si un programa es libre pero no está protegido con su copyleft, entonces algunas copias o versiones modificadas pueden no ser libres completamente. Una compañía de software puede compilar el programa, con o sin modificaciones, y distribuir el archivo ejecutable como un producto propietario de software.

- Software cubierto por la GPL

La GNU GPL (Licencia Pública General), es un conjunto específico de términos de distribución para proteger con copyleft a un programa. El Proyecto GNU la utiliza como los términos de distribución para la mayoría del software GNU.

- El sistema GNU

El sistema GNU es un sistema operativo libre completo similar a Unix. Debido a que el propósito de GNU es ser libre, cada componente individual en el sistema GNU tiene que ser software libre. No todos tienen que estar protegidos con copyleft, sin embargo; cualquier tipo de software libre es legalmente apto de incluirse si ayuda a alcanzar metas técnicas.

- Software GNU

Software GNU es software que se libera bajo el auspicio del Proyecto GNU. La mayoría del software GNU está protegido con copyleft, pero no todos; sin embargo, todo el software GNU debe ser software libre.

- Software semilibre

El software semilibre es software que no es libre, pero viene con autorización para particulares de uso, copia, distribución y modificación -incluye la distribución de versiones modificadas) sin fines de lucro. Pero también incluye otras restricciones.

- Software propietario

El software propietario es software que no es libre ni semilibre. Su uso, redistribución o modificación está prohibida, o requiere que usted solicite autorización que es tan restringida que no pueda hacerse libre de un modo efectivo.

- Freeware

El término "freeware" no tiene una definición clara aceptada, pero se utiliza frecuentemente para paquetes que permiten la redistribución pero no la modificación, y su código fuente no está disponible. Estos paquetes no son software libre.

- Shareware

El shareware es software que viene con autorización para redistribuir copias, pero establece que quien continúe el uso de una copia deberá pagar un cargo por licencia. El shareware no es software libre, ni siquiera semilibre. Existen dos razones por las que no lo es:

1. Para la mayoría del shareware, el código fuente no está disponible; de esta manera, no puede modificarse el programa en absoluto.

2. El shareware no viene con autorización para hacer una copia e instalarlo sin pagar una cantidad por la licencia, ni aún para particulares involucrados en actividades sin ánimo de lucro. En la práctica, la gente a menudo hace caso omiso a los términos de distribución y lo hace de todas formas, pero los términos no lo permiten.

- Software comercial

El software comercial es software que se desarrolla por una entidad que tiene la intención de obtener utilidades con el uso del software. Como se dijo, "comercial" y "propietario" ¡no son la misma cosa! La mayoría del software comercial es propietario, pero existe software libre comercial y software no libre no comercial".

La aplicación de uno u otro tipo de software estará sujeta a las particularidades del lugar donde se aplique y en dependencia del o los procesos que se deseen automatizar. Debido a las diferentes bondades que posee este tipo de software, es posible ofrecer servicios e implantación a la medida, así como escoger el tipo de herramienta más acorde para la solución del problema. Esta elección sólo es posible luego de haber realizado los pasos para definir la estrategia de implantación.

Algunas consideraciones sobre el tema

El software libre constituye una alternativa a las soluciones propietarias y un paso superior a una sociedad con un flujo libre de la información, para la mayoría de los ámbitos, tanto públicos como privados. Este conjunto de soluciones informáticas que se liberan bajo distintas licencias, facilitan la reutilización de la experiencia –como sucede con el conocimiento científico-, permiten su uso generalizado -en algunos casos gratuito-, y ello no solo es recomendable por el hecho económico de favorecer la competencia en cada sector, reducir los costos y delimitar la dependencia tecnológica, sino también porque proporciona una mayor seguridad y fiabilidad en términos de seguridad informática así como una mayor capacidad de evolución que sus contrapartidas comerciales. Y todo esto, sin olvidar uno de los propósitos más importantes que es: contribuir al desarrollo tecnológico de un país bajo el principio de compartir conocimientos. En este caso, muchas organizaciones de información se ven especialmente beneficiadas porque una gran cantidad de estas herramientas permiten la creación de infraestructuras adecuadas para la institución. Se entiende por infraestructura de información a: la capacidad de una organización o de un país para hacer accesible el conocimiento y la información, para asegurar su transferencia y en consecuencia, lograr que el conocimiento cumpla su función social. 9

Como puede observarse, son muchas las motivaciones que pueden impulsar el diseño de proyectos de gestión para la utilización de herramientas libres en una organización, aunque muchas de ellas están motivadas hacia estos sistemas por la filosofía de trabajo en la que se desarrollan y que los lleva a (*Rodríguez Mederos M . Gestión de información y el software libre: herramientas para automatizar un centro de información. Tesis para optar por el título de Licenciada en Ciencias de la Información y Bibliotecología. La Habana: Facultad de Comunicación, 2003*) :

- Escoger el software gratuito que más se adecue a sus propósitos.
- Hacerle modificaciones sin tener que solicitar un permiso, ni que el hecho de hacerlo sea un delito.
- El mismo es un producto que en su mayoría se ha auditado, revisado y perfeccionado por la comunidad que lo usa.

El software libre es generado por expertos programadores voluntarios, empresas y otros tipos de organizaciones que ofrecen sus soluciones desarrolladas al resto de la comunidad para que las utilice de forma libre.

HERRAMIENTAS LIBRES

Avanti 2

Avanti se inicia en octubre de 1998 por iniciativa de *Peter Schlump* , especialista en sistemas de información del Sistema de Bibliotecas Suburbanas Norte en Wheeling, Illinois.

Es un sistema escalable que se adapta a una biblioteca de cualquier tamaño. Avanti se enfoca a una implementación neutral e independiente del sistema operativo. Para ello, utiliza el lenguaje de programación Java que permite usar el mismo programa en diferentes sistemas operativos.

El diseño del programa se orienta en un modelo general de biblioteca, no en una biblioteca en particular. Esto es un problema recurrente en sistemas comerciales que generalmente nacen en una biblioteca determinada y después se venden a otras bibliotecas que pueden tener flujos de trabajos muy diferentes. Los módulos actuales incluyen circulación, catalogación y catálogo público (OPAC). Avanti usa una base de datos Java de desarrollo propio: PicoDB.

OSDLS: Open Source Digital Library System / Phyteas 2

Se inicia a comienzos de 1999 con bibliotecarios e informáticos de las universidades de Arizona, Yale y de Windsor y con el objetivo de crear un sistema de gestión bibliotecaria de próxima generación. A diferencia de los sistemas comerciales actuales que apuntan a bibliotecas de determinado tamaño, este sistema intenta ser independiente del tamaño de la biblioteca.

Actualmente corre bajo Linux. Integran tecnologías como XML, RDF, MySQL, Javascript, Java, RMI, LiveConnect y otras en su arquitectura.

Los módulos disponibles actualmente incluyen a: Phyteas, un editor Marc y OPAC. Actualmente, se buscan interesados para integrar grupos de trabajo para los módulos de adquisición, circulación y publicaciones seriadas.

OpenBook 2

OpenBook es un sistema de gestión bibliotecaria desarrollado para pequeñas bibliotecas escolares y públicas por la Technology Resource Foundation. Se basa en un sistema existente: Koha, que fue creado en Nueva Zelanda para una biblioteca pública. Incluye 3 módulos: OPAC, módulo de catalogación y módulo de circulación. Para el futuro se espera implementar un módulo de adquisición.

Comprende pantallas (interfaces) basadas en web. Esto significa que se utiliza mediante el navegador estándar del sistema. Es multilingüe: inglés, francés, español. Abarca el control de autoridades y se desarrolla un cliente Z39.50 para integrarlo con su módulo de catalogación con vistas a importar registros de otras bases de datos.

Se perfeccionó para Linux pero funciona también con Windows NT. A diferencia de Koha, es

compatible con Marc21. Utiliza tecnologías como: PHP, Perl, MySQL y servidor Apache.

OpenIsis 2,10

Existe el viejo MicroIsis para máquinas viejas que no funcionan con Windows. Ahora se utiliza WinISIS y existen una serie de utilitarios y herramientas para los creadores, programadas por el equipo de Bireme en Brasil.

Sin embargo, después de una revisión exhaustiva del software existente se llegó a la conclusión que el software actual para bases de datos Isis no cubre todo el espectro de aplicaciones requeridas, por ejemplo, las herramientas de desarrollo disponibles (utilitarios y librerías de Bireme) son propietarias y cerradas, únicas opciones en el caso de utilizar Linux como sistema operativo. Es por esto sumamente importante contar con una herramienta completamente abierta y libre que permita trabajar e interactuar con las bases de datos ISIS.

Por esta razón, se creó el proyecto OpenISIS, cuyo objetivo fundamental es dotar a la comunidad ISIS mundial de la biblioteca de funciones necesarias para desarrollar las herramientas finales de trabajo, producto del esfuerzo de los especialistas que laboran en el proyecto nació la primera generación: OpenISIS. Actualmente, existe una segunda generación: MALETE, que es un servidor de base de datos ISIS con varias características que lo hacen superior a las herramientas ISIS tradicionales; entre estas características, se destaca el aumento de la longitud de los índices (512 Kb) y el tamaño de la base de datos hasta varios giga bytes, así como contar con una interfase de búsqueda similar a la interfase Z39.50.

OpenISIS facilita la programación en varios lenguajes modernos como son el PHP, Java, PERL porque cuenta con interfases, módulos y funciones para ellos de manera que un programador puede de forma simple diseñar interfases web para las bases de datos ISIS. Hasta el momento, se conoce la existencia de dos proyectos públicos y libres que utilizan OpenISIS, Clabel e Igloo, ambos pueden encontrarse y descargarse en Internet desde servidores públicos, esto no significa que otras instituciones no posean sus propias aplicaciones basadas en OpenIsis y que las usen pero sin distribuirlas.

Koha 11

Koha se desarrolló en Nueva Zelanda en el año 1999 por Horowhenua Library Trust y Katipo Communications Ltd. Esta herramienta se propone resolver las necesidades de una pequeña biblioteca. Originariamente, se presentó como de código fuente abierto por accidente, la firma que comenzó el proyecto no quiso asumir los costos futuros del desarrollo o manejar la ayuda técnica. Desde entonces, la biblioteca del ILS Horowhenua y la biblioteca pública de Nelsonville continuaron su uso porque podían

desarrollar y modificar los campos, añadir nuevos módulos según sus necesidades y las características y requisitos particulares de su institución. Así se demostró que un proyecto de fuerte abierta, de estas características, prosperará en la comunidad, si las bibliotecas destinan sus recursos humanos a su desarrollo sustentable.

Su modelo de desarrollo lo ha hecho más eficaz y ha atraído a muchos de los programadores de la comunidad a sumarse al proyecto. De hecho, algunos consideran que a raíz de esto último, es un producto más estable y lo bastante acabado para su puesta en práctica.

Varias bibliotecas alrededor del mundo lo experimentan, principalmente en escuelas o en bibliotecas especiales de organizaciones no lucrativas. Como muchos paquetes de SIGB de código fuente abierto, tiene el potencial de convertirse en un SIGB universal siempre que despierte un interés serio en el mercado de las bibliotecas, comenzando con las bibliotecas públicas. Actualmente, su ayuda multilingüe facilita que internacionalmente se difunda. La ayuda técnica se proporciona por medio de la propia comunidad que lo emplea aunque Katipo proporciona foros y listas de correo propias del proyecto.

El sistema funciona bajo la plataforma Linux o Windows, o en cualquier servidor web aunque preferiblemente Apache, MySQL, y los módulos del Perl (el Date::Manip, el DBI, el Set::Scalar, el DBD::mysql o el otro módulo de la base de datos del SQL, AuthenDBI para la seguridad opcional, CDK para el interfaz opcional del telnet). La base del código es altamente flexible y se puede migrar a cualquier número de plataformas, tal vez incluso a Apple Mac OS X o cualquier servidor de Unix comercial. Los módulos completamente puestos en ejecución incluyen OPAC, la catalogación -con ayuda de USMARC y MARC21-, la circulación, así como también el módulo de adquisición.

GNUTeca 12

Se inició por tres programadores brasileños en el año 2001 y se popularizado entre las bibliotecas públicas, académicas y gubernamentales en ese país. Es posible acceder a él mediante su sitio web. GNUTeca apunta al contexto de las "bibliotecas académicas y especiales pequeñas". Una de sus dificultades es que el software y su documentación no se encuentran traducidos a otros idiomas. A pesar de la barrera lingüística, la prueba eficaz de este paquete de software, fue satisfactoria.

Las buenas noticias son que el número de programadores ha crecido desde su presentación inicial, debido a que, es un proyecto académico subvencionado, y contiene un depósito de CVS para animar contribuciones de programadores exteriores. Sus presentaciones oficiales se hacen en intervalos periódicos, las modificaciones en CVS indican un desarrollo activo. A pesar de su juventud, ha alcanzado cobertura nacional con la puesta en práctica completa de los módulos OPAC, catalogación original con la ayuda de MARC21, circulación con la ayuda de reserves/booking y la conversión de CDS/ISIS. Además, se encuentran en desarrollo los módulos de adquisición, de documentos por entrega y los préstamos entre bibliotecas.

GNUTeca funciona sólo bajo Linux, Apache, PHP y PostgreSQL. También depende de una base de datos de cosecha propia llamada MIOLO, componente que es sólo nativo a la comunidad de código fuente abierto de Brasil. La ayuda opcional para los clientes GTK se proporciona con PHP-GTK.

PhpMyBibli 13

PhpMyBibli es la tentativa abierta más reciente en herramientas de SIGB de código fuente abierto, se originó en Francia en el año 2002. La programación de esta herramienta en su mayoría se realizó por un primer programador pero más reciente tuvo ayuda de otros programadores, que han ensamblado y aportado al proyecto la traducción al inglés y español de las interfases y los manuales para darle una apertura internacional. Esta clase de cooperación se ha facilitado con el uso de listas de correo y un depósito de CVS.

El proyecto todavía está en la etapa de prueba y sus presentaciones frecuentes carecen de una enumeración de la versión que se presenta en las distintas ediciones; no obstante, ha logrado una cantidad considerable de trabajo en áreas como el OPAC, la circulación y la catalogación con la ayuda de UNIMARC. La ayuda USMARC o MARC21 es el paso lógico siguiente porque sus creadores principales desean cautivar el mercado norteamericano de las bibliotecas. Funciona bajo Linux o Windows, con cualquier Web Server, PHP y MySQL.

Sus estándares se acercan a una base de datos SQL flexible y el modelo de red cliente/servidor. Una de las mejores características de este paquete es su interfaz amigable e intuitiva, de uso fácil especialmente en el módulo de catalogación.

OpenBiblio 14

OpenBiblio se ha desarrollado constantemente desde el año 2002 por ser el código fuente abierto de uso más fácil, más intuitivo entre los SIGB disponibles en el mercado libre. Como algunas de sus contrapartes de código fuente abierto, emplea plantillas del CSS y de PHP para asistir la gerencia del contenido por parte de los bibliotecarios más inexpertos a cargo de mantenimiento de la red. Aunque ninguna biblioteca grande ha expresado interés por su adopción, el número de creadores activos ha crecido y su funcionalidad ha mejorado proporcionalmente. El blanco de su audiencia se centra más bien en colecciones privadas o personales, clubes, iglesias, escuelas o bibliotecas públicas, tiene el potencial de ser universal. Desafortunadamente, el módulo de la circulación puede manejar sólo los códigos postales de los Estados Unidos; su internacionalización se realiza por medio de plantillas multilingües.

OpenBiblio funciona bajo Linux o Windows, cualquier Web Server, PHP, y MySQL. Se jacta

actualmente de un OPAC completamente desarrollado, el módulo de circulación y la catalogación con ayuda de USMARC.

phpMyLibrary 15-16

El proyecto phpMyLibrary se inició en el año 2001 con el esfuerzo de un sólo especialista en Filipinas como medio para proporcionar una solución barata y accesible a la automatización de las bibliotecas en ese país. Su progreso ha ganado energías gradualmente desde entonces, porque son muchas las bibliotecas públicas, académicas y especiales que pueden utilizarlo en este momento. Su audiencia mas fuerte se centra en las "bibliotecas académicas y especiales pequeñas". Se ofrece en inglés. El modelo de desarrollo se centraliza altamente, como Avanti y PYTHEAS (OSDLS), con el último control del código fuente en las manos del fundador del proyecto; no obstante, existe un depósito de CVS.

phpMyLibrary corre en Linux o Windows, PHP, MySQL, phpMyAdmin (para administrar el MySQL) y python (opcional). Sus módulos de catalogación (originales sólo), circulación y OPAC se complementan con la ayuda y la herramienta de importación de registros desde una base de datos CDS/ISIS.

REFERENCIAS BIBLIOGRÁFICAS

1. Stallman R. Software libre para una sociedad libre. 2004. Disponible en: <http://www.sindominio.net/biblioweb/pensamiento/softlibre/> [Consultado: 5 de febrero del 2005].
2. Lencinas V. Software bibliotecario abierto y gratuito. Disponible en: <http://www.procesos-tecnicos.com.ar/software/OpenSource.htm> [Consulta: 20 de enero del 2005].
3. Open Source Initiative. Disponible en: <http://www.opensource.org/docs/definition.php> [Consultado: 5 de febrero del 2005].
4. Free Software Foundation. La definición de Software Libre. Disponible en: <http://www.gnu.org/philosophy/free-sw.es.html> [Consultado: 20 de febrero del 2005].
5. Free Software Foundation. Qué es copyleft? Disponible en: <http://www.gnu.org/copyleft/copyleft.es.html> [Consultado: 12 de enero del 2005].
6. Free Software Foundation. Diversas licencias y comentarios sobre ellas. Disponible en: <http://www.gnu.org/philosophy/license-list.es.html> [Consultado: 15 de enero del 2005].

7. Free Software Foundation . GNU General Public License. Disponible en: <http://www.fsf.org/licenses/gpl.html> [Consulta: 15 de enero del 2005].
8. Free Software Foundation. El problema de la licencia BSD. Disponible en: <http://www.gnu.org/philosophy/bsd.es.html> [Consultado: 30 de abril del 2005].
9. Ponjuán Dante G. Gestión de información en las organizaciones: principios, conceptos y aplicaciones. Santiago de Chile: CECAPI, 1998. XXII, 222 p. (Serie: Gestión de Información).
10. OpenIsis. Disponible en: <http://www.openisis.org/openisis/doc/OpenIsis> [Consultado: 20 enero del 2005].
11. Koha. About Koha. Disponible en: <http://koha.org/about> [Consultado: 4 de diciembre del 2004].
12. GNUteca. Disponible en: <http://gnuteca.codigolivre.org.br/> [Consulta: 15 diciembre 2004].
13. Pservices. PMB Services: Des services P our M a B ibliothèque. Disponible en: <http://www.sigb.net/> [Consultado: 17 de enero del 2005].
14. OpenBiblio. Disponible en: <http://obiblio.sourceforge.net> [Consultado: 10 de enero del 2005].
15. PhpMyLibrary. Disponible en: <http://phpmylibrary.sourceforge.net> [Consultado: 17 de enero del 2005].
16. MyLibrary@NCState. About MyLibrary. Disponible en: <http://my.lib.ncsu.edu/?cmd=about&id=39> [Consultado: 10 de enero del 2005].

Recibido: 26 de octubre del 2005.

Aprobado: 15 de noviembre del 2005.

[1-Licenciada en Ciencias de la Información y Bibliotecología.](#)

[2- Máster en Bibliotecología y Ciencia de la Información. Centro de Investigaciones del Petróleo \(CEINPET\).](#)

MsC. Mabel Rodríguez Mederos. Centro de Investigaciones del Petróleo (CEINPET). Washington No. 169 esq. a Churrucá, Cerro. Ciudad de La Habana. Cuba. Correo electrónico: mabel_rdguez@yahoo.

[com](#)

Ficha de procesamiento

Clasificación: Artículo docente.

Términos sugeridos para la indización

Según DeCS 1

PROGRAMAS DE COMPUTACIÓN; BIBLIOTECAS.

SOFTWARE; LIBRARIES.

COMPUTER PROGRAMS; SOFTWARE LIBRARIES; LIBRARIES

Según DeCI 2

PROGRAMAS DE COMPUTADORA; BIBLIOTECAS.

SOFTWARE; LIBRARIES.

COMPUTER PROGRAMS; SOFTWARE LIBRARIES; LIBRARIES.

1 BIREME. Descriptores en Ciencias de la Salud (DeCS). Sao Paulo: BIREME, 2004.

Disponible en: <http://decs.bvs.br/E/homepagee.htm>

2 Díaz del Campo S. Propuesta de términos para la indización en Ciencias de la Información. Descriptores en Ciencias de la Información (DeCI). Disponible en: <http://cis.sld.cu/E/tesauro.pdf>