



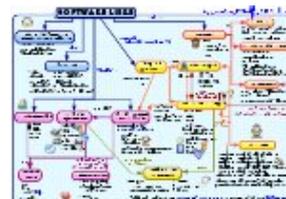
Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

Explicar el software libre y glosario (13625 lectures)

Per René Mérrou, [H \(http://h.says.it/\)](http://h.says.it/)

Creado el 04/03/2008 02:17 modificado el 04/03/2008 02:17

(1)(2)



Para explicar el software libre tengo en la es.gnu.org⁽³⁾ éste [mapa conceptual](#)⁽⁴⁾ que permite de un vistazo tener una buena impresión general [nota posterior: el mapa ha ido evolucionando, actualmente tengo [este material didáctico](#)⁽⁵⁾]. Para un profesor es una buena herramienta, ayuda a preparar unidades didácticas, como por ejemplo dejando ver fácilmente que tiene más importancia y calcular cuantas sesiones dedicar a cada parte.

Me han comentado que vendría bien que añadiese la información necesaria para que un profesor, sin tener mucha experiencia en el mundo del software libre, lo pueda explicar en un aula. Así, partiendo de la necesidad básica de explicar los acrónimos como la FSF (Free Software Foundation) mi idea es detallar todo lo que aparece para darles una buena respuesta a las posibles preguntas que pueden plantear los alumnos intentando hacer un esfuerzo de resumen que permita no dar más de un párrafo a cada referencia.

Ese es el reto que me pongo, vosotros diréis si lo logro. Cuento con vuestras correcciones e ideas para mejorarlo en comentarios. :)

[Traducción: [català](#)⁽⁶⁾]

Índice

[Introduciendo el software libre](#)⁽⁷⁾ . Cómo explico el mapa conceptual

[Realizaciones del software libre](#)⁽⁸⁾ . De qué hablamos y qué tenemos hoy en día.

[Libertades del software libre](#)⁽⁹⁾ . Define mejor el software libre saber que necesita para ser considerado libre.

[Valores del software libre](#)⁽¹⁰⁾ . Eso responderá a muchos el **por qué**.

[La comunidad del software libre](#)⁽¹¹⁾ . Quién participa.

Introduciendo el software libre

Lo primero que se puede apreciar en el mapa es que hay cuatro grandes zonas con colores diferenciados. La violeta que describiría como *realizaciones*, la amarilla que habla de *comunidad*, la azul sobre los *valores*, y finalmente en rojo suave la zona de las libertades o *condiciones del software libre*.



Así pues, los colores casi coinciden con las clásicas preguntas *qué*, *quién*, *por qué* y *cómo*. Veamos primero de qué estamos hablando.

Antes un vistazo rápido al mapa:

Si lo queréis usar o modificar, aquí tenéis el fichero fuente en el estándar [svg](#)⁽¹²⁾ que usa el inkscape: [Mapa Conceptual del Software Libre](#)⁽¹³⁾

Realizaciones del software libre

El software libre se distribuye mayoritariamente en DISTRIBUCIONES como Debian Gnu/Linux que es un proyecto en el que participan más de mil personas de todo el mundo que recogen programas que son libres y los adaptan e integran para permitir al usuario instalar, desinstalar o actualizar cómodamente sus programas en un entorno de calidad.

De Debian deriva toda la serie de Ubuntu como Ubuntu, Xubuntu, Kubuntu i Edubuntu que se diferencian en el entorno de escritorio o en detalles más específicos como es Edubuntu que está más encaminada a la educación. De Ubuntu a su vez otro equipo de gente ha derivado la distribución gNewSense con el objetivo de asegurar al 100% al usuario que sólo está usando software libre sin ningún driver o programa privativo.

BSD (Berkeley Software Distribution) es un tipo de licencia parecida a Unix como GNU/Linux pero que permite que cualquiera pueda cerrar un software libre. GPL (GNU General Public License) es la licencia de la FSF. Apache es otro tipo de licencia libre, precisamente el que usa el servidor de páginas web apache.

OpenBSD es una distribución basada en BSD que enfoca su esfuerzo en la seguridad. Debian GNU/Linux es una distribución que enfoca su esfuerzo en dar a los usuarios el máximo de aplicaciones libres. La respalda una comunidad de desarrolladores y usuario muy grande. OpenSolaris es la distribución de SUN, conocida por la estabilidad de su núcleo y vanguardia de algunas de sus aplicaciones. Mandriva Linux es la única distribución de pago que queda que esté en manos europeas.

Así pues las distribuciones llevan la selección, los empaquetados y la reunión de los contenidos instalables.

Inicialmente todos los programas libres tiene dos formatos a la hora de distribuirse: por un lado el PROGRAMA EJECUTABLE también llamado el binario y por otro lado el CÓDIGO FUENTE DEL PROGRAMA.

El PROGRAMA EJECUTABLE es el que el ordenador sabe poner en marcha. Está hecho en el lenguaje de la máquina y no es inteligible por los humanos porque es como mirar unos y ceros.

El CÓDIGO FUENTE DEL PROGRAMA es lo que escriben realmente los programadores, sus instrucciones como por ejemplo *coger un número, ponerlo en la memoria, sumarle otro, mostrarlo por pantalla...* Son instrucciones, normalmente usando palabras inglesas, que juntas son como la receta de lo que el ordenador debe hacer.

Junto con el programa ejecutable suelen ir unos contenidos libres como pueden ser imágenes, sonidos, plantillas (documentos ya preparados con el formato propuesto pero, sin el contenido), que no son realmente programas pero si forman parte de lo que se puede llamar software. Hay equipos muy variados de personas, (todos se consideran desarrolladores, aunque no programen), que contribullen añadiendo traducciones, fuentes tipograficas (los distintos tipos de letra). La internacionalización (I18n) es preparar un programa para que se pueda poner en diferentes idiomas y la L10n, localización, es darle una darle ya una traducción y arreglos necesarios para poder usarlo con un idioma en particular.

El programa ejecutable se obtiene en el proceso de compilado. Este proceso permite tener a partir de los programas que hacemos poco a poco un ejecutable y, puede cambiar de un tipo de máquina a otra; los programas se pueden compilar para un pentium o para otra cpu como una G4.

Algunos ejemplos de programas libres pueden ser: OpenOffice.org que es una de las aplicaciones de más éxito en el software libre. Implementa el OpenDocument, que es el único estándar internacional ISO reconocido para



documentación. Firefox es el navegador libre más usado. Kontact es la suite de gestión de información personal de KDE, Xine es un reproductor multimedia libre para entornos tipo Unix y windows, Emacs es un editor de texto con un número increíble de funcionalidades usado con mucha frecuencia por técnicos. GIMP (GNU Image Manipulation Program) es un conocido editor de imágenes.

man es la abreviatura de manual. Junto con los man se pueden encontrar Ayudas, guías de usuario o de desarrollador, Preguntas de uso frecuente (PUF o en inglés FAQ)

El COPYLEFT es una forma de dotar al software libre de una protección frente a quien quiera apropiárselo y cerrarlo. La dota de una condición "vídica" que obliga a mantener la libertad del software y que impide situaciones como que una empresa pueda apropiarse de todo el trabajo, cambiarlo de nombre y venderlo como un producto no libre. El símbolo del Copyleft es como el del Copyright, una C dentro de un círculo, pero girada hacia el otro lado.

Los **ESTÁNDARES ABIERTOS** son los han sido desarrollados en un proceso abierto a todos los interesados y que pueden ser leídos y usados por todo el mundo sin tener que pagar por ello. No tienen patentes (visibles u ocultas), se supone que deben tener especificaciones sencillas. Es natural pedir que además tengan una implementación libre ya desarrollada. Son los que permiten que la comunidad se centre en el desarrollo de aplicaciones y compita por los resultados no por quién es capaz de poner más trabas al otro para que sus usuarios se queden atados a su servicio.

La IETF (Internet Engineering Task Force) es una organización que trabaja por los estándares y la educación. Otra organización de estandarización es el W3C (World Wide Web Consortium) OASIS (Organization for the Advancement of Structured Information Standards) es un consorcio sin fines de lucro que trabaja por los estándares internacionales relacionados con el mundo empresarial. Y una organización más, la ISO (Organización Internacional para la Estandarización o International Organization for Standardization) tiene miembros en casi todos los países.

Estas organizaciones internacionales, y muchas otras, trabajan para que podamos usar estándares como el HTML (HyperText Markup Language) que describe el formato de documentos estructurados como las páginas web. IP especifica como deben transmitirse los paquetes de datos en una red y, especifica como implementar el flujo de datos a través de IP. También existen los RFCs (Request For Comments) son documentos que proponen especificaciones de una forma muy abierta y son por ejemplo la base de mucha de la tecnología relacionada con Internet.

Libertades del software libre

Para que pueda decirse que tiene esta LIBERTAD, tiene que proveernos al menos de unas determinadas libertades de forma explícita. Éstas estarán explicadas en su licencia que nos dirá si somos de libres en el uso, distribución, y modificación de nuestros programas.

La FSF (Free Software Foundation⁽¹⁴⁾) fue la pionera en definir cuales son las libertades necesarias para garantizar la libertad del software. Siguiendo ese criterio, mantiene una lista actualizada con todas las licencias que pueden ser consideradas libres y ha creado varias licencias libres (GPL, LGPL y otras) que garantizan la libertad del software en distintos escenarios.

Otros definen el software libre de otra forma, como la OSI (Open Source Initiative⁽¹⁵⁾). La definición de la OSI está relacionada con lo que Debian considera libre a la hora de aceptar nuevas aplicaciones. La diferencia se puede resumir en que detalla más las condiciones pero no tiene el requisito del copyleft. (ver explicación sobre el copyleft en el apartado sobre las realizaciones)

Las libertades de USO indican que la licencia no puede impedir que se utilice en cualquier ordenador, o el tiempo que se quiera, o para lo que se desee. Estas libertades suelen ser restringidas en el software privativo (definido por contraposición como el que no es libre, el que nos priva de estas libertades).

El poder realizar una MODIFICACIÓN en un programa es esencial para ir añadiendo mejoras y que progrese en comunidad. Así al software libre se le pueden añadir mejoras, adaptaciones a situaciones concretas, actualizaciones al paso del tiempo y cambio de las necesidades, y diversificaciones para poder satisfacer simultáneamente las diferentes necesidades (un buen ejemplo es añadir pluguins que permitan usar un software genérico de una forma específica).

Una vez desarrollado un programa se le da, o no, una DISTRIBUCIÓN, que es pasárselo a alguien para que lo pueda usar. Si ha de ser libre y es distribuido, debe permitir que se pueda reaprovechar copiándolo, modificándolo, y darle



una redistribución una vez modificado. Es sorprendentemente sencillo y barato, y ésta es una de las señas propias de la era digital. La misma forma de distribución puede convertirse en un valor añadido más, como pasa con los live-CDs, que son CDs con un sistema operativo y un amplio abanico de aplicaciones que funcionan directamente desde el CD sin necesidad ni de tener disco duro. "No solo puedes copiarlo a un amigo, sino que puedes añadir detalles o darle funcionalidades claves o cambiarle el idioma."

Naturalmente hacer todo ese trabajo no tiene que ser siempre sin remuneración. Se puede cobrar por una amplia variedad de aportaciones resultado de su trabajo desarrollando software libre, como es natural. Por ejemplo: si una entidad quiere una funcionalidad y no existe, no le queda más remedio que pagar a alguien para que la haga, eso es siempre así tanto en software libre como en software privativo. Es más, usar simplemente el software libre durante mucho tiempo y no aportar nunca nada a cambio, aunque sea en la medida de las propias posibilidades, puede ser mal visto.

Valores del software libre

El software libre es UNA FORMA ÉTICA DE ENTENDER EL SOFTWARE en su desarrollo, comercialización, distribución y uso. Un comportamiento responsable por parte del desarrollador o de la empresa que lo contrate no debe ceder a intereses monetarios hasta el punto de perder su ética profesional. Un profesional de la informática no debe añadir a un programa una limitación solamente para exigir un dinero a cambio de que la quiten, es un ejemplo de los muchos abusos que existen en el sector hoy en día. En medicina sería el equivalente a darle la medicina mal mezclada si no paga un extra por la "medicina professional edition"

Es una forma ética en su desarrollo porque da las funcionalidades que quiere el usuario evitando las que no quiere, como por ejemplo los popups que molestan y que los navegadores libres permiten filtrar. También se desarrolla pensando que otros van a querer verlo, aprender de él, modificarlo. No hay que olvidar que eso tiene como efecto secundario darle un poco más de calidad. Otro ejemplo, el DRM no es una funcionalidad deseada. "Es mi ordenador usted no tiene derecho a espiarme, ni a estropear nada, ni a limitar sus funcionalidades. Y menos sin que yo me entere. ¿Dónde está su ética profesional?"

Es ética en su comercialización porque no incluye las trampas a que estamos desafortunadamente habituados. Licencias interminables que impiden comportamientos naturales. Por ejemplo, pagamos por usar un software que por licencia no podemos poner en el siguiente ordenador que nos compremos. O poner dentro de un programa de contabilidad detectores de cambio de dispositivos para obligar a pagar al productor cada vez que se cambie algo en el ordenador, solamente para que pueda seguir funcionando. Funcionalidad que te quitan y te obligan a pagar por volver a tenerla, sin que nada haya cambiado ni estemos pagando por un trabajo necesario. Eso es un engaño, un abuso que por supuesto no existe en el software libre.

Es ética en su distribución por su transparencia, todo el mundo puede saber lo que se distribuye. Venderle a uno algo diferente de lo que uno piensa que está comprando es más difícil, porque se puede constatar y demostrar fácilmente. Distribuir un software libre significa publicarlo o dejar que se publique. Incluso la calidad de lo que se compra es transparente para un técnico que sepa leer el código.

Y ética en su uso porque es ético exigir que no te traten a uno con engaños y no le coarten las libertades. Es ético no dejar que le cierren el acceso al conocimiento. Es ético actuar por el bien de todos, colaborar para que se use lo que sea mejor para todos. En un colegio por ejemplo, es ético usar una suit de ofimática libre en vez de despreocuparse y copiar ilegalmente todo lo que se desee. No lo es cualquier cosa que signifique rechazar estándares internacionales como el OpenDocument y sus beneficios, y ayudar a que se perpetúe un software condenado reiteradamente en los tribunales por sus abusos delictivos.

Colaboración es una de las palabras claves del software libre. Crece como la ciencia, con la colaboración de todo el que pueda aportar algo y hay muchas formas de poder hacerlo. Es evidente que un equipo colaborando rinde mucho más que si todos intentan hacer todo por su cuenta sin compartir nada. Hay colaboración entre todas las partes, usuarios, desarrolladores, empresas, todos ayudan a todos de alguna forma redundando en beneficio mutuo. "Envié el viernes un email al desarrollador y no solo me respondió amablemente y me dijo que me apuntase a la lista de correo, sino que además ya lo ha solucionado y puedes bajarte la versión mejorada".

Algunos piensan que no es aplicable la palabra competitividad. Pues bien, discrepo. Si ponemos a un lado el software privativo, máximo exponente, según algunos, de competitividad en el mercado, y en el otro lado al software libre



equivalente. Pienso que aunque a veces pueda ir a un paso más lento por falta de fuertes inversiones puntuales, el altruismo de miles de personas en todo el mundo puede ser muy competitivo. No sé si hay empresa que pueda invertir el equivalente a lo que la cooperación altruista de miles de personas en todo el mundo a largo plazo. La situación actual demuestra que el GNU/Linux está siendo muy competitivo pues es usado por muchas de las más prestigiosas empresas y entidades como Google y la NASA, por sus cualidades, y el Apache, por ejemplo, es el servidor de páginas web más usado.

La eficacia del software libre también refuerza su competitividad. La eficacia le viene de su forma científica de desarrollo. En el sentido de que al exponer públicamente su código, éste se ve enriquecido por las aportaciones de otros y por la propia exigencia de calidad del desarrollador que desea entregar a la comunidad algo bueno y que no sea demasiado fácil criticar. Dos ojos ven más que 1 y 400 mucho más por lo que si la aplicación es muy usada sus problemas se pueden solventar antes y las mejoras propuestas son muchas y más rápidamente implementadas.

En ese sentido el software libre publicado es más científico. No quiero decir que se desarrolle basándose en un estudio más cuidadoso y con métodos más cuidadosamente diseñados y organizados. En este caso, decimos que es un software científico por que trabaja como lo hace la ciencia en general, con transparencia no solo en los resultados y publicación el material clave, el código fuente, para revisión publica en espera de posibles mejoras, críticas y divulgación máxima. Hasta en las escuelas se puede aprender no solo su uso sino también su desarrollo en los niveles de edad adecuados. No hay secretos tecnológicos. Es todo divulgación y cooperación para aprender lo que se quiera.

Y claro, en cuanto a la seguridad, eso redundaba en una mayor revisión de los errores y correcciones más rápidas. Aquí no ocultamos los bugs, aquí se reportan continuamente por el bien de todos y se corrigen de media muchísimo más rápido que en el privativo. Y por supuesto siempre solemos tener la última versión disponible. No nos dicen después de pagar eso de "el bug que usted necesita arreglado, ya lo está en la versión siguiente que debe usted comprarse y olvidar la que tiene" Como ejemplo un botón, hace más de 5 años que uso software libre 100%. Nunca he usado antivirus y nunca he tenido ese problema.

Privacidad, :). Doy charlas defendiendo el valor de la privacidad porque lo considero incluso más básico que la libertad. El mundo moderno está haciéndonos perder poco a poco parcelas de libertad de forma alarmante para quien tiene una visión de conjunto. Hay muchos ejemplos en el software privativo, de los que nombraré aquí dos: 1- los programas que entran en tu ordenador e informan a gente en otra parte del mundo del software que tienes. Y 2- el hecho de que los títulos de las películas que ves con muchos programas privativos de visualización de vídeo circulen para informar a "alguien". ¡Ambas cosas sin que el usuario ni lo sepa! Ahora incluso el control de tu PC con el pernicioso DRM⁽¹⁶⁾ ya no te pertenece, ya no eres tú quien decide qué puedes hacer con él. Naturalmente la comunidad del software libre considera no ética esta forma de actuar y su transparencia evidencia fácilmente ese tipo de corrupciones.

Es software libre es No discriminatorio porque todo el mundo, sea del grupo que sea, o aunque no sea de ninguno, puede usar igualmente el software sin menoscabo de sus posibilidades. Usarlo, mejorarlo, contribuir, aprender de él ... todo igualmente accesible a todos. Actualmente, por ejemplo, compañías en todo el mundo menosprecian a comunidades lingüísticas pequeñas olvidándose de producir versiones en su lengua e impidiendo que otros lo hagan. Eso no es ético, se debe permitir que todo el mundo pueda traducir un software. Nosotros tenemos traducciones de nuestros programas hasta en Klingon. :)

La libertad es el principal valor del software libre. Libertad para:

- poder participar en el mundo tecnológico actual y no ser excluido
- poder elegir el programa que más te guste sin que te incomoden con incompatibilidades puestas a propósito
- que no te obliguen a cambiar de versión y perder dinero cada vez que lo necesiten
- Que puedas escuchar tu música o vídeos aunque no sean del circuito comercial habitual
- que no te obliguen a tener problemas de seguridad estúpidos
- que no decidan/ordenen otros cuándo tu ordenador no debe seguir funcionando
- ...
- (hay muchos más, no es una lista exhaustiva)

La libertad está intrínsecamente implicada en todas partes en el software libre, sería muy extenso entrar en detalles; así que me voy a limitar a comentar dos puntos curiosos que he aprendido pensando sobre software libre y el valor de la libertad.



1. Donde hay libertad hay diversidad. El software libre es una vivísima muestra de ello. No tenemos un navegador sino muchos, no tenemos un cliente de correo sino muchos, ... tenemos muchas versiones de casitodo lo más relevante. Aunque pueda parecer un defecto, es el efecto natural de la libertad. Y eso me lleva al segundo punto:
2. El software libre contribuye a la autorealización de mucha gente que le dedica tiempo como hobby o pasión. En el fondo queremos la libertad para hacer cosas, para seguir nuestro propio camino y ser feliz con lo que particularmente podamos hacer. Tengo la firme opinión de que contribuir de forma significativa a un bien público es una de las vías más sanas de complementar la propia autorealización. Es sano tener algunas pequeñas pasiones y ésta es, a mi gusto, de las mejores. (mejorar técnicamente, establecer relaciones humanamente e intelectualmente interesantes, satisfacción de ver que algo tuyo es apreciado y sorprendentemente muy usado, libertad para dejarlo y retomarlo cuando quieras o cambiar de ámbito, campo abierto por todos lados para elegir lo que prefieras ...)

La comunidad del software libre

La comunidad del software libre la componen tanto los usuarios como los desarrolladores. Los USUARIOS, es decir las personas o entidades que usan los programas, suelen ayudar muchas veces solicitando funcionalidades o reportando bugs. Un bug, que en inglés es un bicho, es el nombre que se le da a un error que a menudo suele parecer escondido entre miles de líneas de programa. Ejemplos de usuarios, que muchas veces también participan en el desarrollo son la UNESCO, la Wikipedia, y compañías punteras como Google que paga cada verano un montón de dinero para promocionar funcionalidades al software libre o la mismísima NASA.

DESARROLLADORES son todos aquellos que producen principalmente programas; pero también se consideran desarrolladores los que producen sonidos, imágenes, iconos, manuales, plantillas y todo aquello que se pueda adjuntar a un programa. También habría que nombrar a aquellos que ayudan a su divulgación en revistas, libros, eventos abiertos al público, que sin ser desarrolladores, también contribuyen a promocionar su uso.

Desarrolladores hay de muchos tipos, empresas como IBM, SUN o NOVELL, hackers (gente con habilidad y gusto por la informática), gurúes (en este caso, gente diversa con propuestas sobre software), hay organizaciones grandes como KDE (de K Desktop Environment, el proyecto detrás del conocido entorno de escritorio), Gnome (otro proyecto con otro escritorio más simple), Enlightenment (un escritorio especialmente rápido y vistoso), GNU (proyecto hermano principal de la FSF donde se desarrolla software y por cierto se mantiene el mapa conceptual).

Normalmente el diálogo entre todos ellos y con los usuarios suele respetar unas normas de convivencia llamadas netiqueta. Normas como por ejemplo: "antes de preguntar y hacer que alguien te busque la información mira tú mismo en el Google o la Wikipedia y si no lo encuentras o no lo entiendes tras un rato, entonces pregunta diciendo que pasos has dado."

Así pues, toda esta amalgama de personas y entidades dispersas por todo el mundo potencia que se vaya usando el software libre y que este vaya adoptando MEJORAS y CALIDAD a base de ir tomando, los unos de los otros, ideas, programas ya hechos, informes de bugs y produciendo incrementalmente mejores programas. No se reinventa la rueda continuamente. Cuando ya existe algo implementado, es muy habitual rehusarlo mejorándolo.

Sin ninguna duda, el software libre usa ESTÁNDARES ABIERTOS. No estamos interesados en obligar al usuario a usar un determinado programa a través de imponerle el uso de un estándar de facto que luego le dificulta extremadamente usar otros programas. Todo lo contrario. En todas las universidades y escuelas superiores se enseña el valor de los estándares internacionales como el OpenDocument. Permiten interoperatividad y hacen que la competencia no se centre en cautivar al usuario impidiéndole cambiar de programa, sino en darle funcionalidades que le representen un enriquecimiento. Ejemplos son el HTML (a pesar de que el navegador de Micro\$oft no interprete bien las páginas es un estándar del W3C), los RFC (Request for comments) que valen como estándares, el XML ... Estándares que tienen detrás entidades internacionales de mucho peso como la W3C, IETF, IEEE

Los usuarios y los desarrolladores se ayudan entre ellos habitualmente a resolver las típicas dudas en los LUGs (local user groups, grupos locales de usuarios), en listas de correo (que son direcciones de correo que reenvían todo lo que les llegue a todos los que se apunten), en los foros (webs donde se preguntan y responden unos a otros añadiendo comentarios), wikis (webs colaborativas donde todo el mundo puede ir editando partes), blogs o weblogs (webs configuradas como diarios personales o bitácoras), y en chats (IRC). Todo ello constituye un SOPORTE muy funcional sin equivalente en el software no libre, que añade valor extraordinario a los programas, un valor que además es muchas



veces gratuito. Valux es un ejemplo de grupo local de usuarios de valencia. La Bulmailing sirve como ejemplo de una lista de correo muy activa donde se resuelven dudas de usuarios y desarrolladores y freenode o OFTC son servidores de IRC (chat) donde hay canales como #debian-es para poder preguntar sobre Debian en castellano.

Gracias a Nacho de es.gnu por algunas ideas que me envió por email para mejorar el artículo. :)

Lista de enlaces de este artículo:

1. <http://bulma.net/body.phtml?nIdNoticia=2528>
2. <http://bulma.net/body.phtml?nIdNoticia=2260>
3. <http://es.gnu.org>
4. <http://www.es.gnu.org/~reneme/map/mapa-conceptual-software-libre.png>
5. <http://www.es.gnu.org/~reneme/fsmap/es/maps-guide-es.svg>
6. <http://bulma.net/body.phtml?nIdNoticia=2448>
7. <http://bulma.net/body.phtml?nIdNoticia=2428&nIdPage=2>
8. <http://bulma.net/body.phtml?nIdNoticia=2428&nIdPage=3>
9. <http://bulma.net/body.phtml?nIdNoticia=2428&nIdPage=4>
10. <http://bulma.net/body.phtml?nIdNoticia=2428&nIdPage=5>
11. <http://bulma.net/body.phtml?nIdNoticia=2428&nIdPage=6>
12. <http://es.wikipedia.org/wiki/svg>
13. <http://www.es.gnu.org/~reneme/map/map-es.svg>
14. <http://www.fsf.org/>
15. <http://www.opensource.org/>
16. <http://www.defectivebydesign.org/>

E-mail del autor: ochominutosdearco_ARROBA_gmail.com

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=2428>