



Aspectos Legales del Software de Fuentes Abiertas (II)



Manuel Palomo Duarte
*Oficina de Software Libre y
Conocimiento Abierto (OSLUCA)*
Universidad de Cádiz

Índice

- Usuarios de software de fuentes abiertas (SFA)
 - Tipos de usuario: finales, empresa, admón. pública
- Proyectos de creación de FOSS
 - Caso básico, medio y avanzado
- Integrar SFA
 - Arquitectura, licencias y garantías
- Bibliografía (libre, claro ;-)

El usuario de SFA

- Tipos de usuarios:
 - Usuario final
 - Empresas
 - PYME
 - Empresas grandes
 - Administración pública
 - Copyleft y garantías

Usuario final

- Características generales:
 - Decisión sobre el software usado “difusa”
 - Recomendaciones poco rigurosas
 - Inmediatez de instalación y facilidad uso
 - Existencia de comunidad en el idioma que solucione sus problemas
 - Baja comprensión de las implicaciones del SFA:
 - ¿Diferencia gratis / libre / “privativo pirata”?
 - Posibilidades (petición de funcionalidades, informe de errores, etc)

Usuario final

- Existen notables excepciones:
 - Usuarios con interés por la tecnología (y/o principios)
 - Pueden informar de fallos
 - Ayuda en la traducción y documentación básica
 - Participar en actividades de difusión, apoyo a noveles
 - Proyectos amateur, homebrew, etc.
 - Ejemplo
 - El líder del proyecto Synfig trabaja como administrativo de la Junta. En su tiempo libre lidera la comunidad.

Empresa PYME

- Características generales (I):
 - Departamento TIC normalmente no existe
 - La informática suele ser una fuente de problemas
 - Baja capacidad de decisión sobre el software
 - A veces nula: formatos cerrados, administración electrónica, mantenimiento de sistemas “obsoletos”
 - Baja comprensión de las implicaciones del SFA:
 - ¿Diferencia gratis / libre / “privativo pirata”?
 - Posibilidades (petición de funcionalidades, informe de errores, etc)

Empresa PYME

- Características generales (y II)
 - Baja inversión en TIC
 - Entorno poco tecnológico
 - Baja productividad y retorno de la inversión
 - Ausencia de soluciones adecuadas
 - Poco o nada de CRM, ERP
 - Abuso de aplicaciones ofimáticas generales
 - No se valora la independencia del proveedor
 - Al contrario, suelen recibir asesoría de su único proveedor

Empresa PYME

- Existen excepciones
 - Cada vez aparecen más empresas de base tecnológica, pin-offs, emprendedores, etc
 - Red de Empresarios de Internet Cádiz (REI)
<http://www.larei.net/>
 - Andalibre
 - Tienen necesidades más de ingeniería:
 - Certificaciones AENOR/ISO
 - Necesitan software fiable, ágil, interconectado, etc.

Empresas grandes

- Características generales
 - Tienen departamento TIC
 - Su director a veces es la persona más capacitada
 - Pero otras veces es el empleado que más tiempo lleva
 - Modelo plantilla contratada *versus* plantilla de becarios
 - Tienen necesidades software vitales
 - Gestión contable, stock, etc.
 - Administración electrónica
 - Contacto con proveedores, etc.
 - Información en tiempo real (PDAs, etc)

Empresas grandes

- Uso de software libre “sin saberlo”
 - Fruto de adquisiciones de otras empresas o implantaciones realizadas por empresa externas
 - Ejemplo: mundo del videojuego
- Uso consciente de software libre
 - Servidores: LAMP, proxy, etc.
 - Ahorro de licencias: OpenOffice (ejemplo IBM), etc.

Empresas

- Varios niveles de relación con el software libre, de menos a más:
 - Usuario (básico o avanzado)
 - Contribuyente
 - Sponsor
 - Colaborador
 - Liderazgo
- En detalle ... con sus implicaciones legales

Empresas usuarias de SFA

- Como usuarias básicas de software libre:
 - Son conscientes de las ventajas básicas que emanan de las libertades que otorga la licencia
 - Principalmente copia y distribución
 - Soporte de la comunidad
 - Ojo, normalmente el software es proporcionado sin garantías (“as is”):
 - Es más, a veces tiene un fichero BUGS.txt que indica que el software falla. *¿Nos fiamos de él?*

Empresas usuarias de SFA

- Como usuarias avanzadas de software libre:
 - Son conscientes de las ventajas avanzadas que emanan de las libertades que otorga la licencia
 - Independencia de proveedor
 - Interconectividad
 - Menor presión por cambio de versión del software
 - Participación básica en la comunidad: notificación errores, traducc./l10n (ej. *smbwebclient* en JASL4)
 - Consulta el código fuente

Empresas contribuyentes al SFA

- Como contribuidoras a un proyecto libre:
 - Desarrollan software que se integra en el proyecto
 - Normalmente mantienen el compromiso de actualizar y mantener dicho software
 - Es importante hacerlo bajo una licencia compatible con la del proyecto (si es la mismo, mejor)
- Ejemplo: empresas que aportan al proyecto Apache (que no está liderado por ninguna)

Empresas patrocinadoras de SFA

- Como esponsorizadoras a un proyecto libre:
 - Dedican recursos económicos al sustento del proyecto: desarrollo externo, servidor web, alojamiento, promoción, jornadas, etc.
 - Participan activamente en la gestión del proyecto: deciden qué tareas del TODO se hacen con su dinero
 - Debe exigir garantías sobre su inversión
 - El dinero suele ser para las tareas más “duras”
 - Cuidado con información confidencial

Empresas colaboradoras de SFA

- Como colaboradora de un proyecto libre:
 - Dedicar recursos a la dirección del proyecto: programadores, empaquetadores, etc.
 - Participan más activamente en la gestión del proyecto: qué tareas del TODO se priorizan
 - Forman parte del núcleo duro:
 - Gestión de código, participación, etc.
 - A veces reciben el sello de “empresa certificadora” o *partner* del producto

Empresas líderes en SFA

- Como líder de un proyecto libre:
 - Último responsable del proyecto
 - Importancia de hacer el proyecto estable, duradero, etc.: guidelines, tecnologías, web, etc.
 - Gestión de marcas, propiedad intelectual
 - Aspectos de gestión:
 - Abogados, protección frente a demandas, alianzas, participación en asociaciones, popularización, certificación del software, etc.
 - Respeto a las normas no escritas (capítulo sig.)

Administración pública

- Existe un marco muy favorable en EU:
 - UNU-Merit: “... *Competitiveness of FLOSS in EU ...*”
 - Casos de éxito: *Interoperable Delivery of European eGovernment Services to Public Administrations, Businessse*. Ejemplo: migración Munich, Opendia
- También en España:
 - Propuesta de recomendaciones a la Administración General del Estado sobre utilización del software libre y de fuentes abiertas (MAP, junio de 2005)
 - ...

Administración pública

- Más normativa en España:
 - Capítulo III del Título IV de la Ley 11/2007 de 22 de junio, de Acceso Electrónico de los Ciudadanos a los Servicios Públicos (LACESP), titulado “Reutilización de aplicaciones y transferencia de tecnologías”
 - Disposición adicional decimosexta de la Ley 57/2007, de 28 de diciembre, de Medidas de Impulso de la Sociedad de la Información (LISI).
 - Ejemplo: repositorio de la Junta de Andalucía

Copyleft y garantías

- ¿El copyleft obliga a publicar el código?
 - No, sólo a dárselo a quien acceda al programa (por ejemplo comprándolo). Tampoco al autor original
 - Sí que obliga a reconocimiento autoría
- En algunas empresas puede ser adecuada la no publicación de “secretos” de negocio
 - Cuidado, sí hay que dárselo a:
 - Subcontratas que usen software
 - Empresas que presten servicios (por ejemplo, *hosting*)

Copyleft y garantías

- Aunque el copyleft no obliga, se recomienda:
 - Mantener buenas prácticas:
 - Distribuir scripts de instalación
 - Facilitar la documentación del sistema
 - Revertir las mejoras a la comunidad (al menos los errores arreglados)
 - No crear *forks* innecesariamente
 - Etc.

Garantías

- Problemas y soluciones
 - Limitaciones de garantías de funcionamiento
 - ¡Igual que el software privativo!
 - Infracciones de derechos de terceros (ej. plagio)
 - Empresas locales de soporte
 - Responsabilidades ante terceros
 - Empresas intermediarias (Red Hat, SUSE, etc)
 - Garantía de futuro del proyecto
 - El acceso al código da más garantías que soft. privativo

Proyectos de creación de FOSS

Pregunta

De los asistentes al curso
¿Quién ha desarrollado software?

De los asistentes al curso
*¿Quién ha desarrollado software **libre**?*

¿Ha sido muy distinto desarrollar software
privativo y libre?

Proyectos de creación de FOSS

- Caso básico: todo el software es mío
 - Es raro hoy en día: las bibliotecas son software y tienen licencia
 - Aún así, si libero un script o similar hay que:
 - Garantizar la titularidad del código
 - Seleccionar la licencia
 - Gestión de marcas, logos, etc.
 - Administrar la distribución del software
 - **Forja: ¡vamos por el bazar!**

Forja

- Es un sistemas web que proporcionan (gratis) muchos recursos para trabajo colaborativo:
 - Repositorio de software (SVN, Git)
 - Foros, listas de correo, wikis
 - Web
 - Asignación de tareas y seguimiento de errores
 - Noticias RSS
 - Categorización de proyectos
 - ...

Forjas

- En inglés
 - Sourceforge, Freshmeat, GNU Savannah, Gitorius, Assembla, etc
- En español:
 - RedIRIS (Universidad e investigación española), Guadalinux, Proyecto Morfeo, etc

Derechos de autor

- En la rama de la ley europea cada persona es propietaria de los derechos morales de su creación intelectual.
- Pero los derechos de explotación dependerá de si lo crea:
 - Un equipo de desarrollo de una empresa
 - Los derechos serán de la empresa
 - Si hay personal externo, becarios, proveedores
 - Pedir cesión de derechos por escrito mejor que implícita

Liberación de software

- Cuidado si el software contiene
 - Métodos susceptibles de patentarse
 - Esperar para liberarlo
 - Contienen parte confidenciales, secretas, etc
 - Liberarlo parcialmente (comentando las limitaciones)
- Recomendable
 - Tener una web (propia o en forja) y un logo
 - Hacer comunidad
 - Incluir avisos de limitación de responsabilidad

Proyectos de creación de FOSS

- Caso medio: mi software usa otro SFA
 - Ver si las licencias de todo el software que uso son compatibles con la licencia del mio
 - Ejemplo:
 - <http://confluence.sakaiproject.org/display/LIC/3rd+Party+Licenses+and+Software>
 - http://www.zimbra.com/license/zimbra_third_party_licenses_2.1.html
 - Las licencia libres (aprobadas por FSF) permiten uso, modificación y distribución. Pero pueden poner condiciones (por ejemplo, las licencias víricas)
 - Ver quién está detrás del software que voy a usar
 - Si da garantías, mantenimiento, si tiene comunidad, etc.

La redistribución

- Normas clásicas de redistrib. de código:
 - Mantener autoría y notas de cambios
 - Incluir copia de la licencia
 - Respetar uso de marcas
- Recomendable:
 - Informar (y alabar ;-)) al titular del uso del software
- Analizar las condiciones de uso de marcas, logos, etc (normalmente son públicas)

Proyectos de creación de FOSS

- Caso avanzado: proyecto con realimentación
 - Los usuarios envía contribuciones a mi software y/o al software que integro
 - Dos formas de “acuerdo sobre contribuciones”:
 - La aportación se realiza con la misma licencia que el proyecto o compatible
 - Se ceden derechos al proyecto, al menos los derechos de reproducción, transformación, comunicación pública y distribución sin limitaciones
 - Ejemplos: MySQL, Apache u OO.org

- **Revisión legal:**
 - Revisar las licencias y obligaciones de cada componente
 - Incluir las cabeceras de los ficheros de código original, y los avisos de autoría (originales y modificaciones) y de licencia del proyecto.
 - Incluir la documentación jurídica de todos los componentes (software, documentación, etc): “notice.txt”, “licensing.txt”, etc
 - Calcular y publicar el hash del paquete publicado
 - Acompañar el binario con los fuentes (o indicar dónde están)

Buenas prácticas al publicar

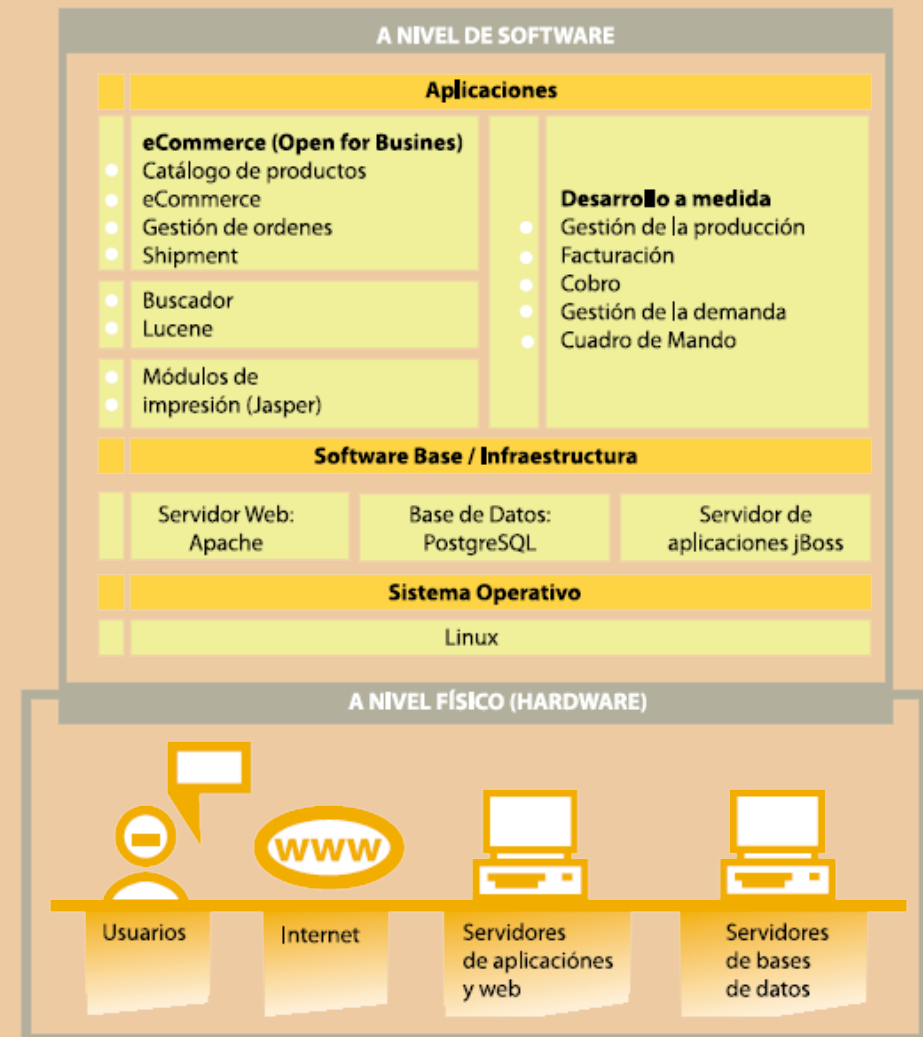
- Buenas prácticas:
 - Reconocimiento de los autores en el programa (sección de créditos, “Acerca de”)
 - Si es un ejecutable distribuido “sólo” hacer notar que está sujeto a una licencia durante su instalación o primer uso.
 - Redactar una política de uso de nombres y marcas
 - Poner la información legal también en la web del software
 - Cumple con la legislación vigente (LSSICE, LOPD)
 - Si se ofrecen servicios, respetar las leyes sobre la protección del consumidor y usuarios

Ejemplos

- En la “Guía del derecho y el Software de fuentes abiertas. CENATIC” se incluyen tres ejemplos:
 - Netscape libera Mozilla
 - OpenBravo (ERP libre líder, creado en Navarra)
 - Proyecto Campus (Generalitat Cataluña)

Integrar SFA

- Los modelos empresariales SFA plantean cuestiones similares a crear un proyecto SFA:
 - Uso e integración
 - Licencia distribución
 - Garantías y responsabilidades



Uso e integración de componentes

- Al igual que al hacer un proyecto SFA avanzado, tengo que ver:
 - Qué componentes necesito
 - Alternativas (distribuciones, forjas, wikipedia, etc)
 - Licencia de cada una
- Depende del modelo que quiero:
 - Vender licencias (componentes licenciados BSD)
 - Negocio en el servicio (cualquier licencia libre)
 - Mixto

Garantías

- La limitación de garantía suele aceptarse en derecho español excepto en sistemas críticos
 - Aclarar en los contratos
 - Limitaciones de responsabilidades
 - Propiedad intelectual: autoría, pago (por servicios o licencias), etc.
 - Garantías cubiertas: uptime, funcionamiento correcto
 - Almacenamiento del código fuente: quién lo hace
 - Licencias sobre el software recibido
 - Certificaciones del producto

Referencias / Agradecimientos

- Documento basado en:
 - Guía del derecho y el software de fuentes abiertas. CENATIC
 - Transparencias del curso “Aspectos Legales del Software de Fuentes Abiertas”. J. Antonio Serrano García. Oficina de Software Libre, Univ. Granada
- Consulta:
 - M2114 - Aspectos legales y de explotación del software libre (OpenCourseware)

Bibliografía adicional

- Richard M. Stallman: *Software libre para una sociedad libre*
- Eric S. Raymond: *La catedral y el bazar*
- Lawrence L. Lessig: *El código 2.0, El futuro de las ideas y Cultura libre*
- Karl Fogel: *Producing Open Source Software*
- Andrew M. St. Laurent: *Understanding Open Source and Free Software Licensing*

© Manuel Palomo Duarte, 2009

Gracias por vuestra atención.
¿Preguntas?

manuel.palomo@uca.es



Universidad de Cádiz
Oficina de Software Libre