Software libre

Vicente Matellán Olivera
vmo@barrapunto.com

León, 2 de Octubre de 2002

¿Por qué hablar de software libre?

Desde hace 30 años, nos han acostumbrado a que:

- Quien me vende un programa me dice en qué condiciones puedo usarlo
- Es normal que una empresa mantenga monopolios casi absolutos en un tipo de programas
- Si un programa tiene errores sólo su fabricante puede arreglarlos
- No tiene sentido que quiera adaptar un programa a mis necesidades
- Copiar programas es "malo"

¿Qué es software libre?

software libre $\neq$ software gratis

Quien lo recibe puede:

- usarlo como mejor le parezca, donde mejor le parezca.
- redistribuirlo a quien quiera, por los medios que quiera.
- modificarlo (y mejorarlo o adaptarlo).
- redistribuirlo con o sin sus modificaciones

Imprescindible: disponibilidad del código fuente.

¿Y por qué es esto y no otra cosa?

Desde luego no es casualidad...:

- Motivos éticos: porque las cosas deberían ser así.
- Motivos prácticos: porque las cosas funcionan mejor así.

Largas discusiones, que han asentado cierto consenso:

- Debian Free Software Guidelines,
- Definición de "Open Source".
- Definición de software libre de la FSF

La ética del programador

Un buen programador debería:

- contribuir con su trabajo a la Comunidad.
- poder aprovechar el trabajo de otros buenos programadores.
- poder "arreglar" y mejorar cualquier programa.
- sentirse orgulloso de usar su código, y de que otros lo usen.

Buen programador = Hacker

Ideas formuladas por Richard Stallman, la FSF, la comunidad BSD...

¿Y los argumentos prácticos?

- Nuevos modelos de desarrollo (bazar frente a catedral).
- Ventajas del escrutinio público y de la mejorabilidad.
- Competencia real en el desarrollo y el mantenimiento.
- Viabilidad técnica frente a mercadotecnia.
- Nuevas posibilidades de negocio (ej: desafío a posiciones de monopolio).

Ideas formuladas por Eric Raimond, promovidas por la Open Source Initiative y otros.

Consecuencias de la "libertad" del software

- **Coste**: modelo de costes distinto al del software propietario.
- **Apertura**: puede modificarse, inspeccionarese, estudiarse.
- **Distribución**: nuevos canales, nuevos métodos.
- **Desarrollo**: modelos de desarrollo "sorprendentes".
- **Mantenimiento y soporte**: Verdadera competencia.
- **Aprendizaje**: alumnos que pueden "leer" código bien escrito

Combinación de dos poderosos mecanismos:

- Competencia (pudiendo usar el mismo programa base)
- Cooperación (incluso involuntaria)

La importancia de las licencias

Las licencias son las herramientas legales que imponen ciertos requisitos a los usuarios y a los redistribuidores.

- Licencias minimalistas: BSD, X Consortium, Apache.
- Licencias diseñadas para "proteger" a la comunidad: GPL
- Licencias diseñadas por empresas para explorar el software libre: MPL, IBM, etc.

Cada licencia refleja una forma de concebir el mundo del software libre.

¿Y por qué me interesa todo esto?

- La visión del usuario final (individual o empresa).
- La visión del desarrollador (o productor de software).
- La visión del integrador.
- La visión del que proporciona mantenimiento y servicios.

El usuario final

Están los "olvídate"...

- Olvídate de los monopolios (verdadera competencia, mejores productos, mejores servicios).
- Olvídate de la "fiabilidad" del productor (el futuro lo asegura la aceptación del producto, y la disposición del fuente).
- Olvídate de tomar decisiones con pocos elementos (puedes probar el software en su entorno real a coste prácticamente cero).
- Olvídate de depender de la estrategia de tus proveedores.

El usuario final

...y los "¿qué tal si...?"

- ¿Qué tal si pudieras adaptar/personalizar el producto como quieras?
- ¿Qué tal si pusieras "estar a la última" a bajo coste?.
- ¿Qué tal si pudieras arreglar los problemas (o pagar para que los arreglen?
- ¿Qué tal si pudieras decidir sobre la evolución futura del producto?
- ¿Qué tal si pudieras contratar la integración de los dos mejores productos en el entorno que te interesa?

El usuario final

Gran parte del control pasa al usuario (frente al productor de software).

El desarrollador/productor de software

El software libre cambia las reglas del juego.

- Puedes competir siendo pequeño.
- Es mucho más fácil adquirir tecnología punta (y más barato).
- Te puedes aprovechar del trabajo de tu competencia (ojo: también tu competencia del tuyo).
- Si lo haces bien, puedes conseguir, a bajo coste, la colaboración de mucha gente.
- El canal de distribución es mucho más barato, y global.
- Es posible convertirte en aplicación de referencia mucho más fácil.

El desarrollador/productor de software

¿Y de dónde saco el dinero?

- El mejor conocimiento sobre el programa lo tiene su desarrollador.
- Si se cuida la imagen, el desarrollador es el "punto más visible".
- Desarrollos a medida, modificaciones, personalizaciones.
- Soporte "a lo grande" (corrección de erratas, acceso preferente a nuevas versiones, nuevas características, etc.)

Si hay gente que quiere software, y está dispuesto a pagarlo, algún desarrollador/productor se beneficiará...

El integrador

¡Bienvenido al paraíso!

- Todos los productos libres están a tu disposición (¡y sin preocuparte de licencias propietarias!).
- Si los productos no "encajan", puedes "limarlos" (tienes el código fuente, puedes conseguir interoperabilidad).
- Puedes integrar "trozos" de productos, o productos enteros, o lo que sea.
- No más cajas negras: las tripas de todo son transparentes.

Puedes construir sobre el trabajo de otros, en igualdad de condiciones con esos otros.

Mantenimiento y servicios

El disponer del fuente lo cambia todo.

- Estás en las mismas condiciones que el productor.
- Competencia en el negocio del mantenimiento.
- El valor añadido de los servicios es mucho más apreciado (el coste del programa es bajo).
- El conocimiento del estado del arte es muy importante (es bueno tener relación con los proyectos libres).
- Negocios nuevos: consejo sobre versiones y combinaciones de programas, información sobre nuevos desarrollos, gestión de proyectos libres.
- Este es actualmente el negocio más claro.

Principales obstáculos

El software libre está demostrando estar aquí para quedarse, pero pueden presentarse problemas:

- Técnicas FUD (miedo, desconocimiento, duda): hasta ahora han mostrado no ser muy problemáticas.
- "Disolución" (sistemas que pueden confundirse con el software libre): división de la comunidad, pérdida de las ventajas del modelo.
- Desconocimiento (pérdida de visión): ¿por qué es interesante el software libre?
- Impedimentos legales: por ejemplo, patentes software.

Y habrá más...

¿Hay conclusiones?

- Aún hay pocos casos para estar seguros de por dónde saldrá todo esto.
- Pero hay muchas buenas perspectivas.
- ¿Eres competitivo?: en este modelo tienes muchas ventajas.
- ¿Eres pequeño?: en este modelo tienes muchas ventajas.
- Se está experimentando con nuevos modelos de negocio.
- Hace falta mucha innovación, imaginación... pero también conocimiento del entorno.
- Nunca ha sido tan importante tener información buena, y de primera mano.

Aún quedan problemas por resolver... ¿o son oportunidades de negocio?

¿Hay conclusiones?

- El software libre muestra ser un modelo económica y técnicamente viable.
- Detrás de él hay motivaciones técnicas, económicas y éticas.
- Es muy importante conocer el mundo en que nos movemos...
- El futuro depende, en gran parte de nosotros (como profesionales, como clientes, como empresarios,...).

Este es uno de esos raros momentos en los que toda una industria puede estar cambiando de paradigma.

Algunas URLs

- Free Software Foundation: `http://www.fsf.org`
- Open Source Initiative: `http://www.opensource.org`
- Grupo de trabajo de la Comisión Europea sobre software libre: `http://eu.conecta.it`
- Curso de doctorado sobre software libre: `http://curso-sobre.berlios.de`
- Open Sources (O'Reilly) `http://www.openresources.com/documents/open-sources`
- BarraPunto: `http://barrapunto.com`