La empresa ante el software libre.

Juan Antonio Martínez, *jantonio@dit.upm.es* v1.0, 11 Abril 1999

Este documento constituye un ensayo sobre la aplicación del modelo de desarrollo del software libre a la empresa y la economía de mercado. Se describen fundamentos sociológicos y modelos matemáticos; se analizan aspectos legales y políticos, y se detallan las diferentes aproximaciones y metodologías de trabajo que puede adoptar la empresa a la hora de hacer suyo dicho modelo.

Contents

1 Prólogo.

El software libre está de moda. Internet está de moda. Dos hechos que no son entendibles el uno sin el otro. Se ha escrito mucho acerca de estos fenómenos, de sus repercusiones y de su futuro.

Pero un nuevo elemento acaba de entrar en escena: las empresas comerciales han visto en el software libre, primero un competidor, y posteriormente una fuente de negocios. Este efecto nunca había sucedido, y mucho menos ha sido estudiado.... ¿o si?

Porque a la hora de plantearme la pregunta básica de qué es lo que impulsa a una empresa cuyo objetivo es la obtención de ganancias económicas, a participar en proyectos, aparentemente altruistas, realizados por voluntarios, sin más herramienta de coordinación que Internet, he tenido que consultar multitud de fuentes, de tratados de filosofía, sociología, economía... buscando modelos matemáticos, e incluso textos sobre el comportamiento animal. No creo haber descubierto nada nuevo, sino que este ensayo es un intento -espero que feliz- de adaptar teorías aplicadas en otros campos del conocimiento, a las técnicas de desarrollo del software libre, con el fin de demostrar su viabilidad comercial

Reconozco que la formación excesivamente técnica ha sido y es un serio obstáculo para mostrar al lector un desarrollo coherente. Pido disculpas por ello, y aceptaré gustoso las correcciones y anotaciones que los lectores me hagan llegar.

Quiero dar gracias a todas las personas que han hecho que un aspirante a ingeniero como yo, se apasione con la sociología. Es triste que la orientación de los estudios de ingeniería en las universidades españolas tienda a dejar a un lado lo que debería ser un pilar básico de la formación del ingeniero, convirtiéndolo en poco más que una asignatura marginal.

Este ensayo, presupone que el lector está familiarizado con el concepto y definición de software libre y de su modelo de desarrollo. Así, conceptos como "software libre y software propietario", "la catedral y el bazar", así como las diversas licencias de software libre son conocidas y comprendidas por el lector. En caso de no ser así remito al lector a la bibliografía incluida en el Apéndice. Recomiendo la lectura del ensayo *"Apuntes sobre Software Libre"* de Jesús González Barahona, como lectura obligada para todo aquel que sea nuevo en este tema.

Una última recomendación: algunos párrafos de este ensayo son duros, pues muestran un punto de vista sobre el software libre, que muchos considerarán falto de ética e incluso contrario a su filosofía. El concepto de moral utilitarista es muy políticamente incorrecto, y su exposición dará sin duda lugar a polémicas. Una lectura sin prejuicios ayudará a comprender nuevos puntos de vista, que no necesariamente son compartidos por el autor.

Un Saludo

Juan Antonio Martínez Castaño

Madrid, Marzo-Abril de 1999

2 Copyright. Licencia de uso.

2.1 Nota del autor.

Cuando me planteé el escribir este ensayo, la primera idea fue el ponerlo bajo licencia GPL. Desgraciadamente mientras desarrollaba los diversos apartados me hice plenamente consciente del problema que existe en la actualidad con el tema de la "documentación libre", así como de las discusiones que existen en estos momentos en Internet acerca de un modelo de licencia abierta para la documentación. En el texto del ensayo, describo esta problemática y las soluciones propuestas.

Del mismo modo el primer borrador incluía una cláusula para evitar la distribución impresa. La idea original era poder editar el ensayo en alguna editorial. Como más de uno me ha hecho ver, este modelo no es -al menos por ahora- realista, y constituye una violación al espíritu del software libre, por lo que he decidido adoptar una licencia más acorde con dicho modelo.

La Open Resources Magazine License que a continuación incluyo puede ser obtenida de *http://www.openresources.com/magazine/license/index.html*.

2.2 Open Resources Magazine License.

This document may be freely read, stored, reproduced, disseminated, translated or quoted by any means and on any medium provided the following conditions are met:

1. Every reader or user of this document acknowledges that is aware that no guarantee is given regarding its contents, on any account, and specifically concerning veracity, accuracy and fitness for any purpose.

2. No modification is made other than cosmetic, change of representation format, translation, correction of obvious syntactic errors, or as permitted by the clauses below.

3. Comments and other additions may be inserted, provided they clearly appear as such; translations or fragments must clearly refer to an original complete version, pr eferably one that is easily accessed whenever possible.

4. Translations, comments and other additions or modifications must be dated and their author(s) must be identifiable (possibly via an alias).

5. This licence is preserved and applies to the whole document with modifications and additions (except for brief quotes), independently of the representation format.

6. Any reference to the "official version", "original version" or "how to obtain original versions" of the document is preserved verbatim. Any copyright notice in the document is preserved verbatim. Also, the title and author(s) of the original document should be clearly mentioned as such.

7. In the case of translations, verbatim sentences mentioned in (6.) are preserved in the language of the original document accompanied by verbatim translations to the language of the traslated document. All translations state clearly that the author is not responsible for the translated work. This license is included, at least in the language in which it is referenced in the original version.

8. Whatever the mode of storage, reproduction or dissemination, anyone able to access a digitized version of this document must be able to make a digitized copy in a format directly usable, and if possible editable, according to accepted, and publicly documented, public standards.

9. Redistributing this document to a third party requires simultaneous redistribution of this licence, without modification, and in particular without any further condition or restriction, expressed or implied, related or not to this redistribution. In particular, in case of inclusion in a database or collection, the owner or the manager of the database or the collection renounces any right related to this inclusion and concerning the possible uses of the document after extraction from the database or the collection, whether al one or in relation with other documents.

Any incompatibility of the above clauses with legal, contractual or judiciary decisions or constraints implies a corresponding limitation of reading, usage, or redistribution rights for this document, verbatim or modified.

2.3 Open Resources Magazine License (Traducción).

Traducido por Juan Antonio Martínez <*jantonio@dit.upm.es*> 24-Marzo-1999

Acorde con los términos de la licencia, el autor de la traducción hace constar que ésta tiene sólo valor orientativo, y en ningún caso legal, declinando el autor de la traducción toda responsabilidad derivada de su uso.

El autor de la licencia original no se hace responsable de los actos derivados de una posible traducción o interpretación incorrecta, siendo la licencia original la única referencia válida a efectos legales o de copyright.

(Comienzo de la traducción)

Este documento puede ser libremente leído, almacenado, reproducido, distribuido, traducido o mencionado, de cualquier manera y mediante cualquier método, siempre que se cumplan las siguientes cláusulas:

1. El usuario o lector de este documento conoce y comprende la advertencia de que no se proporciona garantía alguna sobre su contenido en ninguno de sus puntos, acerca de su veracidad, precisión, y completitud referida a cualquier uso que se haga de este documento.
2. No se autoriza ninguna otra modificación al documento que no sea de aspecto, cambio de formato de representación, traducción, corrección de errores sintácticos evidentes, o aquellas permitidas por las siguientes cláusulas:
3. Se permite la inclusión de comentarios, o añadidos, bien entendido que deben estar claramente marcados y definidos como tales. Las traducciones y fragmentos del texto deben ir acompañadas de una referencia a una versión original completa, preferiblemente una cuyo acceso en cualquier momento sea sencillo.
4. Las traducciones, comentarios, y otras adicciones o modificaciones, deben ir acompañadas de una indicación de la fecha y de una identificación del autor. Se permite el uso de seudónimos.
5. Esta licencia es de aplicación al documento completo, incluyendo modificaciones y añadidos (salvo notas breves), con independencia del formato de representación.
6. Cualquier referencia a la "versión oficial", "versión original", o "cómo obtener la versión original" de este documento deben conservarse literalmente. Cualquier nota de Copyright debe mantenerse literal. Además, el título y autor o autores del documento original, deben ser citados y marcados claramente como tales.
7. En el caso de traducciones, los párrafos literales mencionados en (6.) deben ser conservados en el lenguaje del documento original, acompañados de una traducción literal del documento al lenguaje de la traducción. Todas las traducciones deben marcar claramente que el autor no es responsable del trabajo traducido. Esta licencia debe ser incluida al menos en el lenguaje en que se creó en el documento original.
8. Con independencia del método de almacenamiento, recuperación o redistribución, cualquiera que sea capaz de acceder a una versión digital de este documento, debe a su vez poder ser capaz de realizar una copia digital en un formato directamente utilizable, y si es posible editable, de acuerdo a estándares públicos aceptados y documentados.
9. La redistribución de este documento a terceras partes, requiere que se redistribuya conjuntamente con esta licencia, sin modificación, y en particular sin ninguna otra condición o restricción, expresada o implícita, relacionada o no con esta redistribución. En particular, para el caso de su inclusión en una base de datos o recopilación, el propietario o el gestor de la base de datos o recopilación, renuncia a cualquier derecho relacionado con esta inclusión y concerniente a posibles usos del documento una vez extraído de la base de datos o recopilación, bien de manera individual o en relación con otros documentos.

Cualquier incompatibilidad de las cláusulas anteriores con leyes, contratos, o sentencias judiciales, implica la correspondiente limitación de los derechos de lectura, uso o redistribución de este documento, tanto en su versión original como en versiones modificadas.

(Fin de la traducción)

3 Registro de cambios en el documento.

1999-04-12 Versión 0.9

- Documento final publicado.

1999-04-11 Versión 0.8

- Conclusiones.
- Añadido del epílogo.
- Diversas correcciones ortográficas y de estilo.

1999-04-10 Versión 0.7

- Primera versión SGML.

1999-04-09 Versión 0.6

- Capítulo 7 (política y sociedad).
- Añadidos al capítulo 5.
- Capítulo 4 (perspectiva empresarial).

1999-04-06 Versión 0.5

- Inclusión del capítulo 8 (legislación).
- Añadidos al dilema del preso.

1999-03-29 Versión 0.4

- Inclusión del capítulo 5 (empresa).

1999-03-27 Versión 0.3

- Inclusión de reseñas bibliográficas.
- Escritura del último capítulo.

1999-03-23 Versión 0.2

- Cambio de licencia, adoptando el modelo ORML.

1999-03-22 Versión 0.1

- Primer borrador público en formato texto.

En el texto, el lector encontrará secciones marcadas con [JAMC: < algún texto >>]. Esto indica secciones pendientes de escritura o revisión. Se agradecen comentarios y sugerencias al respecto.

4 Introducción

4.1 Resumen del capítulo.

- Presentación. Objetivos.
- La sociedad la información.
  - Sociedad de la información y sociedad informada.
  - El fenómeno Internet.
  - Reinventar la rueda.
- El fenómeno del software libre.
  - Orígenes y evolución del software libre.
  - La realidad actual.
- El coste del software.
  - ¿Pagar por la herramienta de trabajo?.
  - Realidad de la piratería informática.
  - Minimizar los costes de instalación.

4.2 Presentación. Objetivos.

Este ensayo trata sobre el fenómeno del software libre y sus consecuencias económicas. A la hora de plantear el tema he querido explícitamente huir de todo planteamiento idealista, voluntarista o de juicios morales sobre la bondad del software libre; existe ya demasiada literatura sobre el tema, y todos estamos al día de las características, bondades e inconvenientes del software libre. Pido disculpas anticipadas a todo aquel que sienta herida la idea sobre la "bondad" del software libre, y de la idea del "compartir desinteresado", pero no se puede cerrar los ojos a lo que en estos momentos es una realidad

Utilizo el término "software libre" en el sentido inglés del concepto "Open Source Software", en lugar del entendido en España como "Software gratuito". En todo momento, a menos que se explicite lo contrario, entenderemos como tal a aquel software, generalmente sujeto a derechos de copyright y licencia de uso sobre el que el usuario tiene acceso al código fuente, permitiéndose cierta libertad de uso. Del mismo modo, a menos que se especifique, la licencia de uso corresponderá a la definida en la "GNU Public License versión 2". Es posible que en algún momento en el texto aparezca el traducción literal "software abierto". Pido por ello disculpas anticipadas al lector, que espero tenga la amabilidad de notificar el error, para así corregirlo convenientemente.

Todo escrito tiene un objetivo, y éste no podía ser menos. La idea que quiero transmitir es esta: El Software Libre es un negocio rentable, y subyace en un modelo de economía basado en la colaboración egoísta. Intentaré demostrar que dicho modelo económico no es exclusivo del software sino que es una conclusión lógica de la sociedad de consumo actual, basada en la tendencia de desplazar la economía de los países industrializados desde el sector producción al sector servicios.

El modelo de empresa basada en estos conceptos difiere grandemente de la empresa tradicional, tanto en los objetivos como en los medios disponibles, aunque comparten algo básico: ganar dinero. Veremos como el modelo del Software Libre cumple esto, y además aporta algo de lo que los otros modelos carecen: optimiza la relación coste/prestaciones en toda la cadena de producción, desde el voluntario/colaborador hasta el usuario final del producto.

4.3 La sociedad de la información.

La sociedad de los países industrializados se tiene una serie de características que la hacen destacar sobre otras sociedades: una renta per cápita superior a la media, un desplazamiento de su actividad productiva hacia el sector servicios, una serie de valores económicos y culturales "occidental". Una vez caído el sistema productivo "comunista", la sociedad de mercado con sus ventajas e inconvenientes se ha hecho con la hegemonía de las sociedades industrializadas. Y con ella un fenómeno que en los últimos años se ha hecho patente: vivimos en la sociedad de la información. Tanto es así que nuestro modelo económico se basa en muchas ocasiones para su desarrollo en la posesión y manejo de información mas o menos valiosa o privilegiada. Una frase de periodistas es "Quien tiene la información, tiene el poder".

Sociedad de la información y sociedad informada.

Pero la realidad es otra: El poder no está en quien tiene la información, sino en quien sabe manejarla. En un mundo con superabundancia de fuentes de información (tanto es así que más de un ensayista ha escrito que la sociedad de la información implica el fin de la historia según se describe desde la dialéctica marxista) es un problema real el manejo y proceso de tanta información. Frecuentemente se produce un fenómeno de *overbooking* de información, o bien aparece la problemática de "separar la señal del ruido", es decir, del discernimiento entre información relevante o desechable. No en vano un nuevo concepto ha aparecido dentro del mundo de las tecnologías de la información: el *Data Mining*. Hasta hace unos años era impensable que tal necesidad de control de la información fuera no ya necesaria, sino hasta imprescindible para la supervivencia de una empresa.

El fenómeno Internet.

En este contexto de sociedad de la información, aparece el fenómeno Internet. A nivel sociológico podemos decir que Internet ha convertido el fenómeno de la información y el acceso a ésta en un mecanismo al que todo el mundo puede acceder, sin mas que unos mínimos recursos. Se dice que Internet ha "democratizado" el acceso a la información. Sin entrar a valorar el valor de la información disponible, podemos hacer una primera clasificación en función de los contenidos:

- Información destinada a favorecer el consumo: hablamos de publicidad.
- Información destinada al ocio y tiempo libre: juegos, viajes, turismo, etc.
- Información cultural.
- Información socio-laboral.
- Información científica y técnica.
- Meta-Información: Material sobre como buscar mas información (portales).

Reinventar la rueda.

Ante este hecho, cuando un usuario necesita información tiene dos opciones: o intentar localizarla, o bien generarla por sí mismo. En cualquiera de los dos casos esto tiene un coste, que para una empresa se traduce en un coste económico. En el segundo caso, deberá dedicar una serie de tiempo y dinero en hacer que sus trabajadores elaboren dicha información. En el caso primero, dichos trabajadores deberán emplear el tiempo en localizar la información, y sobre todo en procesarla para que sea de utilidad a la empresa. Sea cual sea el proceso elegido, ello implica muchas veces el que la información tenga que ser varias veces asimilada y procesada. La consecuencia global es que en el mundo empresarial "la rueda se reinventa" constantemente.

En cualquiera de los dos casos una vez que la empresa tiene lo que necesita, se plantea la necesidad de dar una salida a dicha información. En función del modelo elegido obtenemos dos modalidades de empresa: Cerrada (el resultado de su trabajo es de uso exclusivo de la empresa) o Abierta (se comparte -de forma mas o menos aprovechable- el resultado del trabajo de la empresa). Cuando una empresa trabaja con el modelo abierto, decimos que adopta un modelo empresarial tipo "Software Libre".

Uno se puede plantear como un modelo basado en compartir la información -que ha costado dinero obtener y procesar- puede ser beneficioso económicamente. La respuesta reside en varios conceptos que analizaremos en los siguientes capítulos. Baste resumir algunos de ellos:

- Una empresa abierta no vende información: vende servicios.
- Los costes de elaboración son infinitamente menores que los de distribución y marketing.
- Existen multitud de fuentes alternativas de beneficios, aparte de los derivados de la venta de la información, que en el caso de una empresa abierta son -en el caso ideal- nulos.

4.4 El fenómeno del Software Libre.

El software libre es tan antiguo -o mas- que Internet. De hecho podemos decir que Internet no existiría sin el software libre. Desde que en los años 60 los Bell Laboratories cedieron el código fuente de su recién inventado Sistema Operativo UNIX, hasta la última versión del núcleo Linux, la historia del software se basa en el intercambio de información.

Una serie de características hacen en la industria del software la necesidad de una comunicación fluida y de un intercambio de información:

- La necesidad de formatos de intercambio de datos entre distintas aplicaciones.
- Los protocolos de comunicaciones entre sistemas, que exigen por un lado la existencia de un API, o modelo de programación estándar, y por otro lado una "implementación tipo" o modelo de referencia a quien seguir.
- Es un hecho que en Internet, los formatos propietarios son rechazados casi sistemáticamente en los entornos comerciales, debido a que ningún comerciante que se precie, y que tenga un mínimo deseo de pervivir, va a adoptar formatos que no sean manejables por todos sus clientes potenciales.

No obstante, no todo es color de rosa: intereses comerciales hacen que muchos fabricantes quieran imponer sus propios estándares, o añadan extensiones incompatibles a los ya existentes. Un fuerte rechazo a este modo de hacer empresa se observa en el software actual, debido -entre otras causas- a:

- El mundo del software abierto es capaz en poco tiempo de adaptar el código libre para soportar extensiones no estándar a los protocolos. Todo ello se traduce para la empresa que desarrolla dichos protocolos en una carrera por inventar nuevas extensiones, hasta que se llega a la total inutilidad de los esfuerzos realizados en forzar un estándar.
- El desarrollo de extensiones propietarias tiene además un riesgo tecnológico: la carencia de un modelo de referencia da lugar a la posibilidad de cometer errores, difíciles de detectar y corregir -dada la naturaleza "propietaria" del código-, así como la suspicacia del usuario ante "puertas ocultas" o fallos de seguridad. Los errores de codificación de implementaciones propietarias han sido la constante en muchos paquetes software de todos conocidos.
- El empresario que posee un mercado cautivo sobre los usuarios de dicho software muchas veces tiende a ignorar a los usuarios. Del mismo modo, muchos usuarios admiten los errores como algo consustancial al mundo del software. Esta realimentación negativa, a la larga redunda en una pérdida de eficiencia y de productividad del usuario, y a una falta de motivación para la mejora de la calidad en la empresa de software. Finalmente, se acaba produciendo una "desconexión" entre el cliente y su proveedor: el producto deja de responder a las expectativas del usuario, este deja de "realimentar" a su proveedor y

se acaba adoptando otro software similar que realmente responde a las expectativas. Mientras tanto, el proveedor y el usuario sufren pérdidas económicas por el uso de un producto y una metodología que no son capaces de satisfacer a todos los elementos que intervienen en la cadena.

- El mercado del software es un auténtico campo de batalla: podemos hablar de "nichos ecológicos", donde sólo cabe un competidor. No existe tiempo para desarrollar mas que nuevas prestaciones para un producto. Es mucho más efectivo, duradero y rentable, capturar clientes con un buen producto que con un API propietario. Además, esta ultima táctica desvía masa productiva a actividades no directamente remuneradoras, sino "de protección", que a la larga, y según hemos visto anteriormente, se revelan ineficaces. Sólo es efectiva esta estrategia en líneas de desarrollo a muy corto plazo, en las que no tiene sentido para la competencia desarrollar contramedidas.
- Recientes hechos y actuaciones judiciales ponen en tela de juicio el software propietario, en el sentido de que en algunos casos se deriva en situaciones de indefensión para el usuario final: bases de datos de usuarios, archivos ocultos, información confidencial acaba llegando al creador del producto, sin que el usuario pueda hacer nada para evitarlo.

Sería de necios cerrar los ojos a la realidad y afirmar que las empresas no consiguen imponer sus estándares. Lo cierto es que en gran parte de los casos los estándares siguen a las empresas que los lanzan. Lo que se pretende decir con esto es que sólo cuando es asumible el coste de la innovación puede plantearse la creación de protocolos o interfaces propietarios. El mercado tiene gran inercia y necesita apoyarse en unas bases conocidas, por lo que ninguna empresa media puede permitirse el lujo de forzar clientela en un mercado competitivo. La adopción de estándares conocidos -y aceptados- permite a los usuarios arriesgarse a probar el nuevo producto, y la existencia de modelos de implementación públicos permite a la empresa un desarrollo sin necesidad de partir "de cero". Incluso con gigantes de la industria software se ha dado lugar a problemas muy serios -incluso judiciales- por algo tan simple como la compatibilidad hacia atrás entre dos versiones del mismo paquete ofimático.

Orígenes del Software Libre.

En el anterior párrafo, se plantea pues, la necesidad de partir de un estándar aceptado por todos -facilidad para el usuario- y de un modelo de implementación -facilidad para la empresa-. Esta característica es esencial en el software libre: un sistema de producción que favorece a todas las partes implicadas. Veremos en capítulos posteriores los modelos matemáticos subyacentes a esta metodología. Entretanto, haremos una aproximación evolutiva a dicho modelo.

No podemos olvidar que el boom de la informática y de las tecnologías de la información es relativamente reciente: hace tres décadas el mundo del software estaba reducido al ámbito académico -y a veces militar- de unos pocos países privilegiados. El proyecto Arpanet fue en su origen un desarrollo de la universidad de Berkeley encargado por el Departamento de Defensa. Las leyes federales obligan a que, pasado un tiempo, todo proyecto financiado con fondos públicos, sea del dominio público.

En otros casos no fue la ley, sino los convenios entre empresas y organismos públicos los que potenciaron la distribución de software libre: si ATT no hubiera cedido el código fuente de su sistema operativo UNIX, posiblemente Internet -si existiera- sería totalmente distinta de lo que conocemos hoy en día.

Por último hay que destacar el software llevado a cabo mediante colaboraciones o convenios entre empresas. El sistema X-Window es mantenido y desarrollado por un consorcio de empresas que se comprometen a seguir unos estándares y normas de desarrollo comunes, obteniendo a cambio una garantía de interoperabilidad y de acceso a recursos comunes, especialmente el código fuente. (Se ha producido recientemente un intento de "privatizar" este club, estableciendo una "cuota de socio", intento que felizmente por motivos nuevamente económicos -como no podía ser de otra manera- ha sido abortado).

Estos intercambios de información han traído consigo las primeras preocupaciones de índole mercantil: la protección de los derechos de autor. Estaba claro que un sistema de producción basado en compartir información debía establecer una serie de garantías sobre la paternidad y los derechos de uso de dicho software.

Las primeras pruebas en este sentido se limitaban a garantizar que todo trabajo basado en código abierto debía llevar consigo los créditos del copyright del autor. No se establecían protocolos de protección de la integridad del trabajo, o de garantizar la publicidad de todo trabajo derivado. Igualmente, tampoco se imponían restricciones de índole comercial, o de distribución.

Está claro que esta situación no podía llegar muy lejos. De hecho históricamente se ha podido comprobar que la ausencia de reglamentación sobre la propiedad del código abierto ha redundado en un "cierre progresivo" de dicho software, creación de extensiones particulares, o incluso la negación de derechos de copyright cuando el código original era relegado a ser una mínima parte del nuevo desarrollo. Nadie puede olvidar que el API de Microsoft Windows fue un convenio de colaboración con Apple para el desarrollo del porting de una aplicación ofimática al entonces naciente entorno PC. El actual declive de los sistemas UNIX (salvo la honrosa excepción de Solaris) no puede ser sino el resultado de un exceso de actitudes de "abuso del cliente" tanto en relación precio/prestaciones como en falta de interoperatividad. Incluso hoy en día, los intentos de unificación (UNIX System V, Common Desktop Environment, etc.) están condenados al fracaso debido a la negativa de las partes a una colaboración efectiva.

Por consiguiente, la primera generación de software abierto murió de egoísmo... dando paso a los sistemas propietarios, que si bien eran productos de desarrollo cerrados, constituían un refugio para el usuario en cuanto a garantía de estabilidad y continuidad. Esta situación pronto degeneró en una situación de monopolio "de facto", y de abuso sobre el cliente.

Por supuesto, tenía que llegar la reacción: ante un producto que el usuario necesita para su productividad, y cuando dicho producto tiene que ser adquirido en condiciones abusivas para el vendedor, la reacción es obvia: hoy en día la piratería informática mueve el mas del 60% del mercado mundial de software. Como acertadamente postula Richard Stallman: "Cuando el modelo de producción obliga a que mas de la mitad de los consumidores actúe en la ilegalidad, es que algo falla en el sistema". De esta idea surge la segunda oleada del software libre, que tiene sus orígenes en la Free Software Foundation.

Lo que se plantea ahora es un nuevo punto de vista en el software abierto: ya no se intercambia información porque sí, sino que se establece un modelo empresarial, con derechos y deberes para ambas partes.

- La primera premisa es la garantía de la propiedad: al contrario de lo que muchos pregonan, este modelo de software libre no renuncia a la propiedad de la información, sino que la garantiza hasta extremos que alguno considera abusivos. No sólo el trabajo original es propiedad del autor, sino que la licencia de distribución obliga a que todo trabajo derivado del original retenga el copyright de este.
- La segunda premisa es mas sutil: garantiza la continuidad de el carácter abierto de la información. Esto que a primera vista parece loable esconde dos motivos ocultos y llenos de repercusiones económicas (que desarrollaremos en otros apartados). La primera es una garantía de que nadie podrá apropiarse del trabajo con fines particulares. La segunda, y más sutil, garantiza la distribución del trabajo hasta unos extremos que harían el paraíso de cualquier director de marketing de una empresa comercial.

En torno a este modelo de desarrollo, y a esta filosofía ha nacido lo que hoy constituyen la estrella de los sistemas abiertos: por un lado el núcleo del sistema operativo Linux, y por otro, el conjunto de utilidades GNU de la Free Software Foundation.

La realidad actual.

Pudiera creerse que esto es todo, y que este modelo acabará siendo el futuro del software abierto. Es opinión del autor que esto no es ni mucho menos cierto:

- La licencia de uso GNU impone restricciones muy serias al desarrollo comercial; restricciones que muchas empresas no están dispuestas a asumir. Por ello están surgiendo diversas soluciones mixtas.
- La irrupción de las empresas "comerciales" en el mercado del software libre está provocando una desviación de los objetivos de desarrollo desde el software en si hacia los servicios de valor añadido que se crean en torno al software libre. Sin perder su calidad de "apertura" (fuertemente protegida por su licencia), se desplaza el valor del software no por lo que es, sino por sus connotaciones y por las consecuencias de su uso.

Veremos en los próximos apartados como se aplican estas tendencias.

4.5 El coste del software.

Uno de los apartados que sin duda han contribuido al éxito del software libre es su bajo coste, tanto para el cliente como para la empresa:

Es un hecho que la legislación europea pone ciertas trabas a la productividad, especialmente en el capítulo de inversiones: la legislación USA permite al usuario final (empresa o particular) incluir como gasto necesario para producir rendimientos de trabajo, el resultante de las inversiones en adquisición de material: un escritor puede deducir las facturas de los bolígrafos que utiliza. En Europa, la única forma de conseguir la amortización de las inversiones materiales es a través de subvenciones a la mejora de la empresa, no mediante deducciones. Por ello, el empresario europeo mira con lupa toda inversión no directamente productiva. Esto, junto con la desproporcionada relación en el coste del software a cada lado del Atlántico, hace que el ritmo de actualización de equipos y de software en Europa sea mucho menor que en USA.

¿Quién quiere pagar por la herramienta?

Porque si el uso de una determinada aplicación está penalizado, el cliente tiene dos opciones: la primera es buscar soluciones alternativas. El hecho de que Europa sea un fuerte centro de empuje para el software libre no tiene sólo motivaciones sociales, como muchos autores sostienen cuando hablan del "sentido de lo social y lo público" en Europa; sino -y sobre todo- tiene connotaciones económicas: la independencia tecnológica, la mejora de la balanza comercial, etc. son argumentos básicos en un entorno de libre mercado. En un apartado posterior estudiaremos con detalle la política europea respecto al software libre.

El hecho de la piratería informática.

La segunda solución a la penalización de la inversión tecnológica es la más simple: la copia ilegal. Está demostrado que el cliente no copia (salvo comportamientos compulsivos) si el coste del producto original es sólo ligeramente superior al coste de copia. Como la situación de abuso de las empresas no sólo se limita al coste del producto, sino al pésimo servicio post venta, el cliente acaba concluyendo que la piratería no sólo es justa, sino necesaria, y un deber para su "salvación empresarial" (que me perdonen los creyentes católicos por este juego de palabras: no es mío -N. del A.-) A esto se junta el hecho de que la gran mayoría de las casas comerciales no hacen realmente una persecución seria del hecho de la piratería. Todo vale para crear clientes cautivos, que necesitaran cursos, documentación, soporte... y que acabarán pagando de una u otra forma.

Minimizar los costes de instalación.

Un último aspecto sobre el uso empresarial del software tanto libre como comercial. Muchas empresas no son conscientes de los costes a largo plazo del uso de un determinado paquete ofimático o de un sistema operativo. Es fácil tener una mentalidad a corto plazo, y buscar el minimizar los costes de instalación, confiando en la providencia a la hora de evaluar los costes de mantenimiento. La empresa que confía en el software libre debe ser consciente de que dicho software no es tan barato a medio plazo, a menos que este dispuesta a una inversión adicional en mantenimiento y seguimiento de dicho software (ver el apartado "las trampas del software libre").

5 Motivaciones del software libre.

5.1 Resumen del capítulo.

- Nadie regala nada.
- Un poco de psicología y filosofía.
- Motivaciones personales.
- El software libre como potenciador del ego.
- Un modelo "sectario".
- Nichos ecológicos.
- Competencia en el mundo del software libre.
- El software libre en entornos no empresariales.
- Perfil psicológico del voluntario.

5.2 A la búsqueda de un motivo para el software libre.

Vamos a hacer una parada para analizar desde un punto de vista ético y filosófico la realidad del software libre y sus implicaciones. A la hora de abordar este tema, he querido tener en mente una pregunta: ¿cuál es el motivo de la existencia de este modelo?. No quiero establecer juicios de valor ni comparaciones. Es tradicional en el mundo del software libre la existencia de consideraciones "éticas" y morales. En un terreno de competición constante, como es el mundo empresarial, palabras como solidaridad, altruismo, o trabajo desinteresado carecen de sentido. Por eso, cuando una empresa adopta el modelo del software libre es que hay razones económicas o estratégicas en ello.

Para esto, nada mejor que partir de algún modelo sociológico que pueda explicar el éxito del software libre, y especialmente su aplicación en el entorno comercial. Si bien, como hemos visto anteriormente, el movimiento del software libre nació como una defensa ante el acoso a los usuarios del software, y como proclama por la libertad de expresión y de información, no es menos cierto que estos últimos años se ha producido un cambio estratégico: la energía que mueve el software libre está poco a poco tomando matices económicos, de estrategia de empresas. Lo que hace unos años era una bonita aventura, ahora se ha convertido en modelo empresarial. No afirmo la bondad ni la maldad de este hecho, sólo lo constato, e intento mostrar una explicación a tal efecto. Hay, creo, una explicación a tal situación: en sociología, existe un modelo de comportamiento muy conocido y estudiado, que hasta hace unos años era una curiosidad matemática de la teoría de juegos, y que sólo tenía aplicación al estudio de determinados comportamientos animales. Estamos hablando de los modelos de cooperación egoísta, en los cuales los participantes colaboran entre sí para obtener un beneficio particular.

Diversos filósofos han intentado explicar dicho modelo en base a la naturaleza de los seres vivos. A continuación hago un breve resumen de dicha teoría, conocida como "ética de lo útil". Aunque no necesariamente comparto dicha teoría, creo que puede explicar adecuadamente la actitud del mundo empresarial ante el software libre.

5.3 La ética de lo útil.

Nadie regala nada. Todo en la vida tiene su precio y su coste. Cuando el individuo establece una relación con su entorno, inevitablemente está intercambiando con dicho entorno diversos aspectos de su existencia. Dicha interactuación está marcada y motivada por la búsqueda de un resultado.

Un poco de psicología y filosofía.

Podemos buscar explicaciones a este comportamiento desde la filosofía y la sociología, así como establecer símiles con comportamientos y actitudes de otras especies animales. Muchos autores sostienen que la razón de la existencia es el afán de sobrevivir a uno mismo, bien a través de la reproducción, bien a través de las obras, o bien mediante la sublimación del "yo" a un ente superior. Esta supervivencia se consigue a través de la interactuación con el entorno. Desde este punto de vista, el hombre es un ser social en cuanto necesita de los demás para realizarse a si mismo.

Por consiguiente, las actuaciones personales responden al deseo del individuo de su propia pervivencia. Este deseo no es exclusivo del hombre, sino que abarca a toda la escala evolutiva, desde el virus hasta la mayor empresa u organización imaginable. La moralidad de los actos queda definida no por su "bondad", sino por su "utilidad". La sociedad tiende -no sin razón- a rebelarse con esta moral utilitarista, pero sería de necios negar el hecho de que en el mundo empresarial este planteamiento, la obtención de beneficios y la perdurabilidad de la empresa, son los que guían sus actuaciones.

Incluso a nivel particular se puede aplicar el concepto de utilitarismo a la hora de establecer motivaciones. Afortunadamente el libre albedrío de la persona, y la existencia de múltiples valores en su modelo de actuación, hacen que sea más difícil -si no imposible- hablar de un utilitarismo en la moral individual. No obstante, como veremos, ese altruismo individual puede ser -y de hecho es utilizado- por empresas u otros individuos sin tantos condicionamientos éticos.

Desde la ética de lo útil, pues, se deduce que en la búsqueda de la supervivencia del individuo, o del colectivo, haya que obtener resultados. Estos resultados se miden en dos términos: beneficio y efectividad. Por beneficio entendemos las relación coste/prestación obtenida de la interrelación social. Por efectividad entendemos el grado de perdurabilidad y rendimiento a medio y largo plazo de una serie de interrelaciones. Una relación beneficiosa no tiene por qué ser efectiva, y al contrario, una relación efectiva tiene un beneficio próximo a la unidad. Esto no quiere decir que no se obtenga beneficio, sino que ambas partes obtienen recompensas equiparables entre si.

Esta moral de lo "útil" tiene pues, dos vertientes de desarrollo y evolución: aquellos modelos basados en la búsqueda de beneficio personal, y los basados en la búsqueda de beneficio mutuo. Ambos modelos buscan beneficio, pero desde un enfoque distinto. El primero tiende a ignorar al otro, mientras que el segundo colabora con el otro para obtener un beneficio común. No es difícil establecer paralelismos con modelos políticos: tenemos desde el modelo comunista en el que el individuo delega la responsabilidad de su supervivencia en un "ente" conocido como estado, hasta el modelo ultra nacionalista en que el individuo ignora cualquier realidad y objetivos que no sean los suyos.

Motivaciones personales.

Visto pues el origen de la actuación social de la persona, vamos a bajar al terreno del software libre, ¿cuál es la motivación que impulsa a una comunidad de individuos a colaborar desinteresadamente en un proyecto de desarrollo software? Existen multitud de razones para ello: el afán de superación, la necesidad de reconocimiento, la mejora de currículum, adquisición de conocimiento... Eric Raymond comenta en su famoso artículo "La catedral y el bazar" el hecho de que este afán del colaborador puede ser aprovechado -y de hecho lo es- de modo inmisericorde por parte del coordinador del proyecto, y por supuesto por su empresa.

El software libre como potenciador del ego.

Porque a poco que el coordinador tenga un poco de tacto social, la interrelación con el voluntario evoluciona hasta un grado tal que que podemos hablar de sumisión e identificación con la causa. El colaborador, observa que su trabajo es reconocido, que aparece su nombre en Internet, que su currículum empieza a crecer... incluso que puede ganar dinero con esa colaboración. El objetivo de búsqueda de la pervivencia personal se ve tan recompensado que se establecen hasta relaciones afectivas con el proyecto en que se colabora.

Un modelo "sectario".

Es difícil que el colaborador detecte esta situación: el está colaborando en una buena causa, comparte información, trabaja en grupo... objetivos loables y políticamente correctos; cumple a rajatabla con su escala de valores, con su deseo de colaborar, de ser útil. En aplicación estricta de la moral utilitarista está satisfaciendo su deseo de pervivencia de una manera total y gratificante. Esta identificación con el objetivo hace que la efectividad de las transacciones sociales en este modelo de desarrollo sea increíblemente alta. Podríamos hablar de un modelo de "secta" en el que el sectario da su vida por el "líder". No hablo en sentido figurado: conozco personalmente gente que ha dejado hasta de comer por conseguir tener enmarcado en su casa una contestación por correo electrónico de su ídolo Linus Torvalds.

Evidentemente no todo es así. Este tipo de relación de sumisión no dura en el tiempo, dejando paso a una relación, si no entre iguales, si entre dos partes que conocen y asumen su papel: el de producirse mutuo beneficio. En este punto el colaborador pasa a tener un papel activo en la relación y obtiene algo más que compensaciones morales. Podríamos hablar de una madurez en el voluntario del software libre, en la que comprende y asume su papel, lo encaja en su escala de valores, y deja de ser un espectador pasivo para convertirse en actor.

Tenemos finalmente un proyecto de software libre, con una empresa, un coordinador, varios responsables de área, muchos -idealmente toda la Internet- voluntarios, y por supuesto, clientes. Cada uno en su puesto, con sus funciones, objetivos y motivaciones. Ahora vamos a ver cómo se hace un proyecto empresarial de software libre.

5.4 Otros modelos sociológicos.

Como muchos lectores no dejarán de hacer notar, hay modelos alternativos a la moral utilitarista que pueden explicar ciertos aspectos del software libre que no son -salvo mediante una mitificación extrema del modelo- explicables mediante una teoría de lo "eficaz". Este es el caso de las organizaciones o movimientos no empresariales en torno al software libre. Ejemplos como la Free Software Foundation, o el proyecto Debian, aunque podría decirse que son el resultado de la auto-organización de intereses individuales, no es sencillo seguir dicha línea de pensamiento, hasta explicar su situación actual. Aquí la ganancia económica es una ventaja añadida, aunque no constituye el objetivo principal.

En este caso podemos hablar de "organizaciones de voluntariado informático", cuyos objetivos son el garantizar a la comunidad diversos valores como:

- La defensa del derecho de información.
- La independencia ante poderes económicos e intereses comerciales.
- La búsqueda de metodologías alternativas a los desarrollos tradicionales.

La auto organización es pues una consecuencia, no de los intereses particulares de los cooperantes, sino de la necesidad de presentar un frente común ante la reacción mas habitual en la economía de mercado: la búsqueda de beneficios en detrimento de la eficacia.

Está claro que estas organizaciones poseen un componente económico nada desdeñable: se necesitan fuentes de financiación para mantener una estructura organizativa mínima, para la asesoría legal, etc. Frecuentemente se crean al amparo de Fundaciones, centros de investigación, universidades, y se nutren de donaciones, subvenciones y -para qué negarlo- de contribuciones de las empresas que hacen negocio alrededor de sus proyectos. Este punto ilustra una característica importante de la empresa de software libre, que la diferencia de una fundación: mientras esta última trabaja en los componentes esenciales del software, la empresa suele dedicar su actividad en los servicios añadidos al software generado por la primera. El reparto de "nichos ecológicos" vuelve a aparecer.

[JAMC: añadir perfil psicológico del voluntario]

6 Una primera aproximación empresarial al software libre.

6.1 Resumen del capítulo.

- Producto versus servicio.
- Regalar software como estrategia de mercado.
- Haciendo negocios con el software libre.
- El papel del programador.
- Por qué algunas empresas apoyan al software libre.

La empresa, a la hora de acercarse al mundo del software libre se plantea un enfoque radicalmente distinto al que la comunidad tiene de dicho software. Cuando los condicionamientos económicos son el principal motor de actuación, es lógico pensar que conceptos como altruismo, colaboración, afán de compartir información, etc., se convierten en medios para conseguir beneficio económico, dejando de constituir un fin en si mismos.

Vamos a introducir una serie de conceptos, relacionados con la estrategia de la empresa que vive del software libre. Observaremos que en muchas ocasiones, el software en si no es sino una excusa para el verdadero negocio.

6.2 La empresa que vende software.

En primer lugar: el concepto de la venta de software de por si, deja de tener sentido, dando paso a otras alternativas. El software deja de ser un producto, sino que se convierte en un medio para vender otros productos. Hablamos de los servicios de valor añadido en torno al software. Este concepto -servicio versus producto- es una constante en todos los modelos empresariales que utilizan software libre.

Es una práctica muy común pues, sin llegar al uso del software libre el hecho de regalar el software, como estrategia de captación de clientes, de creación de usuarios cautivos, y sobre todo, de crear necesidades alrededor del software que se regala, como pueda ser documentación, cursos, mantenimiento, etc.

Desde este planteamiento, se nos ofrecen dos alternativas: la empresa que produce software libre, y la que utiliza el software libre para dar salida a otros productos. Cada uno de estos modelos tiene diferente metodología de trabajo, pero ambos viven del negocio que se genera alrededor de dicho software. Veremos en un apartado posterior cuál es el papel del programador en dicha empresa y su metodología de trabajo.

Porque cuando los defensores de los derechos del programador, y del software propietario dicen "¿y de qué vamos a vivir los programadores?" estamos olvidando un detalle esencial: hoy en día, casi todos los programadores trabajan al amparo de una empresa, y dada la competencia actual, el precio del software tiene que ser ridículamente bajo, o en muchos casos, nulo. El programador vive del sueldo de su empresa, y ésta vive casi siempre de productos y servicios desarrollados en torno al software, no del software en si. Incluso los grandes del software, como Microsoft, Corel, etc., obtienen la mayor parte de sus ingresos por medios colaterales, como son los cursos , la venta de documentación, la publicidad de sus portales de Internet, o bien de acuerdos con fabricantes de hardware. La piratería del software es incluso tolerada -cuando no potenciada- en aras de conseguir mercados cautivos o nuevos usuarios.

En éste entorno, sólo el software "a medida" puede ser considerado como un producto por si mismo, y -en opinión del autor- el único por el que se debe cobrar un precio mayor que el meramente simbólico.

Por todo ello no es de extrañar el apoyo de muchas empresas al software libre: puesto que el beneficio que se va a sacar del software es casi nulo, es preferible destinar esfuerzos a otros menesteres más beneficiosos. Este apoyo puede realizarse de múltiples maneras, bien colaborando con algún proyecto, directamente o mediante financiación, bien creando servicios de valor añadido entorno a un proyecto de software libre. Otras veces el método consiste en la utilización pura y dura del software libre.

A continuación mostraremos los detalles de funcionamiento y gestión de una empresa que utiliza -o produce- software libre.

7 La empresa basada en software libre.

7.1 Resumen del capítulo.

- Estrategias de desarrollo.
  - Esponsorización.
  - Apadrinamiento.
  - Captura de cerebros.
  - Servicios de valor añadido.
  - Que debe ser libre y que debe ser "de pago".
- Modelo de desarrollo.
  - La pirámide de desarrollo.
  - Estrategias de desarrollo.
  - Marketing y "venta" del software libre.
- Estructura y organización.
  - Gestión de recursos humanos.
  - Gestión de recursos materiales.
  - Gestión de servidores de información.

Una vez sentadas las bases, vamos a pasar a describir el funcionamiento de una empresa orientada al software libre. Aunque, como hemos visto esto puede referirse tanto a que produce software libre como a que basa su actividad en acciones desarrolladas alrededor del software libre, vamos a centrarnos sobre todo en el primer caso, pues la mayor parte de lo que se exponga será de aplicación a los dos modelos.

7.2 Modelo de desarrollo.

Cuando una empresa se dispone a trabajar en torno al software libre, debe ser consciente de las consecuencias, tanto a nivel organizativo, como de metodología de trabajo, que debe asumir y hacer suyas. La primera , y principal es la de adaptar el modelo de trabajo de sus programadores al modelo de software libre.

7.2.1 Estrategias de desarrollo.

Pues el software libre ofrece un modelo de desarrollo contrapuesto al del software tradicional: los conceptos de planificación de trabajos, reparto de tareas, integración, paso a producción, etc., difieren grandemente del modelo de una empresa cerrada, debido a multitud de factores:

- Puesto que el desarrollo es público, las tareas de depuración y publicación de versiones se deben efectuar en paralelo con las tareas de desarrollo.
- Se deben habilitar canales de comunicación con los desarrolladores, establecer una política de distribución de versiones, y un sistema de feed-back desde Internet a la empresa.
- Hay que estar abierto a la posibilidad de que en función de las demandas de Internet, el desarrollo original pueda sufrir cambios drásticos.

Todas las fases de desarrollo del software libre están descritas y documentadas en el artículo "la catedral y el bazar" de Eric S. Raymond, por lo que no nos vamos a detener en exceso aquí. Simplemente haremos algunas anotaciones:

El "bazar" es un sueño imposible...

Si bien el modelo bazar funciona aceptablemente bien en entornos no empresariales, una empresa no puede dejar al azar el desarrollo y evolución de su software. Deberá permitir flexibilidad, nuevas prestaciones, incluso reestructuración completa del desarrollo, pero siempre debe hacer que el proyecto esté dirigido a sus intereses. Por ello el modelo bazar "puro" no es aceptable, sino que el grupo de desarrolladores de la empresa debe asumir las funciones de un "supervisor de proyecto".

"Cambio programador por analista de sistemas".

Vamos a hacer un alto en este punto. Aunque casi todas los artículos existentes hablan sobre la libertad y aleatoriedad de los desarrollos de software libre, lo cierto es que siempre hay detrás una mente que es capaz de coordinar y dirigir el proyecto a buen puerto. En muchas ocasiones no es sino el autor del programa original quien asume la labor de "director espiritual". En otros casos, es un consorcio, o una "junta directiva" quien decide la evolución del software. Es difícil que un director de proyecto "automático", tipo CVS pueda "atraer" suficientes adeptos como para poder hablar de un modelo bazar puro y duro. Por contra, la figura del coordinador, acaba siendo fundamental, y en un modelo empresarial es condición imprescindible para que el proyecto llegue a buen puerto.

Desde este punto de vista, el papel de los programadores en la empresa pasa a ser el de coordinadores, creándose de hecho una "junta directiva" que se organiza el trabajo. Dicho trabajo ya no es el habitual de un grupo cerrado de desarrollo software, sino un trabajo orientado al nuevo modelo. Así tenemos las siguientes tareas:

- Integración.
- Comunicación.
- Control de versiones.
- Documentación.
- Servicios de valor añadido.
- Relaciones publicas y marketing.
- Sistemas de información.

Como se puede ver, las tareas de programación y depuración son delegadas a la actividad en la red, dejándose para el grupo de desarrollo los temas relacionados con la coordinación y distribución. En un apartado posterior describimos el perfil y los roles del grupo de desarrolladores de la empresa.

Canales de distribución.

Es evidente para el lector que esta organización depende fuertemente de los canales de comunicación de la empresa con los demás participantes de la cadena. Una empresa cuya conexión a Internet sea pobre, o que no sepa aprovechar los recursos de la red, mas vale que se dedique a otra cosa....

Porque el software libre vive por, para y de Internet. Es preciso garantizar que los usuarios y colaboradores están informados, que el servidor web y ftp funcionan. El responsable de comunicación deberá dedicarse casi por entero a garantizar que todo lo relacionado con el software llegue hasta el último servidor de correo existente. Es intolerable el menor fallo en la cadena de comunicación, pues la vida misma del proyecto depende de ella.

Del mismo modo, el responsable de marketing hará lo imposible por que el proyecto sea conocido: sabe manejar los portales y los canales de anuncio de noticias, mantendrá permanentemente actualizada la página web... como veremos posteriormente, dicha página puede convertirse en una de las principales fuentes de ingresos de la empresa.

Estrategias de captación de voluntarios.

Para que un proyecto de software libre llegue a buen puerto, hace falta otro componente fundamental: los usuarios y voluntarios para el desarrollo. Es necesario que la empresa llegue a tener un plantel de colaboradores suficiente para poder llevar a buen puerto el proyecto. Existen diversas estrategias de captación de voluntarios, pero se pueden resumir en breves frases:

- El producto ofertado debe ser útil, y cubrir las necesidades de los usuarios. Esto que parece obvio, es frecuentemente olvidado, y dada la fuerte "selección natural" que se lleva a cabo en Internet, a menos que sea algo necesario, rápidamente estará condenado al olvido.
- Se debe partir de algo suficientemente atractivo como para que aquellas personas interesadas tengan algo a lo que acogerse. No vale de nada un mensaje de "busco interesados en realizar XXXX", sino que hay que decir "He hecho XXXX, y me gustaría que alguien lo probara". Muy pocos proyectos de software libre nacen "de la nada", sino que siempre hay alguien que realiza el desarrollo inicial.
- Hay que utilizar sin compasión técnicas de habilidades sociales, relaciones públicas , etc., con tal de mantener el interés del colaborador. Es necesario crear una identificación del voluntario con el proyecto, o de lo contrario se acabara como una "beta 0.9" mas de las que tanto abundan en la red. En el capítulo dedicado al "dilema del preso" y posteriores, se analiza el concepto de "masa crítica" en el desarrollo del software libre.

Minimización de I+D: Un "Outsourcing" de facto.

La primera consecuencia de todo este proceso es que el papel del equipo de software de la empresa pasa a ser de coordinadores de proyecto. La figura del desarrollador se relega a Internet, y engloba las fases de desarrollo, ampliaciones y depuración. Podemos hablar de un Outsourcing del desarrollo software, en el sentido pleno de la palabra.

Internet como servicio técnico.

Otra consecuencia del modelo de desarrollo del software libre es que el concepto de asistencia al cliente y servicio técnico queda también desplazada a la red. Una empresa inteligente hará uso de los recursos de la red para aprovechar y dirigir dicho empeño hacia su empresa: creara listas de correo, tendrá un programador dedicado a moderar dichas listas, pondrá las FAQ, HOWTOS, e instrucciones en su web, y hará lo posible por que dicha información sea distribuida de la forma mas eficiente posible. El ideal debe ser: "ninguna duda resuelta sin que aparezca el URL de la empresa". El hacer que el portal de Internet de la empresa sea referencia obligada para un proyecto de software libre es una fuente ingente de ingresos, tanto de publicidad directa en el portal, como de venta de servicios de valor añadido por parte de la empresa. El objetivo es conseguir una identificación entre el producto y la empresa, aunque dicho producto sea 100% software libre. Veremos en un apartado posterior diversas técnicas para conseguir este objetivo.

7.2.2 La pirámide de desarrollo.

En este modelo, y tal como vamos describiendo, se perfilan claramente una serie de escalas de actuación, con tareas y objetivos concretos. A continuación se describe el perfil de alguna de las partes.

La empresa.

Al margen de los proyectos realizados en torno a fundaciones, universidades, o grupos que podríamos calificar de no-empresariales, la empresa de software libre asume el papel del responsable máximo del proyecto. Dicha responsabilidad no es tanto de "paternidad" o de "líder del proyecto" sino como "padrino". En efecto, la empresa provee de estabilidad al proyecto, proporciona canales fiables de venta y distribución, mantiene la disponibilidad de ftp, web y correo en torno al producto.... Como vimos en el primer capítulo, si el desarrollo es correcto, el nombre del producto queda irremisiblemente ligado al nombre de la empresa. En muchas ocasiones, los desarrolladores son programadores contratados por la empresa, en otras la empresa subvenciona el desarrollo, proporciona cobertura legal, etc.

El coordinador.

El coordinador tiene un papel especial: podemos decir que ejerce de "Dios". Es difícil que un voluntario trabaje para una empresa, pero es muy sencillo hacer que trabaje para su líder. En el mas puro estilo de la sociología de las sectas sobre el coordinador recaen toda responsabilidad de mantener el proyecto vivo, de animar a los voluntarios, de proporcionar nuevas y nuevas versiones y prestaciones a una clientela ávida de noticias frescas sobre su software favorito.

La personalidad del coordinador es pues especial: en una fase inicial deben ser buenos programadores, pero una vez el proyecto alcance masa crítica, su papel pasa a ser el de relaciones públicas y de analista de sistemas. La mayor parte de los líderes de proyecto son buenos oradores, saben atraer la atención del público, responden personalmente al correo electrónico... posiblemente les quede poco tiempo real para desarrollar, pero también un buen líder sabe proveerse de colaboradores -a menudo compañeros de trabajo de la empresa- que asumen las tareas pesadas. Es tan importante este papel, que muchas veces buenos proyectos caen en el olvido por falta de un líder; o lo que es peor para la empresa: el proyecto es "robado" por otra empresa con mejores recursos. Como veremos posteriormente, el apadrinamiento de proyectos y el contacto de la empresa con el líder es fundamental, por lo que en general la mayor parte de los líderes de proyecto acaban trabajando para una empresa que vive de dicho proyecto: Redhat, Sendmail, PostGreSQL, etc...

Los colaboradores.

En organizaciones no empresariales de software libre, frecuentemente el papel del líder está diluido en el "comité organizador". Incluso en estos casos el liderazgo está repartido en áreas de trabajo, aunque la "atracción" de un proyecto sin líder es menor. Salvo honrosas excepciones -Debian, por ejemplo- es difícil, y no está exento de problemas un desarrollo "democrático".

El papel de los colaboradores es por tanto el de responsables de área. Aquí sí se exige un fuerte nivel técnico y capacidad de abstracción. Son los colaboradores los que van a realizar el trabajo de integración, los que van a recopilar la información que les llegue de Internet... en unión con el coordinador -y atendiendo a la comunidad Internet- deciden la política a seguir.

Esta organización recuerda grandemente al modelo bazar. No es del todo cierto, pues el modelo bazar exige una planificación previa, y frecuentemente estos grupos trabajan sobre la marcha. Podemos hablar de este grupo como el destinatario final de la opinión de Internet... y de la empresa o fundación que tengan detrás.

Al ser un grupo relativamente pequeño, la comunicación y toma de decisiones responde frecuentemente a un modelo "comité". Usualmente trabajan todos en la misma empresa, aunque en ocasiones, la facilidad de comunicación que la red ofrece proporciona un modelo de "comité distribuido" al grupo de colaboradores.

Los voluntarios.

El voluntariado constituye la fuerza de choque de un proyecto de desarrollo software. El flujo de comunicación entre el grupo de colaboradores y los voluntarios está profusamente documentado en otros ensayos, por lo que no nos vamos a detener aquí en detallarlo. Únicamente hacer hincapié en algunos detalles.

- El primero es la necesidad de que el voluntario este permanentemente informado del desarrollo del proyecto. Es misión del responsable de comunicación dicha tarea. Es necesario que el voluntario no se quede descolgado y que pueda sentirse copartícipe del desarrollo.
- El segundo es la necesidad que tiene de realimentación. Por experiencia personal, conozco como un proyecto se echa a perder por no atender las opiniones, quejas y sugerencias de los usuarios. Cierto que en última instancia los intereses de la empresa en el proyecto van a tomar la parte principal en el proceso de toma de decisiones, pero también es cierto que muchas veces la opinión de la mayoría de los usuarios suele en su conjunto ser más objetiva que la visión que la empresa tenga del proyecto. No se debe olvidar nunca que estamos trabajando con software libre, y que en el momento que dicho software no responda a las expectativas del usuario, el proyecto quedará abandonado.
- En este ensayo establecemos una diferencia entre voluntario y usuario. El voluntario participa en el desarrollo. El usuario utiliza el desarrollo. No conviene perder de vista esta diferencia.

Los usuarios.

Porque el interés comercial del producto de software libre de por si, reside en la comunidad de usuarios y clientes potenciales del producto. Está claro que el voluntario se basta a si mismo, o utiliza los recursos de la red, para hacer que el software responda a sus necesidades. Por contra, el usuario suele utilizar los paquetes binarios, casi nunca compila el código fuente, y frecuentemente no utiliza sino el manual del usuario (y no siempre). Frecuentemente es el usuario quien hace los comentarios mas oportunos acerca de la apariencia, el modo de funcionamiento, las funcionalidades a añadir... y los errores evidentes del software. Mientras el voluntario busca eficiencia, el usuario busca funcionalidad. Un buen responsable de marketing, conseguirá que el usuario se convierta en cliente: comprando documentación, recibiendo cursos, e incluso mediante soluciones pre-instaladas.

Esta estrategia tiene además una ventaja oculta: si se consigue "enganchar" al usuario como cliente, este queda integrado en la cadena de producción, convirtiéndose de hecho en el departamento de control de calidad del producto.

La competencia.

Finalmente nos queda una última parte en la cadena del desarrollo software: la competencia.

No debemos olvidar un hecho importante en el mundo del software: los conceptos de nichos ecológicos y de evolución , de tanta aplicación en biología, siguen siendo válidos al ser aplicados al software. En un mundo tan altamente competitivo, podemos decir que cada necesidad software tiende a tener un único producto que la cumbre. En los casos en que esto no es así, y hay varios productos compitiendo, se produce el fenómeno de "evolucionar para permanecer en el nicho", esto es, una carrera desenfrenada por añadir prestaciones y funcionalidades para no perder cuota de mercado. Tenemos ejemplos de dicha evolución en el entorno del software libre: las "guerras de los escritorios" en Linux o la búsqueda de un entorno ofimático libre...

La empresa pues, ante la competencia, deberá buscar, o bien desbancar a sus competidores, o bien desplazarse hacia un nuevo nicho, donde no haya competencia. Una tercera alternativa es la búsqueda de "pactos" de interoperatividad, pero normalmente no es una elección atractiva desde el punto de vista económico, pues implica un reparto de beneficios, y a la larga una pérdida de competitividad de la empresa menos "ágil".

7.2.3 Marketing y "venta" del software abierto.

Hemos llegado al núcleo del problema: ya tenemos una empresa organizada en torno al software libre, y con un producto. Ahora toca venderlo y ganar dinero. Pues, ¿cómo se puede vender un producto que es "gratis"?.

- Una primera aproximación consiste en ahorrarle al usuario trabajo, a cambio de un mínimo coste. El concepto de "distribución de paquetes" sigue este modelo. Un ejemplo concreto es el de las distribuciones binarias de Linux: si bien el usuario tiene en todo momento opción a crearse su propia distribución, el coste en horas, o en gasto de grabación del CD-Rom, le hacen que este dispuesto a pagar una cantidad simbólica por tener el trabajo ya hecho.
- La segunda alternativa es la venta de funcionalidades añadida: la licencia GNU permite enlazar aplicaciones libres con aplicaciones comerciales, siempre que la segunda no utilice otros recursos de la primera que no correspondan al API. Por ello es frecuente la venta de plug-ins, así como de productos que desarrollan aplicaciones específicas alrededor de un software libre (manejadores gráficos, gestores de administración, front-ends, etc.).
- Una tercera vía es la venta de documentación. Uno de los grandes problemas del software libre es la ausencia de una documentación fiable y -por la propia naturaleza del software libre- actualizada. Actualmente, el modelo de "documentación aparte" es adoptado por muchas empresas que trabajan con software libre.
- Por último, nos queda -por lo menos- otra opción: la de dar cursillos y entrenamiento a los usuarios. Hace poco saltó la noticia de que RedHat ofrecía unos cursos en los que se daba un diploma que acreditaba como "RedHat certiffied manager".... el paralelismo con otras empresas comerciales es evidente.
- Como efecto marginal, pero no desdeñable, es preciso destacar el papel que los responsables del proyecto tienen: conferencias, charlas exposiciones, etc., son también fuentes de ingreso para la empresa.

Publicidad y distribución.

Para que todos estos métodos lleguen a ser rentables hace falta algo común a todas las empresas, tanto libres como cerradas: el producto tiene que ser conocido, y la gente debe ser convencida de que su posesión es una necesidad vital. El responsable de marketing es el encargado de esta misión. Existen diversas estrategias:

- El uso -y en ocasiones abuso- de Internet debe se una constante. Un aprovechamiento inteligente de la red hará que el software sea conocido por los clientes potenciales en cuestión de horas.
- No tenemos por que estar limitados a nuestro servidor. El convencer a los voluntarios de que mantengan réplicas (mirrors) de nuestro servidor, hará que las posibilidades de que nuestro software sea conocido crezcan exponencialmente.
- Del mismo modo es necesario conseguir que los portales de Internet reflejen nuestro producto. Es más, debemos conseguir que nuestro web se convierta en un portal.

- Este ultimo punto hace que aparezca una nueva fuente de ingresos: los ingresos por publicidad de otras empresas. Actualmente la mayor parte de los ingresos de Netscape se deben a la publicidad de su portal.
- Hay que saber vender. Hoy en día, el software libre está "de moda", y como veremos en otro capítulo, el uso de dicho software es "políticamente correcto". Una adecuada venta del hecho de que la empresa distribuya software libre, abre la puerta a instituciones, organismos públicos, universidades, etc., donde los condicionantes de la decisión de adquisición del software no son económicos sino políticos.
- Por último, hay que ser capaz de dar a los posibles clientes tareas de consultoría. El cliente no quiere un producto, sino una solución a su problema, y si no se produce abuso, normalmente no pondrá impedimento al pago de una determinada cantidad por lo que a todas luces es una mínima adaptación -a veces ni siquiera llega a tal- de un producto de software libre.

7.3 Estrategias de comienzo.

Todo lo dicho hasta ahora parte de la base de que la empresa tiene un producto de software disponible. Como es obvio, esto no siempre ocurre, sino que en ocasiones hay que proceder , o bien a un trabajo inicial de desarrollo, o a la liberalización de un software anterior, o incluso a la "captura" de un proyecto. Veamos en detalle estas técnicas.

Esponsorización o apadrinamiento.

El primer método es el más sencillo -y en cierto modo el menos arriesgado-. Consiste en que la empresa invierte dinero en una fundación, o asociación, o incluso en un proyecto de investigación de una universidad, con el fin de financiar económicamente un determinado proyecto.

Claramente, una empresa que utilice esta estrategia no suele estar interesada en el proyecto en si, sino en derivar parte de su carga de trabajo a otros intereses más rentables. Podemos hablar de optimización de recursos de la empresa, mediante técnicas de "Outsourcing" camuflado.

En otro caso la inversión económica responde a otras necesidades o intereses. Un efecto curioso que se da en algunas empresas de software libre, (caso de RedHat, o Netscape) es que son consideradas como valores de bolsa, sujetos a cotización en el mercado. Antes de la Crisis de Netscape, la mayor fuente de ingresos que poseía dicha empresa se debía a sus operaciones en bolsa. En el caso de Redhat, las inversiones y subvenciones realizadas por grandes compañías de hardware y software, hacen que su valor en bolsa se dispare, llegando a tener una cartera de beneficios más que apreciable.

Captura de cerebros.

Un segundo método consiste en que la empresa contrate a los coordinadores de un proyecto de software libre, asumiendo dicha empresa los fines y objetivos de dicho proyecto... en su propio beneficio. Los nuevos empleados disfrutan de casi total libertad para seguir el desarrollo, con la garantía de que van a cobrar por su trabajo. Muchos de los nombres famosos en el mundo Linux trabajan al amparo de empresas como RedHat o Netscape.

Liberalización de software.

En ocasiones, la competencia hace que deje de ser rentable el mantenimiento y actualización de un producto. La liberalización del código fuente, proporciona una segunda oportunidad al producto, y una nueva fuente de desarrollo.

Hay que hacer constar que esta política de liberalización, rara vez se hace a través de una licencia tipo GPL: es norma casi general que el fabricante desee proteger lo más posible su inversión inicial, y por ello se aplican cláusulas de restricción de la distribución. Las restricciones más usuales son:

- Restricción a la distribución: se suele prohibir el uso comercial del código fuente liberado.
- Restricción al formato: los parches y añadidos deben ir aparte de la distribución oficial.
- Cláusula de terminación: la empresa puede restringir sin previo aviso el uso del software liberado, tanto en fuentes como en ejecutables.

La comunidad Internet está generalmente en contra de estas restricciones especialmente de la última, que ha sido adoptada por empresas como IBM y Apple. Claramente no son sino un intento de utilizar la fuerza de producción de Internet en beneficio exclusivo de la empresa.

Captura de proyectos.

Nos queda aún otro modelo de introducción empresarial en el modelo de software libre. Tenemos un ejemplo clásico en el desarrollo del paquete PostGreSQL (un gestor de bases de datos relacional cuyo copyright ha sido recientemente adquirido por InSight Distributions).

En este caso, la empresa asume poco a poco la coordinación de un proyecto, hasta el punto en que los responsables originales del trabajo "ceden" las labores de mantenimiento del programa. Esto permite a una empresa integrar sus productos de pago en torno a un programa, que a pesar de ser libre, es mantenido y dirigido por la empresa. Aunque el carácter abierto de dicho software no se pueda perder, dadas las características de la licencia, su orientación futura dependerá en gran medida de los intereses de su nuevo patrocinador.

Qué debe ser abierto y qué debe ser "de pago".

Con independencia de la estrategia adoptada, la empresa tiene que ser consciente de que un producto de software libre no basta de por si para la productividad. Es más, como hemos visto existen campos del desarrollo software donde el modelo de software libre no puede ser viable, debido a que no cumple la condición de masa crítica requerida para dicho modelo (ver siguiente capítulo). Estamos hablando de soluciones "a medida", o de programas muy especializados. la empresa utilizará los resultados del desarrollo de S.L. para proporcionar una base de lanzamiento de dichos productos.

Los contratos de outsourcing son pues un modelo idóneo de explotación del software libre. Al cliente no le importa tanto la condición de "libertad" del código, cuanto que le proporcione solución a su problema. Es aquí donde la empresa adquiere sus beneficios directamente del software, y donde el resultado del esfuerzo invertido en el software libre es directamente aprovechado por la empresa, con independencia de otras soluciones marginales.

7.4 Estructura y organización.

Todo lo dicho hasta ahora se refiere a procedimientos de trabajo en la empresa. Analicemos un poco la organización y gestión. Existe mucha literatura acerca de cómo hacer una empresa "orientada a Internet", y no vamos a profundizar en ello, remitiendo al lector a la bibliografía reseñada en el apéndice. Haremos en cambio hincapié en la gestión y control de los recursos.

Gestión de recursos materiales.

Lo primero que caracteriza a una empresa de software libre es la escasez de recursos, tanto materiales como humanos. Esto es explicable si consideramos que dicha empresa delega en la red tareas básicas como los procesos de producción y distribución. La empresa de software libre posee generalmente un inmovilizado (recursos materiales) mínimo, y por contra dispone de una fuerte capitalización, bien mediante inversiones, financiación externa, etc. La mayor parte de las empresas actuales son objeto de especulación en bolsa, y algunas de ellas -caso de RedHat- son consideradas hoy en día como "opciones de riesgo" por los inversores bursátiles.

Las áreas de actuación y de inversión en recursos materiales de esta empresa se derivan hacia los campos de obtención de beneficios: se prioriza pues el departamento de Marketing, y el de atención al cliente. Se potencia el uso de la red, la publicidad vía Internet, y los mecanismos automatizados de respuesta al usuario.

Otra área de inversión constituye el desarrollo de servicios de valor añadido: documentación, soluciones al cliente, productos auxiliares, etc.

El perfil pues es el de una empresa de servicios en su acepción más extrema: el de una empresa que vive de, por y para Internet.

Gestión de recursos humanos.

Visto el perfil personal y laboral de los responsables de un proyecto de software libre, es necesario concluir que la política de trabajo de nuestra empresa, debe diferir bastante de la de una empresa habitual. Se deberán poder aplicar técnicas de teletrabajo, permitir una muy grande libertad horaria, etc. Recordemos que en el software libre, aparte de la motivación económica existe una muy fuerte motivación personal, y a ella tampoco son ajenos los responsables del proyecto. Al coordinador de recursos humanos le corresponde encauzar las energías, y conducir la nave a buen puerto.

De hecho muchos proyectos de software libre se han venido abajo por una falta de organización y coordinación en la cúpula. En el mundo empresarial esto sólo es tolerable por la competencia... para apropiarse del proyecto.

Se impone un control.

Por ello el responsable de recursos humanos deberá ser capaz de:

- Mantener el ritmo de trabajo y los plazos.
- Monitorizar el uso de la red, para evitar abusos.
- No olvidar las actividades relacionadas con el marketing que afecten a los desarrolladores: ferias, conferencias, cursos, etc.
- Decidir que tareas debe asumir el grupo de desarrolladores, y que tareas se deben delegar en la red.

El departamento de marketing y atención al cliente, deberá estar a su vez familiarizado con el "Estado del arte" del desarrollo, atender a las consultas de clientes y colaboradores.

Gestión inteligente de servidores de información.

Como hemos visto, todas las tareas de trabajo y control, giran en torno a un componente principal: la red. Gracias a la red, la empresa consigue voluntarios, clientes, publicidad.... en suma, dinero.

Podemos observar que gran parte de las páginas web de proyectos de software libre abundan en publicidad. Es una publicidad inteligente, personalizada en función del visitante, que no sobrecarga el sistema, compatible con todo tipo de navegadores y entornos... la idea es conseguir ingresos, sin perder visitantes, (esto, que parece obvio es olvidado por empresas "serias").

El servidor de información deberá disponer de listas de correo automáticas, adecuadas, y a ser posible moderadas por un responsable. Se deberá cuidar el mail-spamming, la corrección en el estilo. De ser posible, sólo una voz hablará de la empresa a través del correo. Las listas de correo deberán estar accesibles por Web y FTP.

Del mismo modo, toda la documentación libre deberá ser accesible de manera sencilla y directa. Las técnicas de "registro" previo deberían utilizarse sólo para los "clientes", y nunca para el acceso general. Se deberá dar cuenta en el menor plazo posible de toda novedad existente. Es vital que los servidores de información tengan un buen enlace con la red, y que se ejecuten en entornos seguros y fiables, y -por supuesto- basados en software libre.

Una cosa a evitar es el uso negativo de las estadísticas: aunque quedan muy "bonitos", debería huirse de contadores, y sobre todo de cualquier cosa que pueda ahuyentar a un posible voluntario o cliente.

Mensajes del estilo "Bienvenido Mr. Pepito. Es la quinta vez que se conecta a nuestro web" producen reacción de rechazo en el cliente".

Es de agradecer que se establezca una metodología de trabajo: habrá que definir unas normas de estilo, un sistema tipo CVS para la gestión del software, etc... son normas básicas de coordinación de proyectos, que adquieren importancia fundamental cuando se va a coordinar un sistema de esta envergadura.

8 Un modelo matemático: El dilema del preso.

8.1 Resumen del capítulo.

- Descripción.
- Estrategias del juego.
- Resultados experimentales.
- Exportando el modelo a la vida real.

8.2 Descripción.

En el campo de la teoría de juegos existe un clásico de la literatura, conocido por sus implicaciones y aplicación a las relaciones sociales: el "Dilema del Preso".

En su versión básica plantea un problema simple de toma de decisiones:

Dos atracadores han sido detenidos, estando cada uno aislado del otro en sendas salas de interrogatorio. El fiscal propone a cada uno de los presos un pacto: si delata a su compañero se librará de la cárcel.... siempre y cuando el compañero no le delate a él. Si ambos optan por permanecer callados la pena será de un año de prisión. En el caso de condena, la pena será de tres años.

En el modelo sencillo tenemos las siguientes combinaciones:

```
Preso A         Preso B         Resultado A     Resultado B
-----------------------------------------------------------
calla           calla           1 año           1 año
calla           delata          3 años          libertad
delata          calla           libertad        3 años
delata          delata          3 años          3 años
```

Tanto A como B razonan que en el caso de permanecer callados no tienen posibilidad de librarse de la cárcel, luego la solución ideal es delatar al compañero... salvo que el compañero puede seguir el mismo razonamiento, ¿cúal es pues la decisión correcta?.

La segunda variante del dilema del preso nos introduce de lleno en los modelos de cooperación:

Tenemos de nuevo dos atracadores, esta vez en libertad, que se dedican a la compraventa de productos robados. Los atracadores han pactado que uno de ellos deje la mercancía en un lugar determinado, mientras que el otro deja el dinero en otro lugar. Ambos atracadores se dirigen a recoger posteriormente el paquete destinado a cada uno...

Está claro que si uno de ellos deja un paquete vacío, y recoge a cambio un paquete lleno, ha obtenido un beneficio neto en la transacción. Por consiguiente ambos dejan sendos paquetes vacíos... y recogen paquetes vacíos. Pero, ¿y si en lugar de un único intercambio se realizan una serie de intercambios espaciados en el tiempo?. Si un ladrón traiciona al otro, es casi seguro que será traicionado en la siguiente transacción. ¿Cuál es la estrategia ganadora?.

Para complicar más las cosas, vamos a suponer que en lugar de dos atracadores tenemos n atracadores, interaccionando entre ellos por parejas: a cada turno el atracador m debe decidir qué es lo que va a hacer con cada uno de los demás atracadores, colaborar o traicionar.

Vamos a modelar el juego: Tenemos n agentes y un árbitro. A cada turno el árbitro indica a cada agente con que otro agente va a interactuar, debiendo decidir si entrega como resultado un 1 (colabora) o un 0 (traiciona) Gana el agente que al cabo de X turnos consigue la mejor relación entre colaboraciones e inversiones.

La analogía con el modelo empresarial es evidente: el árbitro es el mercado, y los ladrones son todos aquellos que intervienen en transacciones comerciales (habrá alguien que opine que la analogía debería ser a la inversa, pero eso ya son temas políticos...).

Definimos *beneficio* en un turno de transacciones como:

```
( 1 / numero_jugadores ) * ( total obtenido - total invertido )
```

donde el resultado está en el intervalo (-1,1).

El beneficio total será el resultado del sumatorio de los beneficios parciales.

Aparentemente, el jugador que escoja una estrategia egoísta tiene las de ganar, pues su beneficio nunca será negativo. Del mismo modo, los colaboradores sistemáticos tienen las de perder, por pardillos....

Por ello, vamos a definir un nuevo concepto: el de *efectividad*, entendida como el cociente entre el total obtenido y el número de jugadores. La efectividad total, será la media aritmética de las efectividades parciales. Experimentalmente, se observa que el algoritmo "egoísta" tiende rápidamente a una efectividad nula conforme el número de transacciones crece.

Cuando se simula mediante ordenador este juego se obtiene un resultado sorprendente: la mayor efectividad se corresponde con un beneficio nulo... que no es sino el resultado de una igualdad entre el coste y el resultado. El algoritmo que cumple con este requisito no es sino el conocido "ojo por ojo y diente por diente" de la tradición judeocristiana. Codificarlo es muy simple:

- En la primera jugada el agente colabora (devuelve un 1).
- En las siguientes jugadas se limita a repetir el movimiento anterior de su oponente.

Las implicaciones son sorprendentes, aunque un mínimo de sentido común las explica: en un entorno competitivo, se tiende a potenciar aquellas operaciones que dan un resultado positivo. Del mismo modo se aisla a aquellos de quienes no se espera resultado alguno. Al aumentar el número de operaciones el "universo" se divide rápidamente en dos modelos: los aislacionistas y los colaboracionistas. En el primer grupo, el beneficio a corto plazo es elevado, pero su eficacia disminuye progresivamente con el tiempo. En el segundo modelo, si bien los beneficios suelen ser menores (o incluso nulos) la efectividad a largo plazo se incrementa debido a que toda inversión acaba resultando en un beneficio.

El que el beneficio neto sea nulo tiene una explicación obvia: en una relación proveedor-cliente, ambos tienen que ganar. Si se produce un desequilibrio hacia uno u otro lado, la relación tiende a deteriorarse rápidamente. El "abusar del cliente" sólo puede ser beneficioso a corto plazo o bien si se dispone de un número suficientemente grande de clientes. Como dice el refrán: "A algunas personas se las puede engañar siempre, y a veces es posible engañar a todo el mundo. Pero no se puede engañar siempre a todo el mundo".

La experimentación -y el contraste con la realidad- han demostrado que el "ojo por ojo" no es el procedimiento más eficaz. Diversos experimentos con algoritmos genéticos modelados para jugar al "Dilema del preso", muestran que existe un modelo, que si bien tiene una efectividad similar, aumenta el beneficio neto. Es el denominado "machaca pardillos". En esencia es un "ojo por ojo" modificado, de tal suerte que tras un número aleatorio de secuencias iguales, el agente cambia durante un ciclo el resultado de su salida, volviendo inmediatamente al "ojo por ojo" en la siguiente jugada.

El por qué de dicho resultado se descubre enseguida: el principal problema del "ojo por ojo" es que es un sistema "con memoria", de tal suerte que ante un oponente que comete un desliz, acaba fácilmente ensartado en una secuencia de ceros, con la consiguiente perdida de efectividad. Por otro lado adolece del defecto de ser "demasiado generoso" con aquellos algoritmos que son colaboradores por sistema. El hecho de insertar un 1 en una secuencia de ceros, puede "reconvertir" a un traidor, y de la misma forma, insertar un 0 en una secuencia de unos, puede hacerle aprovecharse de los pardillos colaboradores sistemáticos.

La similitud con el modelo de desarrollo de software abierto es evidente: es un modelo que está basado en la colaboración mutua, que realimenta rápidamente a los agentes que colaboran, que se aprovecha de los pardillos de manera inmisericorde... y que relega rápidamente al olvido a los no colaboradores.

La analogía es incompleta: el problema del dilema del preso adjudica un coste binario a cada transacción (0 ó 1). Esto, como hemos visto, no es exactamente cierto: el coste real es inversamente proporcional al número de agentes que intervienen en cada turno de transacciones: piénsese que desde el lado de la empresa se está compartiendo el coste de poner el código al público entre n potenciales proveedores de soluciones o mejoras, y que al aumentar el número de usuarios, el beneficio potencial se dispara. No obstante nos sirve para demostrar que un sistema que potencie las transacciones con el mayor número posible de agentes, consigue la mayor efectividad conforme el número de intercambios aumenta.

En las referencias y apéndices el lector encontrará direcciones de Internet donde se trata el tema, así como diverso software de emulación del juego del dilema del preso. Es interesante observar la evolución de cada partida en función del número de participantes, del número de turnos, e incluso del "perfil psicológico" de los jugadores...

Una reflexión: este modelo se denomina de colaboración egoísta. El objetivo es obtener beneficios y aumentar la eficacia a medio y largo plazo. La colaboración no sólo no es desinteresada, sino que el voluntarismo está penalizado. La idea de la participación altruista es uno de los mitos del software abierto, pero de ningún modo se corresponde con la realidad, especialmente en el mundo empresarial.

8.3 El dilema del preso y el software libre.

Como muchos lectores conocedores del tema objetarán, el modelo del dilema del preso está estudiado para el caso de n participantes compitiendo entre si dos a dos. Esta no es una aproximación realista, pues no tiene en cuenta que en el mundo real las interacciones se producen entre todos los participantes de manera simultánea. Las alianzas y pactos entre varios participantes, por citar un ejemplo, no están contempladas, ni siquiera estudiadas. Para que el modelo de cooperación sea óptimo en este entorno deben darse unas condiciones de contorno que permitan reducir el problema al caso conocido.

- El primer concepto es el de "masa crítica": para que el modelo sea efectivo, el número de participantes debe ser grande, varias magnitudes superior al de un desarrollo software normal. De lo contrario, no tenemos sino un modelo de trabajo distribuido.
- El segundo concepto es el de "temporalidad": en un proyecto a corto plazo, el modelo de software libre no es capaz de proporcionar suficientes beneficios. Estudiando estadísticamente partidas del dilema del preso se ve que las curvas inversión/beneficios del modelo cooperativo son asintóticas respecto al tiempo, según la gráfica:

```
|                                                     --
|                                             =====--
|                                        =====   --
|                                   =====          --
```

```
|                             =====         --
|                        =====            --
|         Inversion =====               --
|              =====                  --
|         =====                     --
|    =====                       ---
|=====                        ---
|                          --- Beneficios
|                      ----
|                  ----
|             -----
|       ------
|--------
|
|------------------------------------------------------------> tiempo
```

Por ello la empresa que adopte el modelo de software libre debe ser capaz de asumir un periodo de pérdidas iniciales, hasta conseguir su aceptación.

- Otro concepto es el de "Identificación". Los participantes deben conocer y asumir su papel en el modelo, y lo que es fundamental TODOS los participantes deben obtener beneficios de la cooperación mutua. De lo contrario, se está pidiendo a gritos un cambio de estrategia a un modelo de pactos, o a una desviación respecto del proyecto original.
- Haciendo un símil con la física nuclear, la conjunción de estas tres condiciones, provoca la "ignición", esto es, el modelo llega a ser auto-mantenido, y la productividad se dispara por efecto de múltiples realimentaciones. Es el estado "ideal" de este modelo.

No es sencillo llegar a la "ignición". Los usuarios y voluntarios deben ser convencidos de mantenerse en el proyecto, a costa de grandes inversiones por parte de la empresa o director de proyecto. La figura y carisma del líder es fundamental para mantener el estado de ánimo. Una vez alcanzada la ignición, casi no habrá que hacer tareas de captación: el éxito tiene su propia inercia, y la afluencia de "adeptos" será casi automática. Es el momento de recoger los frutos del trabajo....

9 El Software Libre: un producto políticamente correcto.

9.1 Resumen del capítulo.

- ¿Un modelo socialista?.
- Independencia económica y empresarial.
- El software libre en la universidad.
- El punto de vista del usuario.
- Frentes económicos: EEUU versus Europa.
- Por qué Europa apoya el software libre.

Analicemos a continuación una serie de aspectos políticos y sociales que han surgido alrededor del software libre. Con independencia de las motivaciones, modelos matemáticos, y -por qué no decirlo- la ayuda que recibe debido a la falta de calidad del software comercial, es un hecho que el software libre está "de moda". No sólo eso, sino que es tomado ya en serio por políticos, gobiernos, y por aquellos estamentos que tradicionalmente han detentado el poder. La fuerza arrolladora de este movimiento no sería tal sin el apoyo de estos estamentos.

Conforme al espíritu de este ensayo, creo que en este movimiento hay también una serie de intereses, muchas veces económicos, que hacen que, por ejemplo, la comisión europea se plantee temas como la utilización de formatos de documentación abiertos en sus métodos de trabajo, o que el ministerio de defensa francés utilice Linux como sistema operativo...

9.2 ¿El socialismo del software?.

Muchos pensadores dicen que el software libre es un modelo de trabajo que equipara a todos los miembros de la comunidad en cuanto al acceso a la información. Postulan, no sin razón, que en muchas ocasiones la legislación de protección de patentes y de software impone límites al desarrollo del conocimiento humano, y mantiene las barreras de poder existentes en la actualidad. Afirman que el software libre proporciona un modelo "socialista" de distribución y acceso al conocimiento. Es opinión del autor que no es exactamente cierto: como vimos en el primer capítulo el acceso a la información no es garantía de conocimiento, ni mucho menos de poder. Es perfectamente posible, utilizando las técnicas descritas, utilizar el modelo de desarrollo del software libre con criterios empresariales, y como muchas empresas están demostrando hoy en día, no sólo obtener beneficios, sino que en algunos casos, como pueda ser Apache Inc. o Sendmail Inc. constituir lo que son de hecho monopolios en el software.

Pues el principio de selección biológica y de ocupación de nichos se aplica por igual a todo tipo de software, con independencia de su origen. Al mercado no le importa si el servidor web es público o no: mirará su relación coste-calidad, y escogerá el mejor. El software libre no es la panacea, sino que sólo lo es si es además bueno. No obstante cuenta con la inestimable ventaja de que desbanca muy rápidamente a cualquier otro competidor -libre o de pago- de inferior calidad.

9.3 Independencia tecnológica y balanza comercial.

Lo que sí es cierto es que a los gobiernos les proporciona independencia respecto de otros países: las tasas y aranceles de importación son nulos, y el uso de software libre favorece la balanza comercial. Por otro lado, a medio plazo , y con unos buenos planes de desarrollo, puede tender a hacer disminuir la desventaja tecnológica entre países. Desde este punto de vista no es de extrañar que los países tecnológicamente "fuertes" pongan trabas a la exportación de tecnología, y sean los que más férreo control -ideologías aparte- intentan ejercer sobre Internet. Las restricciones de Estados Unidos a la exportación de tecnologías de encriptación de datos son un ejemplo de este movimiento. Modelos simulados de partidas del dilema del preso, muestran como los participantes "débiles" tienden a agruparse y a colaborar entre si, dejando de lado al "fuerte".

La dependencia tecnológica y los altos costes de la importación de software foráneo se hacen más evidentes en aquellos entornos donde dicha tecnología es necesaria para el trabajo. No puede extrañar a nadie que en un estamento público, o una ONG, o en cualquier sitio que tenga que ajustar presupuestos, asuste la sola idea de tener que renovar todo el parque informático para actualizarse a la "última versión de XXXX98R2" que "corrige errores y se bloquea menos". ¿Cómo se puede aceptar que un software comercial se bloquee, el vendedor quede libre de toda responsabilidad , y encima pagar por una nueva versión, que el vendedor sigue reconociendo que tampoco funciona bien?.

9.4 La universidad y los centros de investigación.

Del mismo modo, en los centros de investigación es preciso investigar: no sólo se debe utilizar un programa, sino que se debe aprender su tecnología y diseño, se deben poder estudiar alternativas... en fin, es preciso que el software sea algo más que una herramienta de trabajo: debe ser también objeto del trabajo. No puede extrañarnos, pues que las universidades y centros de investigación no sólo hayan sido -y continúen siendo- cuna del software libre, sino que hasta ahora han sido los principales usuarios y clientes de dicho software.

La irrupción del mundo comercial ha derivado el esfuerzo hacia el cliente final -que es el que paga-, pero aún así la universidad es uno de los impulsores básicos de dicho modelo. Además es una baza de cara a las generaciones futuras: cada año miles de estudiantes que han estado conviviendo y utilizando el software libre salen al mercado de trabajo con todo ese bagaje a sus espaldas, y utilizarán y potenciarán dicho software en aquellas empresas donde trabajen.... El modelo se realimenta de nuevo. Las casas comerciales luchan contra este efecto mediante las denominadas "tarifas de estudiante", "licencias campus", etc... pero aunque puedan introducirse en las facetas "ofimáticas" de la informática, no pueden meterse en las de desarrollo e investigación.

9.5 El punto de vista del usuario.

Por último queda el punto de vista del cliente: no hablamos del voluntario-programador, sino del que paga por la documentación, el que recibe cursos, contrata mantenimientos, etc... El software libre le proporciona ventajas económicas, incluso a igualdad de precio del software:

- La posibilidad de escoger distribuidor.
- Un servicio técnico mucho más eficiente y rápido que el de un software comercial (adiós al "no se puede" o "reinstale de nuevo").
- La posibilidad de sugerir, o incluso añadir por su cuenta aquellas funcionalidades que necesite.

No todo es color de rosa: el cliente deberá conocer perfectamente el mundo Internet, y deberá ser capaz de poder asumir el papel de "control de calidad" del software. Afortunadamente hoy en día la calidad del software libre de uso en explotación es tan grande que el cliente sólo se tiene que preocupar de buscar fallos en raras ocasiones. A pesar de todo, y sobre todo en empresas de bajo presupuesto en equipamiento, la utilización de software libre es una necesidad (a menos que se prefiera lidiar con el software "pirata"....).

9.6 La actitud de Europa frente al software libre.

A este entorno hay que añadir el concepto de "cultura de lo social". La protección de derechos del individuo, la cultura del bienestar, el concepto de lo políticamente correcto, la lucha contra los monopolios tanto tecnológicos, como económicos, como culturales, hacen que Europa sea un caldo de cultivo ideal para una economía del software libre. La dependencia tecnológica con Estados Unidos, y la ausencia de empresas líder en el mundo del software en Europa, hacen que la Unión europea tenga una postura neutral -cuando no favorable- a este movimiento. Eso sin contar con los beneficios económicos que reporta a la balanza comercial con los Estados Unidos....

Así pues, si para otras culturas o países en vías de desarrollo, el consumo de software libre es una necesidad, por motivos puramente económicos; en el caso de Europa es una vía de poder. La Unión Europea dedica un gran capítulo de gastos al apartado de investigación y desarrollo tecnológico.

Desde este planteamiento, el uso de software libre es un aliciente para la economía europea. Bruselas está estudiando y legislando la informática con el objetivo de la independencia tecnológica y comercial. Están en estudio temas como la criptografía, los modelos de documentación, de comercio electrónico. Existe legislación europea sobre tratamiento de información, propiedad intelectual.... Y siempre orientada con los principios del estado de bienestar europeo. Para Europa, el software libre es "políticamente correcto"... siempre y cuando produzca beneficios.

El mundo Linux es un ejemplo de producto que se vende como "europeo": nació en Finlandia, con multitud de desarrolladores en Inglaterra, Finlandia, y sobre todo Alemania, con España trabajando activamente en temas de documentación e internacionalización... y abriendo el mercado al mundo hispanoamericano. Linux se usa en la mayor parte de las universidades y centros de investigación europeos, en el ministerio de defensa francés... El CERN fue la cuna del World Wide Web, del HTML, y del comienzo del acceso sencillo a los conocimientos de la red.... Es una situación que no se puede dejar de tener en cuenta; y por supuesto, de aprovechar comercialmente.

10 Marco legal del software libre.

10.1 Resumen del capítulo.

- Marco legal del software.
- El derecho a recibir y emitir libremente información veraz.
- La ley de Propiedad intelectual.
- ¿Es patentable la información?.
- Control de las licencias de software libre.
- Problemática de la documentación libre.

Hemos visto en anteriores capítulos el modelo de empresa basada en el software libre, los fundamentos filosóficos y modelos matemáticos. Hemos hecho una parada en el aspecto político y social de dicho software. Dedicaremos este capítulo a un tema, que tiene una importancia capital en el software libre: la cobertura legal de dicho software.

Según hemos visto, dada la naturaleza intrínseca de información libre que implica el modelo OSS, es necesario establecer una serie de protecciones a dicho software, que se plasman en las denominadas licencias de uso y distribución. Veremos una serie de temas legales que afectan a dichas licencia.

10.2 Marco legal del software.

La libertad de información es un derecho consagrado en nuestra Constitución. La legislación y jurisprudencia española, tiende a entender dicha libertad en un sentido amplio, proviniendo las principales limitaciones a la protección al honor y la intimidad. De hecho temas como la ingeniería inversa, la realización de copias de seguridad, el uso a prueba del software, etc. son concebidos en un sentido permisivo en tanto que están autorizados siempre que no lesionen los derechos del propietario.

A partir de esta concepción tenemos una serie de leyes orgánicas que acotan y definen la libertad de información:

La Ley General de Telecomunicaciones, establece la libre recepción de señales, con independencia de su emisor. Las restricciones se aplican a los emisores, en forma de licencias, tasas, control del modo de emisión, etc... Actualmente está en estudio la posible aplicación de dicha ley a sistemas de transmisión de datos tipo Internet. El problema es que la aplicación de dicha ley a un entorno transnacional es de difícil solución. Por ello, la Comisión Europea está realizando diversos estudios sobre el tema, para uniformizar la legislación. Un aspecto importante de la LGT es que establece la titularidad de las emisiones: la recepción de la señal es libre. No obstante su re-emisión está condicionada a la autorización del propietario. La aplicación a la transmisión de datos por Internet es obvia, y uno de los puntos claves de la legislación sobre la circulación de datos por la red. Otro aspecto es el del concepto de portador de la señal: se refiere al concepto del canal por donde circula la información. El portador debe garantizar el "derecho de paso", y poner los medios técnicos para que la señal llegue a su destinatario. En lenguaje Internet: el proveedor, debe garantizar la conectividad y encaminamiento.

La Ley Orgánica Reguladora del Tratamiento Automatizado de Datos (LORTAD), no sólo trata temas de protección de la intimidad, sino de garantías de integridad, autenticidad, etc. de los datos que se transfieren. Los temas de seguridad, criptografía, comunicaciones seguras, dinero en Internet, etc. caen bajo el paraguas de dicha ley. En este aspecto, la legislación europea difiere grandemente entre los diversos países. Tenemos así el ejemplo de Francia, donde toda comunicación encriptada no autorizada está prohibida (?), o bien países como Finlandia, donde cada uno puede hacer casi lo que quiera. La Comisión Europea está estudiando una serie de soluciones intermedias, donde se establece una cierta libertad a cambio de restricciones en diversos campos. Caso especial de estudio es el de importación y exportación de técnicas de criptografía. Otro campo de aplicación es el de los contratos electrónicos y el intercambio electrónico de documentación. Existe una amplia normativa que regula el comercio en Internet. Se regula el uso de firmas digitales y se establece el concepto de emisor de certificados digitales de autenticidad. Un último aspecto de la LORTAD, trata sobre los derechos y deberes del propietario de las bases de datos, así como la reglamentación sobre su uso. Diferencia entre la titularidad de la base de datos y la titularidad de los contenidos.

La ley de protección al honor y la intimidad, es de aplicación a los sitios web. La pornografía es ilegal, y los sitios web residentes en España caen bajo la jurisdicción de las leyes españolas, con independencia del usuario. Junto con la LORTAD, se establecen cláusulas de confidencialidad, autenticidad, accesibilidad, etc. a los diversos datos, así como se regula la transferencia y compartición de dicha información. Se establece el concepto de responsabilidad civil, en el sentido de que existen responsabilidades penales por el uso indebido o falsedad en los datos. Desgraciadamente la responsabilidad civil no está extendida al software, por lo que bajo la legislación Española, aún no es posible demandar al creador de un software erróneo, ni reclamar daños y perjuicios por un funcionamiento incorrecto o impropio de un programa.

Es de notar que realmente no existe ninguna ley especifica sobre la información en Internet. La jurisprudencia existente utiliza la legislación existente en la actualidad referente a otras áreas, lo que frecuentemente provoca colisiones e incongruencias entre varias leyes.

De especial interés es la Ley de Propiedad intelectual, y su aplicación al software. Básicamente, la jurisprudencia actual parte de identificar el concepto de producto software con el de el creación literaria o artística: el software es una cosa que se utiliza como un libro: una sola persona a la vez en un único sitio. Resulta curioso el artículo de la Ley de Contratos del Estado, que permite a éste utilizar software legalmente adquirido del modo que considere más conveniente con independencia del uso original... esto desemboca en licencias "campus" para universidades y organismos públicos.

Recientes actualizaciones han incluido el concepto de software, y de programa de ordenador. Se establece la titularidad del software, los derechos de uso y copia. Se garantiza que el derecho de uso no conlleva la transferencia de titularidad. Se define el concepto de licencia y las condiciones de uso y restricción de uso del producto software. Especial mención merece el punto donde se establece la no posibilidad de cesión del derecho de uso, salvo acuerdo en contra. Esto incluye el alquiler y préstamo del software: es ilegal que un vídeo-club alquile software si no tiene autorización del propietario.

10.3 Legislación sobre software libre.

El mundo del software libre no está citado en ninguna legislación. El autor ha visto la perplejidad, cuando no el regocijo con que expertos en derecho le contestaban respecto a las consultas acerca de legislación sobre el software libre. Tras la lectura de la GPL, lo más aproximado que se ha encontrado sobre legislación aplicable es el concepto mercantil de franquicia.

La similitud, es bastante plausible: existe una marca comercial (nombre del programa) un propietario (el creador o dueño del copyright) que establece unos derechos de uso y explotación de la marca comercial (la licencia GPL). Existen diferencias, por supuesto, especialmente en cuanto al uso del derecho de explotación por parte de terceras personas (redistribución). En cualquier caso se hace patente una necesidad fundamental en el software libre: el garantizar la titularidad del producto.

Porque la única garantía legal aplicable en el software libre es la de la titularidad: es fundamental, para que el modelo de software libre en la empresa sea viable, que dicha titularidad sea reconocida y mantenida bajo cualquier circunstancia imaginable. El programa debe estar registrado convenientemente, y la licencia de uso debe reflejar claramente este hecho.

Aquí se produce un hecho fundamental que diferencia la legislación europea con la Americana: el precinto que encontramos en los paquetes de software que nos avisa de que su rotura implica la aceptación de la licencia.... carece de validez. Nuestra legislación precisa de un contrato o una aceptación explícita de la licencia, con conocimiento de las dos partes. Se admite como firma del contrato la solicitud de clave de registro, o la introducción de dicha clave en el proceso de instalación o activación del programa.

De cara a la GPL, esto implica que su aceptación debe ser explicita: no basta con acceder y utilizar el programa para considerar que la licencia es aceptada. Es necesario el registro para que tal licencia tenga validez en Europa. Por ello es tan necesario el que el software libre esté perfectamente registrado y con el copyright vigente. Es casi imposible con la ley en la mano perseguir un uso abusivo de un programa GPL. De hecho las infracciones a dicha licencia son resueltas por los usuarios de la red en forma de boycott al infractor.... lo cual suele ser mucho más efectivo que la actuación legal.

10.4 ¿Es patentable la información?.

Todo esto nos lleva al concepto mismo del software libre y de la libertad de información: el saber qué información está sujeta a titularidad y cuál es distribuíble. En el tema del software nos encontramos con una paradoja: muchas legislaciones diferencian el concepto de algoritmo del concepto de implementación, y añadiendo el concepto de interfaz. Según nuestra legislación, sólo la implementación y el interfaz están sujetos a titularidad. La legislación Americana permite también patentar algoritmos, mientras que en todos los casos el interfaz de programación debe ser público, entendiendo por tal publicidad el que su conocimiento -no su uso- es asequible a todo el mundo.

Desgraciadamente, la evolución de los lenguajes de programación ha hecho que muchos de ellos, no consistan sino en la descripción formal de un algoritmo, ¿cómo patentar este tipo de software?. Otro punto conflictivo lo constituyen los lenguajes de macros o de scripting: el ejecutable y el código fuente son una misma cosa.

La legislación española solventa este problema de una forma bastante ingeniosa: define como programa una secuencia de datos que tienen una funcionalidad determinada... explícitamente incluye todo lo que acompaña a dicha secuencia (manuales, código fuente, etc.) como parte integrante del software.

Muchos autores reconocen pues que el software no es patentable -es decir sujeto a propiedad industrial- pero sí le reconocen derecho de autor -es decir, sujeto a propiedad intelectual-. Este es el caso de la legislación española, pero no de la americana. En cualquier caso la discusión está abierta.

En concreto, las leyes españolas especifican que el software sólo está sujeto a propiedad intelectual, y que no es patentable ningún producto software, salvo que sea una implementación directa de un producto patentado. Ejemplo típico: el software de desarrollo que el fabricante de un chip proporciona con el chip. Explícitamente, los algoritmos, elementos básicos de la estructura de un programa (bucles, saltos, subrutinas), y especialmente los API's no pueden ser patentados... Curiosamente, tanto los API's, como los lenguajes de programación sí están sujetos a propiedad intelectual, y por tanto el autor puede exigir el cobro de los derechos de uso.

El software libre es el último eslabón de la cadena: un producto que sólo está sujeto a titularidad y cuya licencia permite el libre uso y distribución. Dada la "excentricidad" de este planteamiento no es de extrañar la falta de legislación. Es de esperar que a medida que el uso de Internet como medio de intercambio de información, junto con la evolución del concepto de "programa", vaya surgiendo una legislación apropiada.

Un último punto: del hecho de que el software libre sólo esté sujeto a titularidad, implica que los costes de dicho software deben derivarse de los derechos de autor -que en el software libre, y por cuestiones de simple supervivencia del proyecto, son nulos- , y de los costes de producción y distribución. No tiene sentido cobrar por patentes, derechos empresariales, licencias de uso o desarrollo, etc... Extrapolando dicho hecho, el IVA aplicable al software libre debería ser el correspondiente, no a productos industriales, sino al de libros y documentación... que Hacienda tome nota :-)

En los apéndices, el lector podrá encontrar referencias a los diversos modelos de licencia de software libre, así como las leyes y directivas que se aplican al software.

[JAMC: añadir problemática de la documentación libre]

11 Conclusiones.

A lo largo de este documento hemos presentado al lector la panorámica del software libre desde el punto de vista empresarial. Como resumen del ensayo vamos a enumerar una serie de conclusiones:

11.1 Software libre como modelo de desarrollo sostenible.

Hemos visto como el software libre es susceptible de convertirse en modelo de desarrollo empresarial, y como hoy en día hay empresas que están obteniendo grandes beneficios trabajando con software libre.
Del mismo modo hemos estudiado como para algunas organizaciones y sociedades en desarrollo, este modelo es el único viable para su desarrollo tecnológico.

Por ello, se puede afirmar, sin temor a equivocarnos que este modelo es viable, y que constituye un modelo de desarrollo sostenible, basado en que la compartición de información constituye una manera de aumentar el bienestar colectivo, y que con una adecuada orientación y gestión, el modelo es, no sólo productivo, sino también rentable.

11.2 La empresa en la sociedad de la información.

Hoy en día la empresa en las sociedades industrializadas está orientada mayoritariamente al sector servicios. El hecho de considerar la información como un bien común, y orientar la producción de software, - y en general de todas aquellas materias relacionadas con el aumento del conocimiento - obliga a la empresa a un cambio de actitud, tanto en la orientación empresarial como en el modelo de trabajo a aplicar. La metodología del software libre cambia el concepto de software como producto, por el de software como servicio de valor añadido.

11.3 No perder el objetivo: obtener beneficios.

A pesar de todo no se puede perder el objetivo: una empresa debe buscar la rentabilidad. Adoptar el modelo del software libre implica buscar nuevas fuentes de financiación, nuevos nichos de mercado, así como una mayor orientación a los deseos del cliente. La relación entre productor y consumidor pasa a ser bi-direccional, y se debe garantizar que todas las partes ganen con el intercambio. Los modelos matemáticos nos ayudan a analizar este proceso, y a poder prever cuál va a ser la reacción del mercado.

11.4 El futuro del software libre.

Con el advenimiento de la empresa al modelo de software libre es de prever que dicho software sufra una evolución. En opinión del autor, la apariencia es que se está produciendo un proceso de focalización de los objetivos, de manera que:

- Los proyectos coordinados por organizaciones sin ánimo de lucro se centrarán en el software como herramienta, esto es:
  - El núcleo del sistema operativo.
  - El entorno de trabajo.
  - Las herramientas de desarrollo.
  - Las aplicaciones de productividad.
  - Control de estándares y modelos de implementación.
- Las empresas obtienen más beneficios del software de valor añadido, tales como:
  - Herramientas de administración y gestión.
  - Soluciones a medida.
  - Proyectos de integración.
  - Herramientas específicas para soluciones particulares.

Es de esperar que el proceso de adopción del software libre por los grandes de la informática conlleve una cierta "ralentización" del desarrollo, para así proveer de unos API's más estables y documentados. Del mismo modo, la adopción de este modelo puede dar lugar a una mejora en los estándares, pues como hemos visto en uno de los capítulos, la búsqueda de extensiones a los estándares aceptados conlleva un mayor coste económico y un rechazo por parte del cliente.

Por último, el modelo de compartición de información puede ser exportado a otros mercados, no sólo el del software. Tenemos una primera introducción en el concepto de franquicias, de cesión de derechos de explotación, etc. Es de esperar una legislación clara respecto al software libre, en especial en el campo de la elaboración de documentos, y en la publicación en Internet.

12 Apéndices.

12.1 Referencias bibliográficas.

Joint Ventures, Alianzas, Transferencias tecnológicas, Know How.


```
Eduardo Paz.
"Cómo hacer negocios en Internet".
Ed. Gestión 2000.
ISBN 84-8088-283-2.
```


Muestra las técnicas de cooperación que utilizan las empresas que tienen Internet como medio de trabajo.

Técnicas de gestión de empresas orientadas al comercio en Internet.


```
Vince Emery.
"Negocios en Internet, expansión y crecimiento".
Anaya Multimedia. Colección Vía Internet.
ISBN 84-415-0408-3.

Enrique de la Rica.
"Marketing en Internet".
Anaya multimedia.
ISBN 84-415-0186-6.
```

Existen multitud de libros de los que éstos dos son una muestra: Analizan la actitud de la empresa ante la red. Describen metodologías "Internet-Oriented", así como detallan el proceso por el que una empresa se debe abrir a Internet, y cómo adaptar su modelo de trabajo al nuevo entorno.

Política europea sobre la red. Aspectos legales del Software.

```
Juan Viesca.
"La unión Europea en Internet".
Anaya Multimedia.
ISBN 84-415-0586-1.
```

Aunque no trata específicamente sobre los modelos de cooperación, analiza la problemática del comercio en Europa, la dependencia tecnológica, la política de ayudas a la empresa. Contiene además un directorio de recursos para la empresa europea en la Red.

```
Carlos Barriuso Ruíz.
"Interacción del derecho y la Informática".
Ed. Dykinson.
ISBN 84-8155-148-1.
```

Un recorrido completo desde la introducción a la informática para expertos en derecho, tratamiento informático de información jurídica y descripción de la legislación aplicable a la informática, indicando leyes y directivas europeas.

Modelos empresariales de gestión.

```
Jose María Oirtiz Ibarz.
"La hora de la ética empresarial".
Mc Graw Hill.
ISBN 84-481-0320-3.
```

Analiza la problemática de las técnicas empresariales actuales, los problemas del uso y abuso de situaciones de monopolio. Plantea estrategias de cooperación como modelo alternativo.

```
Demetrio Sáez y José Cabanelas.
"Cooperar para competir con éxito".
Ed. Pirámide.
ISBN 84-368-1090.
```

Describe los riesgos y beneficios de la cooperación empresarial, modelos de trabajo entre empresas, y técnicas para conjuntar intereses comunes, y allanar diferencias, manteniendo la identidad empresarial.

```
Francés Cairncross.
"La muerte de la distancia".
Paidós empresa.
ISBN 84-493-0626-4.
```

Plantea los problemas de la globalización, de la sociedad de la información, y de las tácticas empresariales en la Aldea Global.

Bases sociológicas y filosóficas.

```
Loirdes Munduate Jaca.
"Psicología social de la organización".
Ed. Pirámide.
ISBN 84-368-1055-4.
```

Plantea las bases sociológicas de la organización empresarial, la figura del líder, y los modelos de relación entre los componentes de la cadena de producción.

```
Fernando Savater.
"Etica como amor propio".
Ed. Grijalbo-Mondadori.
ISBN 84-397-0259-0.
```

Sienta las bases filosóficas y sociológicas de los modelos de cooperación egoísta, así como la formalización de la "Etica de lo útil". Un texto de lectura difícil para las personas educadas en los valores del humanismo cristiano, pero de plena aplicación en la sociedad de valores capitalista actual.

```
William Poundstone.
"El dilema del prisionero".
Alianza Editorial.
ISBN 84-206-0747-9.
```

Mucho más que la exposición del juego del dilema del preso. Contiene multitud de reseñas históricas, estudios basados en la realidad, y muestra los resultados previsibles a través de la teoría de juegos comparándolos con los hechos tal y como ocurrieron.

Modelos matemáticos.

```
John Von Newman.
"The theory of Games and Economic behavior".
Princeton University Press, 1953.
```

El primer texto publicado sobre el tema, y base de todos los desarrollos y estudios posteriores.

```
Morton D. Davis.
"Introducción a la Teoría de Juegos".
Alianza Editorial. Col. Ciencia y Tecnología.
ISBN 84-206-7905-4.
```

Texto orientado a los profanos, con multitud de ejemplos y ejercicios de aplicaciones de situaciones concretas, acompañados de su correspondiente modelización y estudio.

Otros.

No puedo por menos que citar las inapreciables dotes de humor y sentido común que aparecen en el libro "El principio de Dilbert" de Scott Adams. Si la mitad de las empresas siguieran su doctrina el mundo empresarial no sería lo que es hoy ¿o sí?.

Del mismo modo no puedo sino rendir homenaje póstumo a Isaac Asimov por sus series de Robots y de la Fundación. Fue leyendo "Fundación e imperio", donde leí mis primeras referencias a la ética de lo útil", y donde aprendí el concepto de los modelos de desarrollo a corto y largo plazo. Tal vez una lectura no demasiado filosófica, pero que consiguió interesarme por el tema. Bastantes años después tengo que agradecer a todos estos libros -a pesar de mi educación técnica- mi creciente interés por la sociología.

12.2 Publicaciones.

[JAMC: añadir URLs a las referencias]

El dilema del preso.

En la revista "Investigación y Ciencia", la sección de "juegos de Ordenador", ha dedicado multitud de números al juego del Dilema del preso. El lector tiene acceso a código fuente de algoritmos, simulaciones, resultados de torneos computerizados, y modelos genéticos de dicho juego. Además, existen en la misma revista multitud de artículos sobre modelos de cooperación egoísta en el mundo animal.

La catedral y el bazar.

Un clásico del mundo del software libre. Eric S. Raymond, escribió, junto con este diversos ensayos y artículos sobre dicho modelo de desarrollo, estudiando el papel de cada uno de los intervinientes, y explicando las fases del desarrollo. Como lecturas auxiliares podemos citar "Knowing the Knoosphere", y "How to become a hacker".

El derecho a leer.

Cuando Richard Stallman, escribió este pequeño cuento, se inició una nueva época en el mundo software: el de la lucha por la libertad de información y divulgación. Años después, en pleno auge del mundo del software libre, este texto sigue siendo un aviso ante la presión de los intereses políticos y económicos por restringir la información en aras del mantenimiento del poder.

El manifiesto hacker.

El 14 de Agosto de 1989, durante la Galactic Hacker Party Lee Felstein propuso -y fue aprobado por unanimidad- lo que constituye una declaración de principios acerca de lo que debe ser la actitud y el estilo de trabajo del programador en el mundo libre.

El documento "halloween".

Microsoft publicó este documento como un memorándum interno acerca de cuál debía ser su política empresarial ante la expansión del software libre. Plantea estrategias típicas de empresa con actitud monopolística: personalización de protocolos, compra de empresas, chantaje a los fabricantes de hardware. Todo un ejemplo de "buen hacer" empresarial.

12.3 Artículos de prensa y documentos en Internet.

Dado lo actual del tema, existen multitud de artículos tanto en prensa electrónica como impresa. Es imposible citarlos todos, e indicar referencias. A título de ejemplo, cito los que han llegado a mis manos:

- "Un pueblo barcelonés, pionero en instalar Linux en organismos oficiales"
  CiberPaís (25-Marzo-1999).
- "Why Open Source is the optimun Economic paradigm for Software"
  Dan Kaminsky (2-Marzo-1999)
  http://doxpara.netpedia.net/core.html
- "Is your software in danger of Termination?"
  Bruce Perens (15-Marzo-1999).
- "To be OSS, or not to be OSS, that is the question."
  Paul Ferris (32bits online Marzo 1999).

- "Trampa en el Ciberespacio"
  Traducción de Roberto Di Cosmo 9-Oct-1998 del documento original del mismo nombre de Nikos Drakos.
- "Linux, Presente y Futuro"
  Ismael Olea (28-Enero-1999).
- "Where does Linux want to go Today?"
  Eric CaldWell (32bits online, 24-Marzo-1999).
- "Take my job please!"
  Eric S. Raymond (29-Marzo-1999).
- "Apuntes sobre software libre"
  Jesús M. González Barahona (18-Marzo-1999).
- "Linux, Nacido libre"
  Revista "Ciencia y Vida", Diciembre 1998.
- "Golpe con efecto"
  Revista "Planeta Humano", Diciembre 1998.

13 Epílogo. Agradecimientos.

"Use the Source, Luke".

A lo largo de este ensayo he intentado acercar al lector al software libre, desde un enfoque distinto al tradicional: considerar software libre como un producto de mercado, que proporciona beneficios

Si el lector considera acertado o erróneo dicho planteamiento, o cree que se le podría dar otro enfoque, estaré encantado de recibir sugerencias para futuras ediciones del texto. En cualquier caso, consideraré que he cumplido mis objetivos si después de la lectura de este ensayo el lector considera que el software libre ya no es una curiosidad de "unos locos" sino que es un modelo de economía serio, y posiblemente viable.

No quiero despedirme sin dar las gracias a todos los que directa o indirectamente han apoyado la escritura de este texto:

- A Ismael Olea y a la gente de Hispalinux, por su trabajo en pro del software libre, así como su oferta de inclusión de este ensayo en las publicaciones del proyecto LuCas.
- A Jesús González Barahona, por sus comentarios y sugerencias acerca de la licencia del texto.
- A la asociación de estudiantes Eurielec, por los buenos ratos, la amistad y la distribución en castellano Eurielec Linux.
- A Pablo Ortiz, y Eduardo Toribio, que me dieron la oportunidad de volver a escribir en revistas de informática, siendo un perfecto desconocido.
- Al Departamento de Ingeniería de Sistemas Telemáticos, de la U.P.M. de Madrid por permitirme el trabajo con software libre.
- A todas las personas que a través del correo y los grupos de noticias han enviado comentarios al ensayo.
- Y por supuesto: al hecho de la existencia del software libre, gracias al cual me estoy pagando la hipoteca... ;-D (para que luego digan que esto no da dinero).

Este documento puede ser obtenido en su formato SGML-LinuxDoc original de
ftp://drake.lab.dit.upm.es/pub/docs/freesoft/empresa.sgml

Del mismo modo puede ser consultado en la World Wide Web en:
http://drake.lab.dit.upm.es/~jantonio/articulos/empresa.html

El documento ha sido editado íntegramente en formato SGML-LinuxDoc en un ordenador Pentium-166 ejecutando Eurielec Linux 2.1 con núcleo del sistema operativo Linux 2.2.4. Para su desarrollo se han utilizado herramientas de software libre, sin intervención directa o indirecta de ningún tipo de software propietario

Juan Antonio Martínez Castaño
Madrid, 12 de Abril de 1999