

# Modelos de negocio con software libre

Irene Fernández Monsalve

PID\_00145049



Universitat Oberta  
de Catalunya

[www.uoc.edu](http://www.uoc.edu)



# Índice

|  |    |
|--|----|
| <b>Introducción</b> .....  | 5  |
| <b>Objetivos</b> .....   | 6  |
| <b>1. Caracterizando modelos de negocio con software libre</b> .....   | 7  |
| <b>2. Clasificaciones según distintos autores</b> .....  | 8  |
| 2.1. Clasificaciones de Hecker y Raymond .....   | 8  |
| 2.2. Clasificación del Grupo de Trabajo Europeo de Software Libre (European Working Group on Libre Software) ..... | 10 |
| 2.3. Estudios empíricos .....  | 11 |
| 2.4. Propuesta de clasificación .....  | 14 |
| <b>3. Modelos de negocio con software libre</b> .....  | 16 |
| 3.1. Especialistas/Verticales (una aplicación libre como principal producto) .....                                 | 16 |
| 3.1.1. Modelos mixtos: licencias dobles .....  | 17 |
| 3.1.2. Modelos mixtos: núcleo del producto libre y accesorios propietarios .....                                   | 20 |
| 3.1.3. Modelos libres: "Venta distribuida" del producto .....  | 22 |
| 3.1.4. Producto libre más servicios asociados .....  | 24 |
| 3.1.5. Software como servicio .....  | 26 |
| 3.2. Servicios asociados al software libre .....   | 27 |
| 3.2.1. Empresas distribuidoras de plataformas .....  | 31 |
| 3.2.2. Grandes integradoras .....  | 33 |
| 3.2.3. Servicios en torno al software: pequeñas y microempresas .....  | 35 |
| 3.3. Mercados auxiliares: Hardware .....   | 37 |
| 3.4. Otros mercados auxiliares .....   | 40 |
| <b>Resumen</b> .....   | 42 |
| <b>Bibliografía</b> .....  | 43 |



## Introducción

En los módulos anteriores hemos examinado el mercado del software, los tipos tradicionales de empresas en el sector, y las posibilidades que brinda el software libre dentro de este marco. Dedicaremos este módulo al estudio de los modelos de negocio más usuales que se crean en torno al software libre, y de algunos casos concretos.

Sin ninguna duda, el software libre está emergiendo en los últimos años como un elemento clave en nuevos modelos de negocio. Después del desinfe de la burbuja tecnológica (popularmente conocida como la burbuja de las punto com) a principios de esta década, el software libre ha supuesto un motor para la creación de nuevas empresas en el sector tecnológico, consiguiendo atraer crecientes cantidades de capital-riesgo. En el 2004, se invirtió un total de \$149 millones, distribuidos entre 20 nuevas empresas. En el 2006, la cantidad subió a \$475 millones, distribuidos entre 48 iniciativas empresariales.

A las empresas ya establecidas y reconocidas, como Red Hat o MySQL, se ha unido una nueva generación de numerosas empresas que centran sus estrategias en el uso y desarrollo de software libre. En los próximos años veremos cuál es la evolución real de estas nuevas empresas, y si su modelo de negocio sostenible a largo plazo.

### Bibliografía recomendada

Para más información podéis consultar: Larry Augustin (2007). "A New Breed of P&L: the Open Source Business Financial Model". *Open Source Business Conference (OSBC)*. [http://www.osbc.com/live/images/13/presentation\\_dwn/A\\_New\\_Breed\\_of\\_P\\_and\\_L.pdf](http://www.osbc.com/live/images/13/presentation_dwn/A_New_Breed_of_P_and_L.pdf)

| Primera generación  | Segunda generación  | Tercera generación  |
|---|---|---|
| <p><b>Con cotización en bolsa:</b><br/>Red Hat, Caldera (ahora SCO), VA Linux (ahora VA Software), Turbolinux</p> <p><b>Adquiridas:</b><br/>SUSE, Cygnus</p> <p><b>Otras:</b><br/>LinuxWorks, Linuxcare (ahora Levanta), Sendmail</p> | <p><b>Con cotización en bolsa:</b><br/>Trolltech, Sourcefire, Mandrakesoft (ahora Mandriva)</p> <p><b>Adquiridas:</b><br/>Conectiva, Lycoris, JBoss, Sleepycat, Ximian, Gluecode</p> <p><b>Otras:</b><br/>MontaVista, MySQL, Zend</p> | <p>ActiveGrid, ActiveState, Alfresco, BitRock, Black Duck, CollabNet, Collax, Compiere, Covalent, DB4O, Digium, Exadel, eZ Systems, Fonality, Funambol, Groundwork, Hyperic, Ingres, Interface21, JasperSoft, Joomla, Laszlo Systems, Medsphere, Mozilla Corp, MuleSource, OpenBravo, OpenLogic, OpenXchange, OTRS, Palamida, Pentaho, rPath, SnapLogic, SourceLabs, Spikesource, SQLite, WebYog, SugarCRM, Talend, Terracotta, Ubuntu / Canonical, Vyatta, WSO2, XenSource, Zenoss, Zimbra, Zmanda, etc.</p> |

Modelos de negocio con software libre: casos de éxito. Tabla y datos de inversiones obtenidos de Marten Mickos (2007). "Open Source: why freedom makes a better business model". *Open Source Business Conference (OSBC)*. [http://www.osbc.com/live/images/13/presentation\\_dwn/Keynote-Open\\_Source\\_Why\\_Freedom.pdf](http://www.osbc.com/live/images/13/presentation_dwn/Keynote-Open_Source_Why_Freedom.pdf)

En este módulo vamos a estudiar algunos de los casos empresariales que se presentan en la tabla anterior, además de otros que, si bien no atraen la atención por sus reducidas dimensiones, resultan bastante significativos. Nos detendremos en las ventajas del software libre que explotan, los problemas que encuentran, y cómo los solucionan. Además, examinaremos distintas taxonomías para caracterizar estos modelos, intentando identificar distintos factores clave que determinan el funcionamiento de la empresa según distintos autores.

## Objetivos

Al finalizar este módulo, se deben alcanzar los objetivos siguientes:

- 1.** Entender las principales clasificaciones de modelos de software libre que se han confeccionado hasta ahora.
- 2.** Conocer los modelos de negocio basados en software libre existentes en la actualidad.
- 3.** Entender los distintos mecanismos de generación de ingresos, y de diferenciación que explotan dichos modelos.
- 4.** Ser capaz de analizar cómo aprovechan el software libre distintas empresas para generar una ventaja competitiva.

## 1. Caracterizando modelos de negocio con software libre

Cuando hablamos de modelos de negocio basados en software libre, a menudo nos referimos a qué nuevas e ingeniosas maneras de obtener ingresos son puestas en marcha, ya que el modelo tradicional, de venta de un producto propietario, ya no es evidente. Las empresas, a diferencia de las personas individuales, deben considerar un importante factor a la hora de participar en un proyecto de software libre: cómo capturar un retorno económico que justifique su inversión.

Tal como hemos visto en módulos anteriores, la idea de que los ingresos que genera el software están directamente vinculados a su venta no se ajusta totalmente a la realidad. La mayor parte del software se desarrolla internamente, y son pocas las empresas para las que la venta de software supone el principal ingreso. En la mayoría de los casos, la oferta de servicios complementarios se hace necesaria para asegurar la continuidad de los ingresos y la supervivencia de la empresa en momentos difíciles.

Por otra parte, en el artículo de Perens examinado en el segundo módulo ("The Emerging Economic Paradigm of Open Source"), hemos visto que el software libre ofrece unas perspectivas económicas (coste y riesgo) mucho mejores que las alternativas propietarias para empresas que necesiten desarrollar software no diferenciador.

En cualquier caso, en este módulo veremos cómo distintas empresas gestionan la propiedad intelectual de sus productos, generando también **modelos mixtos**, que tratan de compatibilizar las ventajas de los modelos libres con la captura de retornos económicos directos en base a la propiedad intelectual. En este sentido, la elección de licencia determinará, en gran medida, el rango de modelos de negocio que una empresa pueda implementar.

### Web recomendada

Para más información sobre Perens:

<http://www.uic.edu/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/1470/1385>

## 2. Clasificaciones según distintos autores

En este apartado, vamos a dar un repaso a distintos intentos de clasificación de modelos de negocio que aparecen en la literatura, deteniéndonos en los factores que cada autor ha considerado claves para agrupar distintos modelos. Además de aproximaciones más teóricas, veremos otras que se han basado en observar empresas existentes de una forma más cualitativa, así como una metodología cuantitativa de clasificación de modelos de negocio en el contexto del **proyecto FLOSSmetrics**. Finalmente, propondremos una taxonomía propia que de alguna manera aúne las propuestas examinadas.

### 2.1. Clasificaciones de Hecker y Raymond

Uno de los primeros autores en escribir sobre las perspectivas de negocio que ofrecía el software libre fue Frank Hecker en 1998, en "Setting Up Shop: The Business of Open-Source Software". En su artículo, toma cuatro categorías de OpenSource.org, y añade otras más, examinándolas en base a:

- Qué empresas ponen en práctica dicho modelo.
- Qué tipos de licencias son apropiadas.
- Qué oportunidades de diferenciación presenta el modelo.
- Qué oportunidades proporciona el modelo para poner precios sobre la base del valor percibido en lugar de sobre la base de costes reales.

A continuación presentamos una tabla resumen de su clasificación, añadiendo otro parámetro de caracterización que, aunque no es mencionado explícitamente por Hecker, constituye una característica fundamental: cómo se originan los ingresos de la empresa.

| Modelo                 | Fuente de ingresos  | Tipo de licencia | Oportunidades de diferenciación  | Oportunidades de precios basados en valor percibido vs. costes | Casos                                  |
|------------------------|---|------------------|--|--|--|
| <i>Support sellers</i> | Venta de servicios relacionados (agrupa todo tipo de servicios, desde desarrollos a medida, a formación, consultoría, etc.) | GPL              | Calidad, precio, y simplificación y mejora de la experiencia de usuario. | Limitada. Posible si goza de mucha reputación.                 | Cygnus Solutions<br>Red Hat<br>Caldera |
| <i>Loss leader</i>     | Venta de otros productos propietarios   | BSD o Mozilla    | Basada en el producto.   | Posible.   | Sendmail<br>Netscape                   |

Resumen de clasificación de modelos de negocio ("Setting up shop: the business of open source" Hecker, 1998)

#### Web recomendada

Para más información podéis consultar:

<http://hecker.org/writings/setting-up-shop>



| Modelo  | Fuente de ingresos  | Tipo de licencia | Oportunidades de diferenciación  | Oportunidades de precios basados en valor percibido vs. costes                                   | Casos                    |
|---|---|------------------|--|--|--------------------------|
| <i>Widget frosting</i>                                      | Venta de hardware   |                  | Basada en el hardware – funcionalidad, rendimiento, flexibilidad, fiabilidad, coste...         | Limitada. Típicamente el esquema de precios de hardware está basado en costes.                   | Corel<br>VA Linux        |
| <i>Accessorizing</i>  | Venta de productos físicos (libros, etc.)   |                  | Calidad del producto (libros, etc.), y lealtad por parte de usuarios "pro-<br>software libre". | Limitada. La reputación de marca puede permitir elevar un poco los precios.                      | O'Reilly &<br>Associates |
| <i>Service enabler</i>                                      | Venta de servicios en línea proporcionados por el programa  | GPL o Mozilla    | Atributos del <i>backend</i> , creación de servicios únicos y útiles.                          | Posible en la medida que se logra crear un servicio único y no imitable.                         | Netscape                 |
| <i>Sell it, free it</i>                                     | Como "loss leader" cíclico  | BSD o Mozilla    | Funcionalidad del software (mientras se mantiene cerrado).                                     | Posible, hasta que el producto se convierte en un bien intercambiable (en ese momento se libera) | –hipotético–             |
| <i>Brand licensing</i>                                      | Venta de derechos de marca. Convive la versión con marca con la versión "genérica".               |                  | Valor añadido por ejemplo, mediante validación y testeo adicional al producto sin marca.       |  | –hipotético–             |
| <i>Software franchising</i>                                 | Venta de franquicia y porcentaje sobre ingresos de las franquicias                                |                  | Como <i>support-seller</i> y <i>brand licensing</i>  | Posible si se goza de reputación.  | –hipotético–             |
| Híbridos (licencias no son ni libres ni propietarias puras) | Limitar disponibilidad de código: venta de licencias bajo condiciones determinadas                |                  |  |  | Trolltech<br>Qt          |
|   | Tratamiento según usuarios – venta a usuarios comerciales   |                  |  |  | Open<br>Group            |
|   | Tratamiento según uso – venta para usos comerciales, o venta para uso en determinadas plataformas |                  |  |  | Qt                       |

Resumen de clasificación de modelos de negocio ("Setting up shop: the business of open source" Hecker, 1998)

En *El Caldero Mágico*, Eric S. Raymond también dibuja el papel del software libre en el mundo empresarial, centrándose, entre otros, en cómo el software libre afecta al "valor de uso" (valor como producto intermedio) y al "valor de venta" (valor como producto final) del software, proponiendo una taxonomía basada en cuál de los dos explota la compañía.

Para Raymond, sólo el valor de venta se ve afectado por un modelo de software libre, por lo que su clasificación describe modelos basados en valor de uso, y modelos basados en el valor de venta indirecto, en los que el software libre hace viable la venta de otro producto o servicio:

- Modelos basados en el valor de uso
  - Compartir costes (por ejemplo, Apache)
  - Compartir riesgo (por ejemplo, Cisco)
- Modelos basados en el valor de venta indirecto
  - Posicionador de mercado (*loss-leader/market positioner*)
  - Venta de hardware (*widget frosting*)

#### Web recomendada

Para más información podéis consultar:

<http://catb.org/~esr/writings/magic-cauldron/>

- Regala la receta, abre un restaurante (*give away the recipe, open a restaurant*)
- Venta de accesorios (*accessorizing*)
- Libera el futuro, vende el presente (*free the future, sell the present*)
- Libera el software, vende la marca (*free the software, sell the brand*)
- Libera el software, vende el contenido (*free the software, sell the content*)

Como podemos ver, Raymond incluye entre los modelos basados en el valor de venta indirecto los recogidos por Hecker, más uno nuevo "Free the software, sell the content". En este modelo, el valor está en la información proporcionada por la plataforma de software, y es esa información la que se vende mediante suscripciones. El software se libera, consiguiendo que sea portado a distintas plataformas, ampliando por lo tanto el mercado potencial del verdadero producto: **el contenido**.

Aunque se propone sólo como modelo hipotético, Raymond se adelanta a los conceptos de "web social", y de cambio de paradigma que propone O'Reilly en su artículo "Open Source Paradigm Shift".

Sin embargo, no llega a reconocer el papel de Internet como plataforma, y del posterior *software as a service*, creyendo que el valor de liberar el software será el de conseguir que se porte a otras plataformas, contribuyendo a la difusión y a la ampliación del mercado.

## 2.2. Clasificación del Grupo de Trabajo Europeo de Software Libre (European Working Group on Libre Software)

Los modelos de negocio presentados por Hecker y Raymond se basan en la observación de empresas que usaban software libre como parte de sus modelos de negocio, pero quizás les falta cierta sistematización y abstracción en su taxonomía. El **grupo de trabajo europeo de software libre**, en su documento "Free Software/Open Source: Information Society Opportunities for Europe?" (<http://eu.conecta.it/paper/>), hace su análisis sobre la base de cómo se financian los proyectos de software libre en lugar de hacerlo sobre la base de los modelos de negocio, e independientemente de si el proyecto tiene relación con alguna empresa en concreto o no:

- Financiación pública.
- Financiación privada sin ánimo de lucro.
- Financiación por quien necesita mejoras.
- Financiación con beneficios relacionados (O'Reilly y Perl).
- Financiación como inversión interna.

### Web recomendada

Más información sobre "Open Source Paradigm Shift" en:

[http://www.oreillynet.com/pub/a/oreilly/tim/articles/paradigmshift\\_0504.html](http://www.oreillynet.com/pub/a/oreilly/tim/articles/paradigmshift_0504.html)

- Otros (bonos, cooperativas de desarrolladores, utilización de mercados para poner en contacto clientes y desarrolladores).

Dentro de su apartado "Financiación como inversión interna", sin embargo, encontramos una clasificación de modelos de negocio, que destaca, entre otras, la posibilidad de generar **ingresos mediante servicios**, gracias a la ventaja competitiva que otorgaría ser los principales desarrolladores de un proyecto de software dado.

| Modelo                                       | Diferenciación  | Ingresos   | Licencias             | Ejemplos                               |
|--|---|--|-----------------------|--|
| Mejor conocimiento                           | Mejor conocimiento del producto: debe ser la desarrolladora del producto, o una colaboradora.                                       | Servicios relacionados: desarrollos a medida, adaptaciones, instalación, integración.    | Libres                | LinuxCare (en sus inicios)<br>Alcove   |
| Mejor conocimiento con limitaciones          | Mejor conocimiento del producto: debe ser la desarrolladora del producto.<br>Se mantiene una parte propietaria.                     | Servicios relacionados, y venta de parte propietaria.                                    | Libres y propietarias | Caldera<br>Ximian                      |
| Fuente de un producto libre                  | Productora, casi en su totalidad de producto libre.   | Servicios relacionados: desarrollos a medida, adaptaciones, instalación, integración.    | Libres                | Ximian<br>Zope Corporation             |
| Fuente de un producto libre con limitaciones | Producto propietario en principio.<br>Liberación posterior como estrategia para ganar adopción y otras ventajas del software libre. | Venta de versión comercial.  | Libres y propietarias | Artofcode LLC<br>Ada Core Technologies |
| Licencias especiales                         | Mejor conocimiento.<br>Oferta de versión propietaria para clientes que no quieran GPL.  | Venta de versión comercial, y servicios relacionados.                                    | GPL y propietarias    | Sleepycat                              |
| Venta de marca                               | En base a imagen y marca, que permitan vender producto a mayor precio.  | Venta de distribuciones, y servicios relacionados (incluyendo certificación y formación) | Libres                | Red Hat                                |

Modelos de negocio basados en software libre. (Manual didáctico de "Introducción al Software Libre")

### 2.3. Estudios empíricos

En "Business models in FLOSS-based companies", Carlo Daffara describe un estudio empírico de los modelos de negocio de empresas basadas en el uso de software libre, emprendido en el contexto del **proyecto FLOSSmetrics**. El estudio también examina cómo esos modelos manejan la comercialización de sus productos, y qué licencias emplean.

El estudio partió con 120 empresas, de las que se eliminaron aquellas que no se consideraron basadas en FLOSS (*free, libre and open source software*), como las que permitían acceso al código sólo a usuarios no comerciales o que no permitían la redistribución. También se eliminaron empresas que, a pesar de prestar contribuciones importantes a proyectos de software libre, no basan su modelo de negocio principal en él (como IBM, HP y SUN).

#### Web recomendada

Para más información podéis consultar:  
<http://opensource.mit.edu/papers/OSSEMP07-daffara.pdf>

Se seleccionó un conjunto de aspectos caracterizadores, como licencia, productos y servicios ofrecidos (instalación, integración, formación, consultoría, certificaciones técnicas y legales), tipos de contratos (suscripciones, licencias, o por-incidencia), así como literatura auto-referencial ofrecida en sus páginas web, e información relativa a su relación con la comunidad. Finalmente, se recogieron los datos, y se eliminaron las variables no significativas, obteniendo las siguientes variables caracterizadoras:

- Principal generador de ingresos
  - Selección.
  - ITSC (*installation/training/support/consulting*). Se agrupan los distintos tipos de servicios, ya que el estudio encuentra que las empresas que ofrecen uno de ellos tienden a ofrecer también el resto.
  - Suscripciones.
  - Licencias.
  
- Modelo de licencia

Aplicando un análisis de conglomerados (análisis de *clusters*) a las empresas caracterizadas en términos de estas variables, el estudio obtiene seis modelos principales de negocio, y un séptimo grupo analizado por separado:

- 1) **Doble licencia:** Esquema de doble licencia GPL y propietaria para vender a quienes quieran desarrollar código cerrado basado en el producto libre.
- 2) **Productos separados OSS y comerciales:** Venta de productos comerciales basados en otro libre.
- 3) **"Badgeware":** Protección de marca, productos liberados deben mantener logo/autoría original visible.
- 4) **Especialistas de producto:** Creación de un producto libre y comercialización de servicios entorno a él.
- 5) **Proveedores de plataformas:** Servicios de selección, integración y soporte, proporcionando plataformas probadas y verificadas.
- 6) **Empresas de selección/consultoras:** Analistas y servicios genéricos que, en general no contribuyen a la comunidad, dado que los resultados del análisis y consultoría se mantienen privados.
- 7) **Mercados auxiliares:** Como ejemplo, SourceForge/OSTG recibe la mayoría de sus ingresos de las ventas de su sitio afiliado ThinkGeek. Aunque este modelo no es uno de los caracterizados por el estudio (el limitado número de casos de esta categoría no permite la extrapolación), no debe subestimarse, suponiendo un modelo de financiación importante.

A continuación, mostramos la tabla de resultados obtenida del estudio.

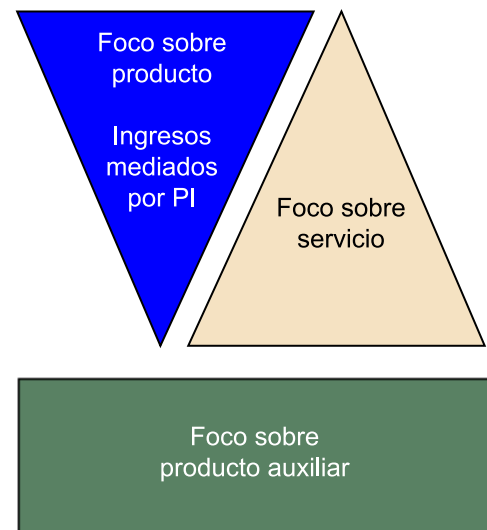
|                                   | Company                 | Main Licensing model |                             |           |          | Main revenue generation   |           |      |              |
|-----------------------------------|-------------------------|----------------------|-----------------------------|-----------|----------|---------------------------|-----------|------|--------------|
|                                   |                         | twin licensing       | OSS and commercial versions | Badgeware | Pure OSS | multiple packages covered | selection | ITSC | Subscription |
| twin lic.                         | Funambol                | •                    |                             |           |          |                           | •         |      | •            |
|                                   | Lustre                  | •                    |                             |           |          |                           | •         |      |              |
|                                   | MuleSource              | •                    |                             |           |          |                           |           | •    | •            |
|                                   | Mysql                   | •                    |                             |           |          |                           |           | •    | •            |
|                                   | OpenClovis              | •                    |                             |           |          |                           |           | •    |              |
|                                   | Pentaho                 | •                    |                             |           |          |                           |           | •    | •            |
| Split OSS/<br>commercial releases | sleepycatdb             | •                    |                             |           |          |                           |           |      | •            |
|                                   | Adaptive Planning       |                      | •                           |           |          |                           |           |      | •            |
|                                   | Alterpoint              |                      | •                           |           |          |                           | •         |      | •            |
|                                   | Altinity                |                      | •                           |           |          |                           | •         |      | •            |
|                                   | Codeweaver (WINE)       |                      | •                           |           |          |                           |           |      | •            |
|                                   | Coupa                   |                      | •                           |           |          |                           |           |      | •            |
|                                   | Digium (Asterisk)       |                      | •                           |           |          |                           |           | •    |              |
|                                   | Enomalism               |                      | •                           |           |          |                           |           |      | •            |
|                                   | EnterpriseDB            |                      | •                           |           |          |                           |           |      | •            |
|                                   | GreenPlum               |                      | •                           |           |          |                           |           |      | •            |
|                                   | GroundWork              |                      | •                           |           |          |                           |           | •    |              |
|                                   | Hyperic                 |                      | •                           |           |          |                           |           | •    |              |
|                                   | JasperSoft              |                      | •                           |           |          |                           |           |      | •            |
|                                   | Knowledge Tree          |                      | •                           | •         |          |                           |           |      |              |
|                                   | OpenCountry             |                      | •                           |           |          |                           |           |      | •            |
|                                   | Open-Xchange            |                      | •                           |           |          |                           |           |      |              |
|                                   | NoMachine NX            |                      | •                           |           |          |                           |           |      | •            |
|                                   | rPath                   |                      | •                           |           |          |                           |           | •    |              |
|                                   | Scalix                  |                      | •                           |           |          |                           |           |      | •            |
|                                   | Sendmail                |                      | •                           |           |          |                           |           |      | •            |
|                                   | Smoothwall              |                      | •                           |           |          |                           |           | •    |              |
|                                   | Sourcefire (SNORT)      |                      | •                           |           |          |                           |           | •    |              |
|                                   | Splunk                  |                      | •                           |           |          |                           |           | •    |              |
|                                   | SSExplorer              |                      | •                           |           |          |                           |           |      | •            |
|                                   | SugarCRM                |                      | •                           | •         |          |                           |           |      | •            |
|                                   | TenderSystem            |                      | •                           | •         |          |                           |           |      | •            |
|                                   | VirtualBox              |                      | •                           |           |          |                           |           |      | •            |
| Vyatta                            |                         | •                    |                             |           |          |                           | •         | •    |              |
| XenSource(Xen)                    |                         | •                    |                             |           |          |                           | •         |      |              |
| Zen(PHP)                          |                         | •                    |                             |           |          |                           |           | •    |              |
| Zimbra                            |                         | •                    |                             |           |          |                           |           | •    |              |
| Badgeware                         | 1bizcom                 |                      |                             | •         |          |                           | •         |      |              |
|                                   | CATS applicant tracking |                      |                             | •         |          |                           |           | •    |              |
|                                   | EmuSoftware/Netdirector |                      |                             | •         |          |                           | •         | •    |              |
|                                   | Jbilling                |                      |                             | •         |          |                           | •         |      |              |
|                                   | OpenBravo               |                      |                             | •         |          |                           | •         |      |              |
|                                   | OpenEMM                 |                      |                             | •         |          |                           | •         |      |              |
|                                   | Open Terracotta         |                      |                             | •         |          |                           |           | •    |              |
|                                   | SocialText              |                      |                             | •         |          |                           |           |      | •            |
| product specialists               | Alfresco                |                      |                             |           | •        |                           | •         | •    |              |
|                                   | Babel                   |                      |                             |           | •        |                           | •         |      |              |
|                                   | CentraView              |                      |                             |           | •        |                           | •         |      |              |
|                                   | CleverSafe              |                      |                             |           | •        |                           | •         |      |              |
|                                   | Compiere                |                      |                             |           | •        |                           | •         | •    |              |
|                                   | Exadel                  |                      |                             |           | •        |                           | •         |      |              |
|                                   | Jitterbit               |                      |                             |           | •        |                           | •         | •    |              |
|                                   | Mergere                 |                      |                             |           | •        |                           | •         |      |              |
|                                   | Mindquarry              |                      |                             |           | •        |                           | •         |      |              |
|                                   | Mirth                   |                      |                             |           | •        |                           | •         |      |              |
|                                   | OBIZ                    |                      |                             |           | •        |                           | •         |      |              |
|                                   | Qlusters (OpenQRM)      |                      |                             |           | •        |                           | •         |      |              |
|                                   | Symbiot/OpenSIMS        |                      |                             |           | •        |                           | •         |      |              |
|                                   | Talend                  |                      |                             |           | •        |                           | •         |      |              |
|                                   | UltimateEMR             |                      |                             |           | •        |                           |           | •    |              |
| VISTA                             |                         |                      |                             | •         |          | •                         |           |      |              |
| vTiger                            |                         |                      |                             | •         |          | •                         |           |      |              |
| Zenos s                           |                         |                      |                             | •         |          |                           | •         |      |              |
| Platt. Prov.                      | Jbos s                  |                      |                             |           | •        | •                         | •         | •    |              |
|                                   | RedHat Linux            |                      |                             |           | •        | •                         | •         | •    |              |
|                                   | Sourcelabs              |                      |                             |           | •        | •                         | •         | •    |              |
|                                   | SpikeSource             |                      |                             |           | •        | •                         | •         | •    |              |
|                                   | SUS Linux               |                      |                             |           | •        | •                         | •         | •    |              |
|                                   | WSO2                    |                      |                             |           | •        | •                         | •         | •    |              |
| selection/consu-<br>lting         | ayamon                  |                      |                             |           |          | •                         | •         |      |              |
|                                   | Enomaly                 |                      |                             |           |          | •                         | •         |      |              |
|                                   | navica                  |                      |                             |           |          | •                         | •         |      |              |
|                                   | openlogic               |                      |                             |           |          | •                         | •         |      |              |
|                                   | Optaros                 |                      |                             |           | •        | •                         | •         |      |              |
| Other                             | x-tend                  |                      |                             |           |          | •                         | •         |      |              |
|                                   | CiviCRM                 |                      |                             |           | •        |                           |           |      |              |
|                                   | Eclipse                 |                      |                             |           | •        |                           |           |      |              |
|                                   | Mozilla                 |                      |                             |           | •        |                           |           |      |              |
|                                   | OSAF Chandler           |                      |                             |           | •        |                           |           |      |              |
| Sourceforge                       |                         |                      |                             | •         |          |                           |           |      |              |

Modelos de negocio en empresas *FLOSS* (Carlo Daffara, "Business models in FLOSS-based companies <http://opensource.mit.edu/papers/OSSEMP07-daffara.pdf>)

## 2.4. Propuesta de clasificación

La última clasificación analizada resulta interesante al proporcionarnos datos empíricos sobre empresas reales que están centrando, en la actualidad, su modelo de negocio en torno al software libre. Sin embargo, al igual que Hecker, Daffara propone una caracterización aislada, más que una taxonomía. A continuación propondremos un esquema propio para ordenar e incorporar las ideas que hemos analizado hasta ahora, clasificando los modelos según el grado en el que sus ingresos se derivan de la propiedad intelectual sobre ese software, y según su foco esté centrado en proporcionar productos o servicios:

- Especialistas/Verticales  
(Empresas desarrolladoras con un programa libre como producto principal)
  - Mixtas OSS/propietarias: Dobles licencias
  - Mixtas OSS/propietarias: Núcleo libre, accesorios propietarios
  - Puras OSS: "Venta distribuida" de producto libre
  - Puras OSS: Servicios sobre el producto
  - Software como servicio (SaaS)
- Prestadoras de servicios asociados al software
  - Empresas distribuidoras de plataformas
  - Grandes integradoras
  - Pymes y microempresas de nicho
- Mercados auxiliares
  - Hardware
  - Otros



Nuestra clasificación, como la de otros autores, se basa en la fuente de ingresos. Sin embargo, además de considerar cómo recuperan la inversión en el desarrollo de software libre distintas empresas, también es importante examinar cómo aprovechan las ventajas un modelo de desarrollo libre puede aportar.

Un modelo de negocio se caracteriza, además de por su fuente de ingresos, por el mercado hacia el que se orienta, cómo elabora y comercializa sus productos, y cómo se relaciona con la competencia. En este sentido, hay un aspecto transversal a cualquier modelo de negocio que cobra especial relevancia gracias al uso del software libre: el concepto de **coopetencia** (*coopetition*).

### Coopetencia

Entre otros aspectos diferenciadores frente al software propietario, el empleo de software libre podrá aumentar la calidad de los servicios prestados, contribuyendo a la eliminación de las barreras de entrada, y dibujando un escenario de mayor competitividad y esfuerzo por la diferenciación y la especialización; así como de una competitividad distinta, abierta, cooperativa, en la que las empresas tendrán que cooperar, además de competir, si quieren prosperar. Es-

te concepto empresarial, que en cierta medida está sustituyendo al de "el ganador se lleva todo" en el contexto de una nueva economía de red, se llama *coopetencia* (*coopetition*).

**Coopetencia:** Cooperación entre empresas competidoras para buscar escenarios ganador-ganador, bien para incrementar el valor del producto, bien para incrementar el mercado.

En este contexto, las empresas deben examinar cuidadosamente su **ecosistema económico** –clientes, proveedores, competidores y complementados– poniendo en marcha estrategias para la creación de nuevas alianzas, y repensando las asociaciones tradicionales.

Este concepto no es exclusivo del software libre, y está extendido a otras áreas. Empresas en una misma industria pueden colaborar entre ellas para ampliar sus mercados, compitiendo más tarde a la hora de segmentarlos.

### Ejemplo

**Intel** invertirá cantidades importantes en ampliar el mercado de los microprocesadores, a pesar de que una parte de esa inversión beneficiará directamente a sus competidores, AMD. En este caso, dada la dominancia de Intel, el porcentaje de su inversión que beneficiará a otros será bastante bajo.

Aunque la coopetencia no sea de aplicación exclusiva en el campo del software libre, sin duda cobra un significado especial en escenarios de desarrollo de código abierto. El hecho de que la competencia se beneficie de la inversión propia es inevitable, por lo que se hace necesario buscar maneras de convertir esta aparente desventaja en una ventaja empresarial. Por otro lado, incorporar a los usuarios (clientes) dentro del proceso de desarrollo, involucrándoles de manera participativa como aliados, es también una característica del modelo de desarrollo del software libre.

El uso de software libre también limitará en gran medida la posibilidad de establecerse como monopolio, además de ofrecer una garantía anti-cautividad. De nuevo, una pregunta clave para cualquier empresa se hace especialmente relevante en un escenario de software libre: ¿cómo crear valor para un cliente extrayendo al mismo tiempo parte de ese valor para la empresa?

### Web recomendada

Para más información podéis consultar:

Henry Chesbrough; Wim Vanhaverbeke; Joel West.  
"Open Innovation: researching a new paradigm"

<http://www.openinnovation.net/Book/NewParadigm/Chapters/index.html>

### 3. Modelos de negocio con software libre

En este apartado examinaremos cada modelo de negocio, viendo ejemplos concretos. Es importante tener en cuenta que no se trata de modelos estancos, sino más bien de un continuo difuso. Muchas de las empresas que mencionaremos combinan varios de los modelos, aunque las etiquetemos para sistematizar su estudio.

#### 3.1. Especialistas/Verticales (una aplicación libre como principal producto)

En este apartado incluimos empresas que están produciendo software libre, siendo las promotoras y/o líderes de proyectos concretos. Su implicación con el software libre es, por lo tanto, muy importante, y uno de los aspectos clave en su estrategia empresarial será el manejo de la comunidad, y poder aprovechar las posibilidades de innovación, difusión y trabajo voluntario que ofrece. En esencia, estos modelos mantienen un producto libre para la comunidad, y un producto o servicio relacionado como oferta comercial, y su clave de éxito a menudo estará en conseguir mantener ambas facetas en equilibrio. Según Marten Mickos, consejero delegado de MySQL AB:

"Las empresas FOSS no funcionarán a no ser que sirvan por igual a aquellos que quieren gastar tiempo para ahorrar dinero, como a aquellos que gasten dinero para ahorrar tiempo".

Este tipo de empresas son las más numerosas en el estudio de Daffara, incluyendo las cuatro primeras categorías (dobles licencias, versiones OSS/proprietarias, *badgeware*, y especialistas de producto). Equivaldrían a las empresas de productos vistas en el módulo 3 de esta asignatura, por lo que su problema principal será cómo recuperar la inversión inicial dedicada al desarrollo.

Como hemos visto en las anteriores clasificaciones, una estrategia común es la de obtener ingresos a través de licencias propietarias, que de diferentes maneras se combinan con licencias libres.

También podemos incluir los **modelos que combinan las licencias propietarias cíclicamente**, como los *loss leader* y *sell it, free it* de Hecker. El concepto de *loss leader* no es exclusivo del software, y es una estrategia de amplia difusión en cualquier sector de actividad: se ofrece un producto de manera gratuita, o a un precio tan bajo que supone pérdidas para la empresa proveedora, como forma de atraer la atención de un gran número de clientes potenciales a los que se les pretende vender otros artículos. En este sentido, tanto los modelos de dobles licencias, como los de producto libre más extensiones propietarias, usan en cierta medida una estrategia de *loss leader*.



Además de fomentar la venta del producto relacionado, en el campo del software una estrategia de código abierto de este tipo traerá varias ventajas, como contribuir a establecer su tecnología como estándar *de facto*, atraer mejoras y complementos que hagan al producto más atractivo, generar simpatía entre una audiencia que puede incluir a los clientes potenciales del producto relacionado, y reducir los costes de mantenimiento del proyecto.

A continuación trataremos en detalle el modelo de dobles licencias y el de producto principal libre con accesorios propietarios. Omitimos otros modelos en los que el producto principal no es libre, porque en realidad representan modelos de negocio basados en software propietario: la liberación de código sólo supone una estrategia de negocio complementaria para mejorar la posición de su producto principal propietario.

Daffara también nos muestra un número importante de empresas que lideran proyectos de desarrollo con licencias puramente libres, y que obtienen sus ingresos del ITCS (*installation/training/support/consulting*). Quizás este grupo es uno de los que más modelos distintos puede englobar, estando sus ingresos enmarcados en una categoría bastante vaga. En este sentido, será importante examinar con más cuidado a qué mercados se dirigen y qué diferenciación presentan ante otros productos equivalentes, aparte de la de mejor conocimiento.

### 3.1.1. Modelos mixtos: licencias dobles

Este modelo se basa en la distribución de un producto bajo dos licencias distintas: una licencia propietaria tradicional, y una licencia libre restrictiva (tipo GPL). De esta manera, si alguien quiere generar un trabajo derivado, y redistribuirlo sin el código, puede hacerlo, pero deberá pagar una licencia. De lo contrario, todos los trabajos derivados deben redistribuirse con el código.

Michael Olsen, gerente de Sleepycat Software Inc., productores de BerkeleyDB, describe su modelo de doble licencia de la siguiente manera:

"La licencia *open source* de Sleepycat permite el uso de Berkeley DB [...] sin coste, bajo la condición de que si se usa el software en una aplicación que más tarde se redistribuya, el código completo de la aplicación debe estar disponible, y debe poder ser redistribuido de nuevo libremente bajo condiciones razonables. Si no se quiere ofrecer el código fuente de una aplicación derivada, se puede comprar una licencia de Sleepycat Software."

S. Comino; F. M. Manetti. "Dual licensing in open source markets". Disponible en: [http://opensource.mit.edu/papers/dual\\_lic.pdf](http://opensource.mit.edu/papers/dual_lic.pdf)

Esta estrategia resulta apropiada cuando una parte relevante de la demanda se genera por usuarios comerciales que necesitan el software para embeberlo en sus propios productos. Estos clientes usan el producto comprado como un *input* para producir otro software, bien como un producto final, o como parte de una tecnología más compleja producida y vendida por este cliente comercial.

Sea para poder vender sus productos derivados bajo un esquema tradicional propietario, o porque el software que genera es parte fundamental de su diferenciación, este cliente necesitará poder cerrar el código que genere, y pagará por ello.

Estos modelos segmentan a sus usuarios en dos grupos: la comunidad –todos aquellos usuarios que estén satisfechos con licencias libres, y usen el producto bajo esos términos– y clientes corporativos sensibles a los términos de reciprocidad de licencias libres.

Sin embargo, el mantenimiento de una comunidad de personas colaboradoras en torno al producto puede resultar problemático. Por un lado, la obtención de ingresos directos por el producto puede influir en la motivación de voluntarios que contribuyen sin recibir nada a cambio. Por otra parte, las compañías que lo implementan deben recoger de manera formal de sus voluntarios la asignación de *copyrights*, para evitar problemas en el futuro de colaboradores descontentos reclamando su parte de ingresos en concepto de licencias por el producto que contribuyeron a desarrollar.

En la práctica, las empresas que basan su modelo en la doble licencia no se benefician extensamente de las ventajas de conseguir aportaciones externas al desarrollo, consiguiendo sólo resolución de errores a pequeña escala y algún parche por parte de la comunidad. El equipo principal de desarrollo típicamente estará dominado, casi al 100%, por empleados de la compañía.

Otro problema que pueden encontrar estos modelos es que sus clientes puedan construir sus extensiones propietarias, sin necesidad de modificar el código original, por lo que podrán usar la versión con la licencia libre, y mantener sus adiciones como una aplicación separada e independiente.

A menudo, estas empresas combinan los ingresos generados por las dobles licencias con otras actividades, como la prestación de servicios que veremos en apartados posteriores. Ejemplos de este modelo son Funambol, MySQL, Sleepycat DB, y TrollTech/NOKIA.

### El caso de Funambol

|                                      |                               |
|--------------------------------------|-------------------------------|
| Nombre de la compañía                | Funambol, Inc.                |
| Sede                                 | Redwood City (Estados Unidos) |
| Página web                           | www.funambol.com              |
| Fecha de creación                    | 2001                          |
| Nº de personas empleadas en 2007     | 40                            |
| Volumen de ventas en 2007 (millones) | \$4,8                         |

Datos corporativos de Funambol, Inc. Tabla elaborada a partir de estadísticas de Hoovers (<http://www.hoovers.com>)

### Web recomendada

Para más información podéis ver:

<http://www.funambol.com/blog/capo/2006/07/my-honest-dual-licensing.html>

**Funambol** es una corporación norteamericana dedicada, según su lema, al "mobile 2.0 messaging powered by open source". La empresa desarrolla un servidor de aplicaciones móviles (proporciona *push e-mail*, libro de direcciones y calendario, sincronización de datos, y servidor de aplicaciones para dispositivos móviles y PC), además de una plataforma de desarrollo para aplicaciones móviles, ambas desarrolladas bajo el nombre de "Funambol".

Comercializa su base de código bajo dos licencias: la AGPLv3 para su "Community Edition", y una licencia propietaria comercial para su "Carrier Edition", pero combina esta estrategia con la de proporcionar funcionalidades adicionales necesaria para grandes implementaciones en la versión cerrada, además de servicios basados en la "Carrier Edition".

En este caso, Funambol ha elegido como licencia la "Affero" GPL, que le proporciona una protección extra frente a usos comerciales de sus aplicaciones en la modalidad de software como servicio (SaaS). Como vimos en el módulo tres de esta asignatura, la GPL permite la modificación del código sin su redistribución, siempre que la aplicación en sí tampoco sea redistribuida, como ocurre con la prestación del software como servicio. La "Affero" GPL resuelve el problema de este vacío, requiriendo la redistribución del código fuente también cuando la funcionalidad del software se ofrezca mediante el modelo "SaaS".

La naturaleza del software lo hace idóneo para el modelo de doble licencia, ya que previsiblemente resultará atractivo para otras empresas que quieran desarrollar aplicaciones cerradas sobre su plataforma, como operadores de telefonía móvil, fabricantes de dispositivos y otras compañías de software. Gracias al uso de la AGPL, aquellas empresas que usen Funambol como base de sus ofertas "SaaS" también deberán pagar si no quieren redistribuir el código. Entre sus clientes se encuentran Vodafone, Earthlink y Computer Associates.

Funambol intenta explotar al máximo las necesidades de grandes clientes corporativos con la incorporación de funcionalidad adicional en su versión comercial y la prestación de servicios. Para evitar los problemas derivados del modelo "núcleo libre + accesorios propietarios" que veremos más adelante, se asegura de que la funcionalidad cerrada sólo sea interesante en escenarios de grandes implementaciones corporativas, por lo que su comunidad de usuarios libres no sentirán la necesidad de desarrollar esa funcionalidad por su cuenta.

Fabrizio Capobianco, gerente de Funambol, justifica el modelo de doble licencia en "My Honest Dual Licensing" como el modelo más "honesto" de mantener los principios del desarrollo de software libre, compatibilizándolo con la necesidad empresarial de generar un beneficio.

Sin embargo, como hemos visto anteriormente, definir una fuente de ingresos viable no garantiza el éxito de ninguna empresa, y el uso de software libre permitirá poner en marcha estrategias cualitativamente distintas a las de un modelo basado en software propietario. Funambol es un caso muy ilustrativo en este sentido, en el que la empresa tuvo que redefinir tanto sus prácticas de marketing como la población objetivo de éstas, antes de tener un modelo de negocio viable.

En sus inicios, Funambol intentó construir un modelo clásico de vendedor de software en torno a su producto de software libre. La compañía había desarrollado Sync4j, que permitía a desarrolladores construir aplicaciones para dispositivos móviles bajo el paradigma "a veces conectado" (la aplicación puede funcionar desconectado, sincronizando los datos cuando recupera conexión). Identificó como sus clientes potenciales grandes compañías y operadores inalámbricos, que dado el gran número de trabajadores con los que cuentan, y aprovechando las crecientes posibilidades de movilidad, necesitarían sincronizar datos entre diversos dispositivos móviles y sus servidores corporativos.

Para hacer llegar su producto a estos clientes, Funambol decidió seguir una estrategia de ventas proactiva, poniendo un gran esfuerzo en su equipo de marketing y ventas, que trataba de acceder a ellos de forma directa, mediante campañas telefónicas.

Su éxito fue muy limitado. Funambol no consiguió cumplir las expectativas de ventas que tenía, encontrando que las grandes corporaciones eran reacias a tratar con una pequeña nueva empresa como ella. Por otro lado, los ciclos de venta se hacían muy largos, y pronto se hizo patente que para mantener esta estrategia se requería un equipo de ventas y marketing mucho más grande del que Funambol se podía permitir.

Funambol identificó pronto que sus problemas partían de esta estrategia de ventas activa, tradicional en el mundo del software propietario, pero que supone una barrera de entrada que pocos consiguen cruzar: para acceder a un conjunto de clientes potenciales compuesto por grandes corporaciones, a menudo hace falta tener una gran capacidad

de marketing y ventas, además de contar con un tamaño y reputación suficientes para transmitir la necesaria confianza.

El uso de software libre permitía revertir esta estrategia, centrándose en un marketing reactivo, en respuesta a la iniciativa del cliente. En este nuevo escenario, serían los clientes potenciales los que buscarían a Funambol, dejando a la compañía el papel de estar atenta para identificarles una vez éstos se ponían en contacto con ella.

La efectividad de esta estrategia sólo dependía de un factor: el número de descargas de su producto. Con un número de descargas suficiente, identificaron el siguiente ciclo de venta típico (mucho más corto que el que experimentaron con su anterior estrategia):

- 1) El usuario potencial accede a la web de Sync4j para buscar información sobre el producto y documentación técnica.
- 2) El usuario descarga el producto.
- 3) Más tarde, se suscribe a la lista de correo, buscando más información.
- 4) Tras un uso intensivo de su producto (normalmente en proyectos I+D), el cliente contacta con Funambol para preguntar por precios y condiciones de licencia. Internamente, se le clasifica como cliente potencial.
- 5) Finalmente, pide un presupuesto y oferta oficial, y puede convertirse en un Cliente Funambol.

(Fabrizio Capobianco; Alberto Onetti. "Open Source and Business Model Innovation. The Funambol case". Disponible en: <http://oss2005.case.unibz.it/Papers/4.pdf>)

El factor clave para seguir alimentando este ciclo es, como hemos dicho anteriormente, mantener el número de descargas del producto alto. El ciclo se retroalimenta, generando por sí mismo más descargas, por lo que tras un esfuerzo inicial, este mecanismo tomaría suficiente inercia para funcionar casi por sí solo.

Para lograrlo, Funambol se centró en **crear comunidad en torno a su producto**, focalizando sus esfuerzos de marketing sobre los usuarios de su versión libre, tanto los expertos como aquellos con menos habilidades técnicas. Aunque esta estrategia no se orienta hacia sus clientes generadores de ingresos de forma directa, resultó ser mucho más barata y eficiente.

La empresa se dedicó a dar a conocer el producto entre desarrolladores, participando en foros de desarrollo, en listas de correo, revistas especializadas, conferencias, creando alianzas con organizaciones sin ánimo de lucro de promoción del software libre, o creando sinergias con otros productos de código abierto bien establecidos. De cara a los usuarios más inexpertos, fue necesario asegurarse de que el producto fuera de fácil instalación, con suficiente documentación accesible desde su página web. La empresa vio cómo, al trabajar estos dos últimos factores, el número de descargas del producto se incrementaba notablemente, poniendo en marcha, de esta manera, su ciclo generador de ventas.

### 3.1.2. Modelos mixtos: núcleo del producto libre y accesorios propietarios

En este modelo ("Split OSS/commercial releases", según Daffara), existen dos versiones distintas de un programa, una versión básica libre, y una versión comercial propietaria, basada en la anterior, pero con funcionalidad adicional implementada a través de *plugins* o accesorios. La versión libre debe emplear una licencia de tipo MPL o BSD, que permita la combinación para crear un producto cerrado.

El problema principal de este modelo será mantener el producto libre suficientemente interesante, sin restarle valor al producto propietario generador de ingresos. También corre el riesgo de que la comunidad en torno al producto decida por su cuenta desarrollar la funcionalidad de la versión propietaria, haciendo difícil la generación de ingresos por su venta.

En este tipo de modelos, se distinguen dos clases de usuarios: aquellos que estarían dispuestos a pagar a cambio de obtener un producto con alguna funcionalidad adicional (medianas y grandes empresas), y aquellos muy sensibles al precio, como pequeñas empresas, microempresas o usuarios individuales. Al combinar versiones libres y propietarias, se consigue una mayor adopción de la solución propuesta, sin perder por ello la captura de ingresos a través de las versiones propietarias. Como hemos visto en módulos anteriores, en un escenario de "the winner takes it all" común en el mundo del software, la estrategia basada en una amplia adopción resulta de gran importancia.

En este sentido, parte de los mismos principios de segmentación de sus usuarios que el modelo de dobles licencias, pero corre mayor riesgo de perder la simpatía de la comunidad, que no tiene acceso a todo el código fuente.

Un ejemplo de este modelo es **Sendmail Inc.**, que comercializa una constelación de productos propietarios en torno al servidor libre sendmail. Otros ejemplos son Hyperic (IT Operations/Monitoring), SourceFire (SNORT comercial), Zimbra/Yahoo (mensajería, *groupware*), y XenSource/Citrix (virtualización).

### El Caso de Sendmail

|                                      |                                     |
|--------------------------------------|-------------------------------------|
| Nombre de la compañía                | Sendmail, Inc.                      |
| Sede                                 | Emeryville, CA.<br>(Estados Unidos) |
| Página web                           | www.sendmail.com                    |
| Fecha de creación                    | 1997                                |
| Nº de personas empleadas en 2007     | 125                                 |
| Volumen de ventas en 2007 (millones) | \$23                                |

Datos corporativos de Sendmail, Inc. Elaborada a partir de estadísticas de Hoovers (<http://www.hoovers.com>)

Al estudiar modelos de negocio basados en software libre, a menudo pensamos en corporaciones que deciden abrir el código como ventaja competitiva para ampliar su cuota de mercado. Sendmail resulta un caso interesante, en el que el proceso tiene lugar de forma inversa: con orígenes libres y sin ánimo de lucro, el establecimiento de una iniciativa comercial en torno al proyecto no sólo pretende obtener ingresos del desarrollo, sino mantener la posición dominante del proyecto en su sector, y ampliar su base de usuarios.

Sendmail es un agente de transferencia de correo ("mail transfer agent", MTA) y supone uno de los ejemplos más conocidos de proyectos nacidos en el seno de comunidades de software libre. En 1998, se estimaba que el 80% de todo el tráfico de correo electrónico se enviaba a través de Sendmail. Actualmente, continúa siendo el MTA más popular en Internet, aunque ha perdido parte de sus usuarios en Microsoft Exchange Server, Exim y

Postfix. Igualmente destacable es el largo periodo de vida que está teniendo el producto, y que tiene su origen en desarrollos iniciados en la década de los setenta.

Eric Allman desarrolló la primera versión de Sendmail a principios de la década de los ochenta, en el seno de la Universidad de Berkeley, a partir de trabajo previo en el programa Delivermail, y en 1997 fundó Sendmail, Inc. La estrategia de la empresa se centraba en vender funcionalidades adicionales relacionadas con Sendmail de manera propietaria (por ejemplo, interfaces más amigables), además de en proporcionar servicios complementarios. Al mismo tiempo, la compañía se esforzó por mantener la continuidad del desarrollo de Sendmail de forma abierta, proporcionando servicios de alojamiento y recursos humanos para el desarrollo.

Al crear la compañía, Allman esperaba no sólo desarrollar una actividad empresarial, sino proteger la posición dominante de Sendmail, que estaba en peligro por la aparición de formatos propietarios que amenazaban el estándar abierto SMTP. La empresa orientó sus esfuerzos hacia el entorno corporativo, ofreciendo no sólo servicios de integración y soporte, sino un producto que se ajustara más a sus necesidades. Las extensiones creadas por la compañía se centran en ofrecer interfaces gráficas y mayor facilidad de gestión, y se comercializan de manera propietaria.

"Sendmail, Inc. develops commercial products and services for ISPs and enterprises for whom email is mission critical, while continuing to drive innovation and standards through Open Source software development."

Sendmail, Inc

Podemos considerar que la creación de Sendmail, Inc. fue el paso necesario para cruzar el "abismo" (*the chasm*), y lograr la adopción del producto por las mayorías pragmáticas y conservadoras. Sin embargo, para Allman era importante mantener la funcionalidad original de Sendmail libre, para lo que se creó Sendmail Consortium, entidad sin ánimo de lucro responsable del desarrollo de la versión libre. De esta manera, capitaliza las ventajas de un modelo de desarrollo libre, tanto en lo que a contribuciones y reducción de gastos se refiere, como a innovación y evolución del producto.

Allman aprovechó así "el abismo" para poder vender de forma propietaria extensiones a su producto, sin correr el peligro de la ramificación de su proyecto. Siguiendo el modelo de Moore, la comunidad en torno al proyecto libre Sendmail estaría formada por los innovadores y los entusiastas tecnológicos, interesados en la funcionalidad en bruto, y en las propuestas nuevas. Los clientes comerciales, sin embargo, serían los pragmáticos y conservadores, con necesidades y objetivos muy distintos. Las extensiones propietarias, centrándose en las funcionalidades de empaque y acabado del producto (facilidad de uso, interfaces gráficas, estabilidad, etc.), no sólo no resultan interesantes para los innovadores, sino que les pueden llegar a parecer innecesarias. La presencia de ese abismo entre los intereses de la comunidad y los clientes comerciales permite la co-existencia de la versión central libre, y la extendida propietaria, sin peligro de bifurcaciones (*forks*), ya que la comunidad no tendrá interés en las extensiones del otro lado del abismo.

### 3.1.3. Modelos libres: "Venta distribuida" del producto

Es habitual asumir que al licenciar de una manera libre un producto se pierde la oportunidad de obtener ingresos directos vinculados a la propiedad intelectual sobre él, haciéndose necesario explotar otros productos o servicios complementarios.

Sin embargo, elegir una licencia libre para un trabajo no tiene por qué significar la renuncia a la obtención de ingresos relacionados de forma directa con ese producto. La idea de que nadie pagará por algo que eventualmente puede conseguir de forma gratuita, aunque extendida, no es fiel a la realidad. Mucha gente está dispuesta a pagar una cantidad pequeña por un trabajo que valora si cree que ese dinero irá a las autoras originales. Si un proyecto es suficientemente exitoso, puede recibir pequeñas aportaciones de mucha gente, llegando quizás a financiar su creación, de la misma manera que un artista callejero, sin cobrar entrada, puede recaudar lo suficiente como para hacer rentable su in-

#### Web recomendada

<http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/673/583>

versión de tiempo y esfuerzo. Esta es la idea que subyace en el *The Street Performer Protocol and Digital Copyrights*, de John Kelsey y Bruce Schneier al proponer un mecanismo de financiación distribuida de trabajos digitales, en el que la autor no realizaría su obra hasta haber recaudado financiación suficiente.

Distintos mecanismos para articular esta financiación directa y distribuida se han descrito y puesto en marcha en el contexto del desarrollo de software, desde donaciones y recompensas (*bounties*), a la creación de mercados vía web, para poner en contacto a desarrolladores y clientes potenciales, siguiendo un esquema de bonos similar al descrito por Chris Rasch en *The Wall Street Performer Protocol*.

Las donaciones son el mecanismo más sencillo de este tipo de financiación, pero demasiado inciertas para los creadores, que necesitan la seguridad de un ingreso antes de invertir su tiempo. En los sistemas de bonos y recompensas, las personas interesadas en una funcionalidad concreta ofrecen una recompensa por que sea implementada. Una vez la recompensa total (a la que pueden contribuir diversas personas) es suficiente para un desarrollador, puede ofrecerse a realizarla, cobrando una vez se haya completado. Algunos de estos sistemas se basan en la confianza entre el equipo de desarrollo y los usuarios, sin exigir garantías de pago, otros proponen el establecimiento de algún tipo de intermediario neutral.

La clave del éxito en estos escenarios puede encontrarse más en las facilidades de pago ofrecidas que en la disposición de pagar de los usuarios:

"La mayoría de las personas estarán dispuestas a pagar una cantidad pequeña sobre una mayor, si ya han sacado su cartera, y si piensan que es por una buena razón. Cuando las personas dejan de hacer pequeñas donaciones voluntarias a una causa que les gusta, suele ser más por las molestias (escribir un cheque, mandarlo por correo, etc.), que por el dinero en sí."

(Karl Fogel. "The Promise of a Post-Copyright World". Disponible en: <http://www.questioncopyright.org/promise>)

Aunque muchos proyectos ponen en práctica estas ideas para obtener financiación complementaria, es difícil encontrar escenarios corporativos en los que el grueso de sus ingresos se obtenga a partir de estos mecanismos.

Por un lado, en el contexto del software este tipo de financiación se puede hacer más difícil al no existir una identificación y simpatía con las personas autoras tan fuerte como la que se deriva de otros trabajos creativos.

Por otro lado, este modelo seguramente tendrá más éxito si se trata de un proyecto de software libre sin ánimo de lucro e integrado en su totalidad por voluntarios, que despertará más fácilmente las simpatías de sus usuarios. Una empresa que quiera utilizarlo con éxito, seguramente tendrá que buscar un

#### Web recomendada

[http://www.firstmonday.org/issues/issue6\\_6/rasch/index.html](http://www.firstmonday.org/issues/issue6_6/rasch/index.html)

#### El servidor Cherokee

Este servidor decidió implementar un sistema de recompensas con el objetivo principal de atraer a nuevos desarrolladores al proyecto. Además de recompensar el esfuerzo, el ofrecer un retorno económico atraería a más personas a la comunidad de desarrollo, haciendo que el proyecto creciera.

#### Mercados virtuales

Varios intentos de creación de "mercados virtuales" de software basados en este tipo de financiación se han puesto en marcha. Algunos de los que están en funcionamiento en la actualidad son BountyCounty (<http://bountycounty.org/>), MicroPledge (<http://micropledge.com/>) y BountySource (<https://www.bountysource.com/>).

reconocimiento previo, mediante transparencia y confianza, mostrando que su ánimo de lucro no es a costa de todo y revertirá en el bien común (más adelante veremos modelos de negocio basados en estos principios).

Estos esquemas presentan un modelo económico más directo, se eliminan intermediarios, y se proporciona mayor cercanía entre usuarios y desarrolladores. Desde cierto punto de vista, podrían considerarse como la manera natural de financiación de un proyecto de software libre: de la misma manera que voluntarias contribuyen en distinta medida y en distintos aspectos del ciclo de desarrollo del software, los usuarios pueden formar parte del propio proyecto contribuyendo con un aporte económico de acuerdo a sus posibilidades e intereses.

### **3.1.4. Producto libre más servicios asociados**

Las empresas de esta categoría implementan una estrategia del tipo "mejor conocimiento" y "mejor código", desarrollando un producto libre, y ofreciendo servicios sobre él como motor de ingresos.

En este apartado incluimos tanto los especialistas de producto como los *badgeware* del estudio de Daffara, ya que ambos representan el mismo modelo de negocio. Por otro lado, aunque las licencias de tipo *badgeware* incluyen una restricción adicional de atribución, mantienen las características esenciales de apertura y libertad de conocimiento, pudiendo generar las mismas ventajas a través de sus comunidades de desarrollo que aquellos que empleen licencias sin esta restricción. Probablemente aquellos ejemplos enmarcados en *badgeware* deseen poner en marcha también cierta estrategia de marca, por lo que depositan una importancia especial en la atribución a la hora de redistribuir los productos que generan.

Este modelo tiene varios problemas, como pocas barreras de entrada en el negocio –cualquier empresa puede adquirir conocimiento sobre el producto y ofrecer servicios– o los problemas a la hora de conseguir contratos de soporte –las empresas clientes pueden preferir continuar con sus empresas de servicios o consultoría habituales, o contratar empresas proveedoras que ofrezcan soporte sobre toda su infraestructura de nuevas tecnologías, y no sólo frente a un producto concreto.

Otro problema al que se suelen enfrentar estos modelos para generar ingresos a partir de servicios es el de los innovadores y entusiastas: cuando un nuevo producto entra en el mercado, los primeros usuarios suelen ser personas con competencias técnicas, que no contratarán servicios de soporte sobre él, y preferirán adquirir el conocimiento necesario por sí mismas. Este modelo, por lo tanto, necesitará ofrecer un producto extendido y que transmita fiabilidad, para alcanzar un mercado potencial capaz de pagar por obtener servicios sobre el producto.



El éxito de este tipo de modelos de negocio es cuestionado por algunos autores (por ejemplo, Perens), sin embargo, también hay un número importante de empresas que se asientan sobre este modelo, habiendo atraído cuantías importantes de capital riesgo. Para llevar a cabo un modelo de negocio sostenible, sin embargo, tendrían que acometer los problemas mencionados anteriormente.

Entre los modelos de especialistas verticales prestadores de servicios, encontramos a Alfresco (gestión de contenidos), Compiere (ERP, CRM), vTiger, y Openbravo.

### El caso de Openbravo

|   |                   |
|---|-------------------|
| <b>Nombre de la compañía</b>            | Openbravo, S. L.  |
| <b>Sede</b>                             | Pamplona (España) |
| <b>Página web</b>                       | www.openbravo.com |
| <b>Fecha de creación</b>                | 2001              |
| <b>Nº de personas empleadas en 2007</b> | 26 a 50           |
| <b>Volumen de negocio en 2007</b>       | Hasta 300.000 €   |

Datos corporativos de Openbravo. (Obtenidos de <http://www.camerdata.es>)

**Openbravo** supone un ejemplo interesante de este tipo de modelo. La empresa, fundada en 2001, se centra en el desarrollo de dos aplicaciones libres para PYMES- OpenbravoERP (planificación de recursos empresariales), y OpenbravoPOS (punto de venta), que tratan de satisfacer las necesidades de gestión y planificación, y de terminal de punto de venta de pequeñas y medianas empresas, respectivamente. El código fue publicado en el año 2006, y actualmente está entre los proyectos más activos en el ranking de Sourceforge.

La empresa ha atraído cantidades importantes de capital riesgo, contando como inversores con Amadeus, Gimv, Adara, y SODENA (Sociedad de Desarrollo de Navarra), que ha invertido 5 millones de euros en la compañía.

Su estrategia empresarial está orientada a convertirse en producto líder en el sector, intentando que OpenbravoERP sea el software gestión de referencia entre las pymes. Para lograrlo, la empresa está explotando al máximo las posibilidades que le brinda el software libre, mediante un cuidado manejo de la comunidad, y la aplicación del concepto de cooepetencia.

De una manera similar a la que hemos visto en el caso de Funambol, Openbravo vio que por sí misma no tenía capacidad para difundir y distribuir su producto entre sus usuarios potenciales. Aunque OpenbravoERP y OpenbravoPOS no se dirigen a grandes corporaciones, sino a pymes, para lograr sus objetivos estratégicos de convertirse en líder en el sector, el producto debía llegar a innumerables pequeñas y medianas empresas en todo el mundo.

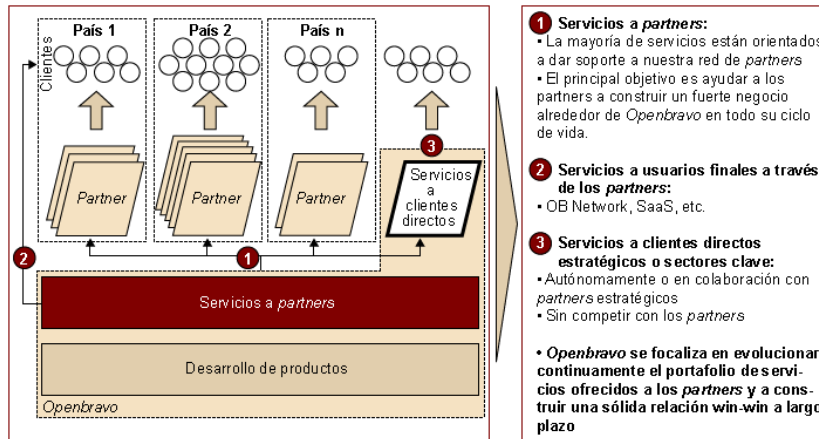
Además del tamaño necesario para llevar a cabo una campaña de marketing de esta escala, la empresa también era consciente de las dificultades que podía encontrarse a la hora de competir por prestar servicios directamente a sus usuarios finales, que podrían preferir a empresas locales, o que ofrecieran soluciones integrales, y no sólo frente a un producto.

Para solucionar estas barreras, Openbravo posicionó a estas empresas no como competidoras, sino como colaboradoras. De esta manera, admite que no porque sean los desarrolladores del producto, hayan de ser los idóneos a la hora de prestar servicios sobre él a los usuarios finales. Su misión sería la de crear un buen producto que pudiera ampliar mercados, generando nuevas oportunidades de ingresos para empresas de servicios informáticos, que podrían completar su oferta con OpenbravoERP y OpenbravoPOS.

De esta manera, Openbravo define como motor de ingresos la prestación de servicios a otras empresas de servicios informáticos, intermediarias entre ella y los usuarios finales.

Estas empresas constituirán una red de *partners*, que llevarán a cabo las tareas de implementación de OpenbravoERP y OpenbravoPOS en las pymes.

Openbravo ofrece a sus *partners* diversos servicios (soporte, formación), poniendo en marcha un esquema de consultoría piramidal similar al descrito en el módulo 3 de esta asignatura, además de transferirles fiabilidad y confianza. Al estar respaldadas por los desarrolladores del producto, pueden explotar la estrategia de "mejor conocimiento", y "mejor código" en sus mercados.



Openbravo: Oportunidad de negocio y vías de crecimiento (obtenida de la presentación de Openbravo en WhyFLOSS Madrid 2008 "Openbravo: keys to success in free software application development". <http://www.whyfloss.com/es/conference/madrid08/getpdf/49>).

Openbravo establece así una estrategia de *coopetencia*, dejando a empresas de servicios la posibilidad de explotar Openbravo en el contexto de sus mercados naturales, pero beneficiándose de una mayor difusión de su producto, y obteniendo ingresos directamente de sus *partners*. Hasta el momento, ha tenido bastante éxito con esta estrategia, contando en la actualidad con 85 *partners* en todo el mundo.

### 3.1.5. Software como servicio

Las empresas desarrolladoras de un producto también podrán explotarlo mediante el paradigma de software como servicio. En lugar de ofrecer servicios de instalación y soporte, la empresa se hace cargo de toda la infraestructura de hardware y software, ofreciendo directamente la funcionalidad a través de Internet. Los ingresos generados, de naturaleza recurrente, tomarían la forma de suscripciones al servicio.

#### Collabnet: software como servicio

Un buen ejemplo de este tipo de modelo lo presenta **Collabnet**. Ofrece servicios para el desarrollo colaborativo de software (control de versiones, seguimiento de incidencias, comunicación, etc.), que genera gracias a, entre otros, la plataforma de control de versiones Subversion. En este caso, además de mantener el código abierto, la empresa dedica esfuerzos importantes al mantenimiento de la comunidad, de manera que su trabajo sobre el proyecto no es más que una aportación (aunque sea grande) dentro de una comunidad libre. Otros ejemplos de empresas que comercializan sus productos siguiendo el modelo de "software como servicio" son sugarCRM, SocialText y JasperSoft.

Estas empresas, en la modalidad de "software como servicio", no encontrarán dificultades adicionales respecto a sus equivalentes propietarias en la generación de ingresos, ya que en este caso las ventas no se derivan de derechos de *copyright* sobre el producto. El que un cliente pueda descargar, instalar, configurar, alojar y mantener la aplicación supondrá más una herramienta de mar-

keting y difusión que una pérdida de ingresos. Como hemos visto anteriormente, el cliente corporativo estará dispuesto a pagar una cantidad a cambio de ver sus problemas resueltos.

Sin embargo, liberar todo el código presentará problemas en cuanto a diferenciación, y posibilidad de entrada a competidoras. Cualquier empresa con capacidad técnica e infraestructura suficiente podrá ofrecer un servicio similar, a partir de la disponibilidad del código. Ante este problema, la empresa que ha desarrollado el producto podrá basar su diferenciación en el "mejor conocimiento", y "mejor código", además de poder ganar las simpatías de la comunidad. Por otra parte, ante empresas competidoras que decidan contribuir también al desarrollo, se podrán poner en marcha mecanismos de *cooperencia*, colaborando en la ampliación del mercado, y segmentándolo más tarde mediante la especialización.

De forma análoga a las estrategias mixtas OSS/propietarias que hemos visto anteriormente, algunas empresas de esta categoría pondrán en marcha soluciones que incorporen alguna limitación sobre su código, principalmente manteniendo una pequeña parte del código cerrada, sobre la que basarán su diferenciación.

### 3.2. Servicios asociados al software libre

Considerando los servicios asociados al software libre, son posibles multitud de negocios ya que, en general, cualquier modelo de servicios basado en software propietario (como los que vimos en el módulo 3) será extrapolable de una forma bastante directa al software libre. Todos los pasos descritos en la cadena de creación e implementación de una solución tecnológica seguirán siendo viables en un contexto de aplicaciones abiertas. Sin embargo, el empleo de software libre amplía las posibilidades y los factores de diferenciación de modelos de negocio centrados en servicios.

Como principio básico diferenciador, podemos destacar la **ausencia de costes derivados de licencias**, que proporciona una clara ventaja competitiva frente a soluciones propietarias. Aun así, para aprovechar este factor, será importante que la solución propuesta sea más barata a largo plazo (considerar el "*total cost of ownership*"), y ofrezca un grado de calidad al menos equivalente a competidores propietarios. También es importante destacar que una empresa de servicios en torno al software libre debería resultar más atractiva a sus clientes por la reducción de la posibilidad de situaciones de captura (*lock-in*): estos proveedores no podrán contar con la continuidad de los ingresos en una situación con clientes cautivos, sino que deberán basarse en la **continua prestación de servicios de calidad**.

Por otro lado, que un software sea libre no significa que sea accesible a todo el mundo. El mercado para empresas de servicios no disminuirá por la disponibilidad de aplicaciones libres o eventualmente gratuitas, ya que el trabajo

de selección, instalación, formación y soporte siempre será necesario en entornos corporativos, y resultará más interesante si el presupuesto de licencias se dedica a un mejor servicio.

Como regla general, este tipo de empresas participan en varios proyectos, aunque no de manera intensiva en ninguno de ellos. Algunas contribuirán, como en el caso de los distribuidores de plataformas, en la solución de errores, especialmente en áreas de interés de sus clientes, así como en tareas de integración y de conseguir la compatibilidad entre distintas aplicaciones. Otras, como las centradas en consultoría y selección (y sin capacidad de desarrollo), no contribuirán a los proyectos sobre los que se sustentan, ya que normalmente su trabajo se mantiene privado y no será visible al público. En estos casos, sin embargo, se puede producir un retorno en forma de promoción y adopción de la solución sobre la que trabajen.

El rango de modelos posibles en esta categoría es enorme (diferenciación en cuanto a tamaño, a segmentación de solución -horizontal o vertical- a segmentación del sector, especialización en algún servicio en concreto: desarrollos a medida, selección, consultoría, integración, formación, etc.), y la mayoría de las empresas ofrecerán una combinación de los servicios posibles. Vamos a examinar primero las características especiales del software libre en las distintas fases de implementación de una solución tecnológica, y más adelante nos centraremos en tipologías concretas de modelos de negocio, que agrupan ciertos servicios de una manera particular.

### **Desarrollos a medida**

El software libre ofrecerá a las empresas un término medio ante la cuestión de "**comprar o desarrollar**". Estas empresas podrán partir de un producto estándar libre, y ya sea de forma interna, o a través de una empresa de desarrollo, construir las adaptaciones necesarias a sus necesidades. Tanto las empresas de servicios que examinaremos a continuación, como las anteriores orientadas a productos, recibirán ofertas para llevar a cabo este tipo de personalizaciones. Sin embargo, realizar estas adaptaciones de forma privada, y sin intentar que sean incorporadas al proyecto principal puede resultar problemático a la hora de mantener las adaptaciones compatibles con las siguientes versiones. En este sentido, trabajar junto con la comunidad, diseñando las nuevas funcionalidades de manera que puedan ser interesantes para más personas, e incorporarlas al código principal del proyecto ahorrará mucho trabajo y complicaciones.

### **Selección**

La presencia de infinidad de aplicaciones, al alcance (económico) de cualquier empresa, hace de la selección una tarea fundamental. No sólo será necesario encontrar productos que cumplan de la forma más ajustada las necesidades de la empresa cliente, sino también evaluar la salud de determinados proyectos, el ritmo de solución de errores y nuevos lanzamientos, y su estabilidad.

Para entornos corporativos, un proyecto con mucho movimiento, y un rápido ritmo de incorporación de mejoras puede no ser el mejor, sino que eventualmente puede resultar más adecuado un producto estable y que no vaya a cambiar sustancialmente con el tiempo.

## Instalación e integración

Aunque esta fase también genera necesidades en entornos comerciales, el software libre presenta una oportunidad de negocio especial en este campo: su falta de empaquetamiento y acabado final. En *Open Source for the Enterprise*, Woods y Guliani aluden al concepto de "productización" como una de las carencias principales del software libre de cara a una extensa adopción. Con el término se refieren al grado en el que la aplicación ha sido empaquetada y preparada para usuarios finales, con el desarrollo de instaladores automáticos, de interfaces gráficas de configuración y una documentación suficientemente extensa que, en suma, permita su instalación y manejo por parte de usuarios inexpertos.

Como regla general, el software comercial está más empaquetado y acabado que el software libre desarrollado de manera voluntaria. Los *scripts* de instalación, las interfaces administrativas y la documentación suelen ser más completos para un producto propietario comercial, que para un producto de software libre de la misma edad. Mientras que para los entusiastas tecnológicos esta falta de terminación del producto no es importante, o incluso es atractiva (permite realizar el ajuste y la administración de una manera más directa y personal), para cruzar el abismo y conseguir al cliente corporativo, el software libre debe alcanzar un mayor grado de empaquetamiento y acabado. Según Woods y Guliani:

"Una amplia simplificación sobre el software abierto frente al comercial es que el software de código abierto representa principalmente una inversión de tiempo, y el software comercial representa principalmente una inversión de dinero. Cualquier organización que pretenda usar software libre deberá reservar tiempo para dedicar a la investigación y la experimentación."

Dan Wods y Gautam Guliani. "Open source for the enterprise"

Esta inversión de tiempo en completar una aplicación o selección de aplicaciones de código abierto presenta una importante oportunidad de negocio tanto para los integradores como para los desarrolladores de plataformas. En este sentido, una buena simbiosis se podría establecer entre el sector privado y los proyectos de software libre sin ánimo de lucro, en la que la inversión se dedicaría a realizar trabajos más monótonos, dejando a la comunidad voluntaria el trabajo más creativo y de innovación, pero permitiendo la creación simultánea de productos de mayor madurez, y con más posibilidades de alcanzar un alto nivel de adopción.

### Lectura complementaria

D. Woods; G. Guliani  
(2005). *Open Source for the Enterprise: Managing Risks, Reaping Rewards*.

Por otro lado, tanto la modularidad del software libre, como la convivencia con sistemas propietarios pueden generar serios problemas de compatibilidad, que necesitarán un trabajo de integración cuidadoso. La generalización de estándares será positiva para minimizar los aspectos adversos de la combinación de distintos elementos de software.

### Certificación técnica

Las características inherentes al acabado del software libre también abren la oportunidad para la certificación por parte de integradores y consultores externos. Puede tomar dos formas, la certificación de someterse a estándares internacionales, o la certificación de la idoneidad para entornos tecnológicos concretos. El certificador proporciona la seguridad de que el paquete cumple una serie de requisitos, y es legalmente responsable de su cumplimiento.

En este sentido, el certificador proporciona un intermediario responsable a un conjunto de soluciones, factor que para muchos departamentos de nuevas tecnologías de empresas consumidoras de software es fundamental. A menudo, cuando un departamento de tecnologías de la información contrata soporte y mantenimiento, no sólo está contratando un método de resolución de incidencias, sino un responsable al que imputar los problemas y fallos que puedan surgir. La decisión de adoptar una solución concreta de software libre sin intermediarios responsables que ofrezcan garantías pone todo el peso del éxito o del fracaso sobre el propio departamento, el cual puede preferir que esta responsabilidad la asuma el intermediario.

### Formación

La formación puede suponer una fuente de ingresos fácil. Además de que el modelo de desarrollo abierto pone a disposición de cualquiera toda la información disponible de un producto, la mayoría de los proyectos de software libre carecen de programas de formación oficiales, permitiendo a cualquiera entrar en el negocio. Muchas empresas establecidas dedicadas a la formación ya han incorporado programas de software libre entre sus ofertas.

### Soporte y mantenimiento

Hemos visto ya cómo los servicios de soporte y mantenimiento suponen una fuente de ingresos importante para empresas dedicadas al desarrollo de un producto libre, pero también formarán parte de la oferta de empresas orientadas sólo a la prestación de servicios de forma horizontal, que veremos a continuación.

Como hemos dicho, el rango de empresas de servicios posible es enorme, desarrollando modelos basados en la especialización sobre ciertos servicios, sobre un tipo de aplicaciones, de manera local o a gran escala, etc. Vamos a elegir tres tipologías para estudiar en detalle. Por un lado, las empresas distribui-

#### Lectura complementaria

S. Sieber; J. Valor (2005). *Criterios de adopción de las tecnologías de información y comunicación*. IESE.  
<[www.iese.edu/en/files/6\\_15211.pdf](http://www.iese.edu/en/files/6_15211.pdf)>

doras de plataformas, por ser uno de los primeros modelos de negocio implementados con software libre, y por ser bastante representativo de un número de importantes empresas en el sector. Por otro, hemos escogido dos ejemplos que se sitúan en los extremos en cuanto a escala: las grandes integradoras, y las pequeñas microempresas de nicho. Entre ambas se situarán el resto de los modelos de negocio posibles centrados en la prestación de servicios.

### 3.2.1. Empresas distribuidoras de plataformas

La actividad de este tipo de empresas se centra en la integración y selección de componentes para generar una **solución de software integral**. La diversidad de aplicaciones y resultados que genera el modelo de desarrollo del software libre hace necesario el trabajo de equipos integradores que den cohesión y aseguren la compatibilidad entre las partes, lo que ha generado la aparición de distintas distribuciones, desarrolladas por distintos actores. Sin duda, esta actividad también representa un modelo de negocio potencial.

Las empresas distribuidoras de plataformas siguen un modelo similar al de las empresas desarrolladoras de una aplicación y prestadoras de servicios, pero el núcleo de su trabajo es la **selección e integración de una base amplia de productos**, en lugar del desarrollo.

Las empresas de este modelo se dedican a generar y distribuir conjuntos integrados de software teniendo en cuenta, principalmente, al cliente corporativo. La plataforma generada constituye el producto principal de la empresa, lo que genera un problema importante: la diferenciación del producto se hace muy difícil, ya que es accesible de forma libre.

Además de distribuir software según un modelo tradicional vendiendo CD empaquetados, estas empresas suelen completar su oferta con servicios como instalación y soporte de calidad, a menudo mediante un sistema de suscripciones. El valor añadido que ofrecen se basa en la fiabilidad y en la confianza, transmitida por la marca que les representa. Ofrecen rellenar las carencias que puede presentar un producto de software libre en un entorno corporativo, en el que se busca una solución adecuada, estable y fiable, aun a costa de prestaciones y rendimiento.

Así, sus clientes potenciales serán empresas medianas y grandes, que requieran madurez y estabilidad, existencia de un soporte profesional, y un ecosistema de soluciones viable. La inversión en aplicaciones informáticas se amortiza en cinco años, por lo que una empresa que vaya a invertir en software necesitará saber que al menos durante ese tiempo, va a poder contar con soporte sobre

esos productos. Dados los costes adicionales que se generan al cambiar una solución tecnológica por otra, la duración del soporte más allá del periodo de amortización será muy deseable.

En este sentido, la generación de confianza es fundamental en su estrategia de empresa, y debe incluir el desarrollo de una marca que transmita fiabilidad añadida a un producto de software libre. Al basar su modelo de negocio en un producto accesible a cualquiera de forma libre. Estas empresas buscan como factor diferenciador el desarrollo de una marca fuerte que les permita ganar cuotas de mercado frente a productos iguales o muy similares.

Aunque estas empresas no se suelen centrar en el desarrollo de aplicaciones específicas, sí que suelen contribuir a los proyectos sobre los que se nutren corrigiendo errores, y sólo elaboran productos nuevos cuando es necesario para ampliar el mercado de su producto.

#### Nuevos distribuidores

Recientemente, han aparecido nuevas compañías distribuidoras, ofreciendo conjuntos de software más especializado, para mercados más limitados. SourceLabs, SpikeSource, y Wild Open Source son ejemplos de este tipo de iniciativas. SourceLabs, por ejemplo, proporciona colecciones certificadas de software que se suele usar de forma conjunta, como Linux, Apache, PHP y MySQL. Por otro lado, Wild Open Source personaliza distribuciones para su uso en contextos de alto rendimiento, o de sistemas embebidos. Junto con el conjunto certificado, al igual que las compañías clásicas de suscripciones, las empresas ofrecen servicios de soporte y mantenimiento en torno a su selección.

El principal reto de este tipo de empresas será lograr definir colecciones de software suficientemente amplias como para mantener una base suficiente de clientes, pero ser capaces de ofrecer soporte sobre todos los elementos del conjunto.

#### El caso de SpikeSource

|   |                                    |
|---|------------------------------------|
| <b>Nombre de la compañía</b>                | Spikesource, Inc.                  |
| <b>Sede</b>                                 | Redwood City, CA. (Estados Unidos) |
| <b>Página web</b>                           | www.spikesource.com                |
| <b>Fecha de creación</b>                    | 2003                               |
| <b>Nº de personas empleadas en 2006</b>     | 80                                 |
| <b>Volumen de ventas en 2007 (millones)</b> |                                    |

Datos corporativos de SpikeSource, Inc. Elaborada a partir de estadísticas de Hoovers (<http://www.hoovers.com>)

**SpikeSource** supone un claro ejemplo del potencial de negocio tras la falta de acabado de los productos de software libre. Nació en el 2003, en el seno de una de las firmas de capital riesgo más importantes del boom de Internet, Kleiner Perkins Caufield & Byers, y lanzó sus primeros productos en abril del 2005. En octubre del 2006 la empresa anunció su expansión en Europa a través de una red de proveedores de soluciones locales y socios tecnológicos.

Murugan Pal, fundador de la empresa, resume la actividad de la compañía de la siguiente manera:

#### Red Hat

El ejemplo arquetípico de empresa distribuidora de plataformas es Red Hat, Inc. siendo también el modelo que sigue Novell con SuSE, Canonical con Ubuntu, y que siguió Caldera Systems con Caldera Linux.



"La meta de SpikeSource es facilitar la adopción de software de código abierto en la empresa a través de pruebas, certificación y servicios de soporte. Nosotros innovamos, automatizamos y optimizamos técnicas de testeo avanzadas como parte de nuestra actividad principal".

(Murugan Pal. "Participatory Testing: The SpikeSource Approach". <http://www.oreillynet.com/pub/a/network/2005/04/07/spikesource.html>)

Como factores diferenciadores frente a otros distribuidores de soluciones integradas clásicas, como Red Hat, la empresa destaca sus esfuerzos por promover la automatización de las pruebas, así como la combinación de aplicaciones específicas, que puedan instalarse sobre distintas plataformas y sistemas operativos. Incluye versiones para diversos sistemas operativos, tanto libres como propietarios, además de incluir software cerrado en algunos de los productos.

Además de los conjuntos que integra, como SpikeWAMP-1.4, que proporciona las últimas versiones de PHP, MySQL y Apache, para ser instalado sobre Windows, y "SuiteTwo", orientado a proporcionar una amplia gama de aplicaciones integradas para la colaboración, y funcionalidades "Web 2.0", recientemente ha lanzado una plataforma para desarrolladores, sobre la que éstos pueden probar e integrar sus aplicaciones, obteniendo así la certificación de Spikesource, y consiguiendo como resultado un mayor acabado (productización) de su software.

El trabajo de este tipo de empresas puede resultar muy favorable a la hora de dar visibilidad y fomentar la adopción de soluciones libres, y SpikeSource destaca esta labor. La empresa se esfuerza por demostrar que su trabajo beneficia a la comunidad de software libre, y no sólo la explota, incluyendo figuras reconocidas del mundo del software libre en su comisión asesora, como Brian Behlendorf y Larry Rosen para avalarlo. También ha desarrollado una web orientada a desarrolladores (<http://developer.spikesource.com>), desde la que ofrece sus servicios de pruebas automáticas de integración y compatibilidad sobre distintas plataformas.

Sin embargo, el **software de automatización** con el que trabaja la empresa combina partes que han sido liberadas, con partes que permanecen cerradas. Reservarse una porción del código supone en este caso una estrategia para proteger su diferenciación, y evitar la entrada de competencia con servicios equivalentes. Se pone de manifiesto con esta elección que más que la pérdida de ingresos por licencias (que como hemos visto a lo largo de la asignatura no supone un obstáculo real), el uso de software libre afectará a las posibilidades de diferenciación, y por lo tanto de negocio, que tenga la empresa. En el caso de SpikeSource, el trabajo invertido en sus aplicaciones de pruebas se verá recompensado no por la venta de licencias sobre ese software, sino por la protección de su factor diferenciador frente a otras empresas ofreciendo servicios similares.

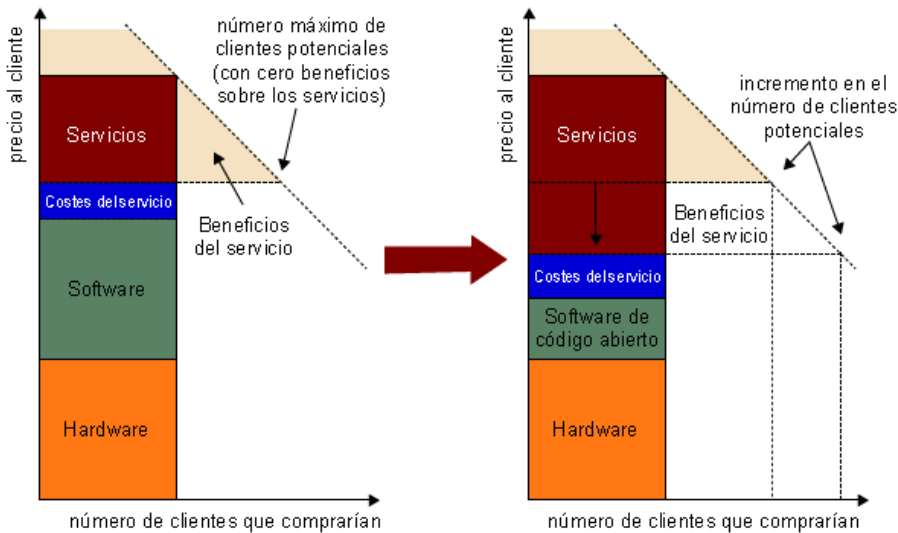
### 3.2.2. Grandes integradoras

Las grandes integradoras de sistemas, o generadores de soluciones, son unos de los tipos de empresas que más pueden ganar basando su negocio en el uso de software libre, debido al ahorro de costes directo, y la posibilidad consecuente de alcanzar más clientes.

Normalmente, un cliente estará buscando una empresa que le proporcione soluciones a un problema de tecnologías de la información y comunicación (TIC), sin importarle los detalles de implementación. Una solución completa combinará hardware, software y servicios, facilitando el proceso al cliente, que sólo necesitará contactar con una empresa para resolver sus problemas TIC, sin tener que preocuparse por la compatibilidad entre proveedores. Por lo tanto, todo lo que la empresa pueda ahorrar en costes de software empleando software libre lo podrá trasladar a los costes de servicios, mejorando así su solución. Podrá bajar el precio, aumentando así su número potencial de clientes, o

simplemente podrá mejorar su rentabilidad. Este tipo de grandes integradores que, en general, trabajan en proyectos complejos, podrán mantener los precios dadas las dificultades de entrada para otros competidores.

El siguiente gráfico ilustra esta situación, dibujando la curva de demanda de soluciones integrales, y los costes del proveedor.



Curva de la demanda de servicios informáticos integrales. Márgenes de ventas y número de clientes. Fuente: Dirk Riehle, "The Economic Motivation of Open Source Software: Stakeholder perspectives". <http://www.riehle.org/computer-science/research/2007/computer-2007-article.html>

Existen multitud de empresas de consultoría y selección como Ayamon, Enomaly, Navica, OpenLogic, Optaros, X-tend. Como grandes integradoras, podemos destacar a IBM, Sun, y HP.

### El caso de IBM

|   |  |
|---|--|
| <b>Nombre de la compañía</b>            | IBM  |
| <b>Sede</b>                             | Armonk, NY (Estados Unidos)                                |
| <b>Página web</b>                       | www.ibm.com  |
| <b>Fecha de creación</b>                | Su origen se remonta a 1896. En 1924 toma el nombre de IBM |
| <b>Nº de personas empleadas en 2007</b> | 394.540  |
| <b>Ingresos en 2007 (millones)</b>      | \$91.423   |

Datos corporativos de IBM. Elaborada a partir de estadísticas de SoftwareMagazine ([www.softwaremag.com](http://www.softwaremag.com)) y Wikipedia.

Hace veinte años, IBM era uno de los más fuertes defensores de los derechos de propiedad intelectual para el software. Defendía que sin una protección fuerte de *copyright*, no habría incentivos para que las empresas invirtieran en el desarrollo de software.

Actualmente, aunque mantiene el grueso de su software propietario, IBM ha puesto en marcha importantes campañas de apoyo al software libre, con considerables contribucio-

nes económicas al desarrollo de Linux y otras aplicaciones, y la liberación de aplicaciones como la plataforma de desarrollo Eclipse, y parte de su sistema operativo AIX.

El modelo de negocio actual de IBM se enfoca en la **venta de hardware de alta gama**, software propietario sobre Linux, y la **prestación de servicios de integración a clientes corporativos**. Aunque IBM ha sido uno de los principales fabricantes de software a nivel mundial, sus programas han sido normalmente comercializados como una solución combinada con su propio hardware. En este sentido, la empresa no tiene mucho que perder por la falta de diferenciación del software que usa: dadas las dificultades de entrada a competidores en el campo de los *mainframes*, el uso de software de bajo coste permitirá a la empresa bajar sus precios y ampliar su gama de clientes, pero no sufrirá una pérdida de diferenciación que aumente de forma significativa la competencia.

De esta manera, el uso de Linux ha permitido a IBM ofrecer un precio global más bajo por su hardware y sus servicios, pero también ha proporcionado una plataforma común sobre la que construir y vender aplicaciones y servicios especiales. En este sentido, cabe destacar también el ahorro que supone para la empresa el uso de un sistema operativo con gran adopción previa, en cuanto a marketing, difusión y ventas, además de la reducción del riesgo y la inversión en desarrollo. Los beneficios obtenidos en cuanto a imagen pública, por otro lado, resultan asimismo significativos.

Desde luego, la actividad de IBM en torno al software libre responde a una estrategia más compleja, que le permite obtener una mejor posición competitiva en diversos frentes. Desde las estrategias basadas en el uso de software libre para mejorar la comercialización de sus productos propietarios (como los "*loss leader*" y núcleo libre más accesorios propietarios), a conseguir una mejor posición respecto de otras grandes empresas proveedoras de software.

El uso de software libre ha permitido a IBM obtener más independencia respecto a otras grandes empresas como Microsoft, y una mejor posición respecto a competidores directos como Sun. Esta última ha basado su estrategia de negocio durante mucho tiempo en la venta combinada de hardware más sistemas operativos "mejores que la media", por lo que tendría más que perder frente a la reducción de costes y la presencia de software equivalente de bajo precio.

### 3.2.3. Servicios en torno al software: pequeñas y microempresas

Otro fenómeno fundamental que desencadena el software libre es el de la **transferencia tecnológica y de conocimiento**. La inversión en formación, desarrollo y tecnología, tanto a escala de grandes empresas como a escala individual, se hace disponible a través de desarrollos abiertos a cualquiera con una conexión a Internet, y con ciertos conocimientos.

Este fenómeno puede tener un gran impacto a nivel de transferencia tecnológica tanto entre países más y menos desarrollados, como de forma interna, entre grandes empresas multinacionales y microempresas locales.

La posibilidad de acceder tanto al código como a las decisiones de diseño y desarrollo libremente supone un gran potencial para pequeñas empresas tecnológicas, que pueden estar en contacto y hacer suya la tecnología más innovadora apoyada por una gran inversión financiera.

Dado su tamaño, por lo general estas empresas orientarán su actividad hacia nichos concretos, necesitando un número reducido de clientes para mantener el negocio. Las posibilidades de segmentación de mercados son múltiples,

aunque un factor común podrá ser el de proporcionar un trato más cercano y personalizado (muchos clientes preferirán ser los clientes principales de una pequeña empresa, que uno más sin importancia de una gran multinacional).

Entre este tipo de empresas, es interesante destacar aquellas que basan su diferenciación en emplear software libre no sólo por las ventajas que hemos visto hasta ahora, sino como una **declaración de intenciones**, como un elemento más dentro de una lógica empresarial que no busca la acumulación de beneficios, sino la generación de medios de vida autosustentados, a través de la prestación de servicios que contribuyan al desarrollo y bienestar de la sociedad.

El funcionamiento interno de estas empresas a menudo refleja también esta filosofía y aproximación al mundo empresarial, basándose en la **horizontalidad** y la **transparencia**. Es interesante destacar que el marco jurídico español ofrece una figura empresarial bastante ajustada a lo que hemos descrito: las cooperativas de trabajo, en las que no existen socios capitalistas, y son los propios trabajadores los que dirigen y controlan la empresa.

De nuevo, el concepto de la **ética empresarial** no es nuevo ni exclusivo del software libre, pero cobra un especial significado en este tipo de empresas. A menudo, estas pequeñas empresas se agrupan mediante distintos tipos de redes, lo que supone una estrategia fundamental para promover el apoyo y la colaboración entre ellas, siguiendo los principios éticos y políticos sobre los que se asientan.

Una parte importante de los clientes potenciales serán otras empresas con similares principios de funcionamiento, organizaciones con una motivación social o política, y administraciones públicas.

Como ejemplos de este modelo, varias empresas españolas con un tipo de funcionamiento similar se han estado uniendo en torno al grupo Ikusnet (<http://www.grupoikusnet.com/>), bajo los siguientes principios:

"Nuestra base metodológica se asienta en la cooperación y la 'horizontalidad' para la toma y aplicación de decisiones, de forma que el propio modo de cooperación se convierte en fuerza productiva que pretende desplegar sus efectos en el marco de la sociedad de la información y el conocimiento."

También podemos destacar la cooperativa madrileña **Xsto.info** (<http://xsto.info>), una microempresa de menos de diez trabajadores. Nacida en el seno de movimientos sociales, se constituyó como cooperativa de trabajo en el año 2003. Esta elección de forma jurídica supone, al igual que el uso de software libre, una declaración de intenciones en cuanto a sus principios de funcionamiento, que completan en su página web, con el siguiente lema:

"Aún estamos a tiempo de participar de esta transformación social para que se produzca de una forma participativa, abierta, libre y democrática".

Cuenta entre sus clientes con administraciones locales como el Ayuntamiento de Parla, y distintos tipos de asociaciones, como la Federación Regional de Asociaciones de Vecinos de Madrid.

Otro ejemplo muy representativo, y de especial interés por su edad, es el caso de Easter-eggs que examinamos a continuación.

### El caso de Easter-eggs

|                                  |                     |
|----------------------------------|---------------------|
| Nombre de la compañía            | Easter-eggs         |
| Sede                             | París, Francia      |
| Página web                       | www.easter-eggs.com |
| Fecha de creación                | 1997                |
| Nº de personas empleadas en 2007 | 15                  |
| Cifra de negocio en 2006         | 800.000€            |

Datos corporativos de Easter-eggs. Obtenidos de su página web (<http://www.easter-eggs.com>)

Easter-eggs es una pyme francesa dedicada a prestar servicios en torno al software libre, con una consolidada trayectoria. Creada en 1997, ofrece un amplio abanico de servicios, desde instalación, administración y seguridad de sistemas GNU/Linux; adaptación de aplicaciones y desarrollos a medida; a consultoría, auditoría y formación. Es una de las empresas de servicios en torno al software libre más antiguas, y goza de buena salud: rentable desde el momento de su formación, actualmente cuenta con quince personas asalariadas, y en el 2006 alcanzó una cifra de negocio de 800 mil euros. Entre sus clientes, podemos mencionar a la Universidad René Descartes de París (<http://www.univ-paris5.fr/>), y Europcar, para la que pusieron en marcha un programa de migración a GNU/Linux en 3.500 de sus agencias.

Para la empresa, la elección de prestar servicios en torno al software libre se basó sobre principios éticos más que financieros, y estos principios son los que la han llevado también a definir un esquema de funcionamiento empresarial muy particular. De una manera muy similar a la que rige el funcionamiento de las cooperativas de trabajo españolas, la empresa Easter-eggs está totalmente controlada por sus personas asalariadas, y sólo por ellas. No hay socios capitalistas, ni ningún tipo de inversión externa. Para conseguir este esquema organizativo, crearon una asociación, la Asociación de Asalariados de Easter-eggs (<http://www.easter-eggs.org>), que posee el 99,8% del capital de la empresa.

Sobre esta base, Easter-eggs construye su diferenciación empresarial, denominándose a sí misma como "una empresa social" con una preocupación central por crear una "empresa ciudadana", que dé respuesta a las aspiraciones crecientes de los ciudadanos que comienzan a percibir los límites del consumismo, y reclaman un comportamiento con sentido de las empresas. Entre sus principios de funcionamiento destacan la transparencia financiera (su contabilidad está disponible para la descarga desde su página web: [http://www.easter-eggs.org/rubrique\\_10\\_Comptabilite.html](http://www.easter-eggs.org/rubrique_10_Comptabilite.html)), la igualdad salarial, y mecanismos de implicación y corresponsabilidad por parte de sus trabajadores.

Como parte de la estrategia de creación de redes, y de agrupación de pequeñas empresas socialmente responsables, para poder prestar servicios a gran escala, y como medio de promoción conjunto, la asociación Easter-eggs creó, en el año 2002, la red *libre enterprise* (<http://www.libre-entreprise.org>), en la que se pueden encontrar unas dieciséis empresas francesas de servicios basados en software libre, siguiendo similares modelos de negocio.

### 3.3. Mercados auxiliares: Hardware

Uno de los primeros modelos de negocio descritos por Hecker, "Widget Frosting", sigue igual de vigente en la actualidad. Para fabricantes de hardware, el desarrollo de software supone un gasto necesario para poder vender sus pro-

ductos, por lo que cualquier estrategia que permita bajar los costes relacionados es deseable. Además, seguir un modelo de desarrollo de software libre amplía las posibilidades de portabilidad a otras plataformas, aumentando su segmento de mercado. Hemos visto anteriormente cómo las grandes proveedoras, que incluyen hardware en su oferta, están incluyendo sistemas operativos libres como una manera de reducir los costes finales del servicio, y por lo tanto aumentar su base de clientes potenciales.

En este campo, es interesante destacar el papel que está teniendo Linux en la nueva generación de **dispositivos empotrados**. Hemos vivido un retorno a la venta conjunta de hardware y software en este tipo de dispositivos, que deben venderse con su funcionalidad concreta incorporada, a menudo con sistemas operativos simples y de escasa funcionalidad. Sin embargo, la posibilidad de usar Linux empotrado ha multiplicado las oportunidades de negocio de este tipo de hardware.

El uso de software libre ofrece importantes ventajas en términos de ahorro de costes, de acortamiento de los periodos de desarrollo (fundamental en un mercado que se rige por ciclos de vida cortos), de facilidad a la hora de subcontratar desarrollos (al partir de una base existente de gran modularidad), y las posibilidades de innovación que brinda construir una comunidad en torno al producto. Por otra parte, el uso de software libre otorga a los fabricantes una importante independencia frente a las plataformas Windows Mobile y Symbian, y por lo tanto, de las agendas de Microsoft y Nokia.

Actualmente, los sistemas operativos basados en Linux son los más utilizados en sistemas empotrados, y su adopción por parte de empresas consolidadas en el sector, como Wind River, avala la continuidad de esta tendencia. En el mercado de los móviles SmartPhones, Linux pasó del 3,4% en el 2004 al 14,3% en el 2005, mientras que Windows empotrado sólo creció del 2,9% al 4,5% en el mismo periodo.

Por otro lado, la existencia de software a un precio asequible para una extensa audiencia también genera un ecosistema de necesidades a su alrededor, del que muchas veces forma parte el hardware. La plataforma de voz por IP **Asterisk**, por ejemplo, pone al alcance de muchos negocios el establecimiento de centralitas telefónicas, y una importante reducción de gastos. Sin embargo, su implementación lleva consigo la necesidad de adquirir determinados elementos de hardware, como terminales IP, tarjetas Asterisk, *routers*, sistemas de grabación, etc.

Los fabricantes de estos productos se podrán beneficiar de la difusión de software como Asterisk, por lo que tendrán mucho que ganar de la participación y contribución en su desarrollo. De manera inversa, las empresas desarrolla-

#### Web recomendada

Para más información:  
Alejandro Lucero, "Seminario UAM: Linux en Sistemas Empotrados".  
[www.os3sl.com/Documents/Seminario\\_UAM\\_I.pdf](http://www.os3sl.com/Documents/Seminario_UAM_I.pdf).

doras del software podrán obtener ingresos mediante la venta de hardware y servicios relacionados, tal y como hace Digium, responsable principal del desarrollo de Asterisk.

Pero además, surgen otros espacios y nichos susceptibles de ser explotados gracias a esta tecnología, como los que aprovecha **Avanzada7**. Esta empresa malagueña vende el hardware necesario para la implementación de Asterisk, pero reconoce que no es un fabricante, ni un gran distribuidor. Su aspecto diferenciador es la oferta de servicios de soporte gratuitos a partir de la venta de los dispositivos. Además, Avanzada7 ha establecido relaciones con Digium, la empresa responsable del desarrollo del software, generando una red de confianza que extiende hacia otras empresas que deseen implementar Asterisk en clientes finales. Pone en marcha, de esta manera, una red del tipo piramidal descrito anteriormente, basándose en necesidades generadas por un software libre, y explotándolas a través de estrategias de *coopetencia*.

### El caso de Chumby

|                                  |                                   |
|----------------------------------|-----------------------------------|
| Nombre de la compañía            | Chumby Industries, Inc.           |
| Sede                             | San Diego, CA<br>(Estados Unidos) |
| Página web                       | www.chumby.com                    |
| Fecha de creación                | 2005                              |
| Nº de personas empleadas en 2007 |                                   |
| Cifra de negocio en 2006         |                                   |

Datos corporativos de Chumby Industries, Inc.

**Chumby Industries** se creó con el objetivo de crear y comercializar "el Chumby", lanzado en agosto del 2006. Se trata de un dispositivo inalámbrico (wifi), orientado a reemplazar la radio-despertador, con capacidad de conexión a la "red Chumby", desde donde descargar distintos tipos de información. Puede reproducir *podcasts*, radio por Internet, y algunos vídeos. El dispositivo funciona con Linux, y con Flash Lite, programa de Adobe para proporcionar pequeñas aplicaciones interactivas: "*widgets*". No cuenta con un navegador, el contenido sólo se puede descargar a través de los *widgets*, diseñado cada uno para una función concreta: leer las últimas noticias de un blog, descargar las últimas fotos de una galería, etc.

Tanto el software como el hardware de Chumby son libres, se pueden descargar tanto sus esquemáticos y circuitos impresos, como su código fuente. La empresa centra su actividad de marketing en torno a su apertura: el Chumby se puede personalizar a todos los niveles, cambiándole la carcasa exterior y cosiéndole (literalmente) extensiones al gusto, crear nuevos *widgets*, o "hackear" el hardware. De esta manera, el dispositivo no sólo se vende como "amigo del usuario", sino que abre la puerta a la expansión de sus funcionalidades más allá del control y financiación de la empresa, dejando que evolucione hacia lo que cada usuario quiera que sea.

Sin embargo, el modelo de negocio de Chumby no es obtener ingresos por hardware, y el precio del dispositivo se mantiene relativamente bajo. Steve Tomlin, fundador y gerente de la compañía argumenta que varios modelos de negocio se hacían posibles con Chumby: podrían cobrar más y seguir un modelo de vendedor de hardware tradicional, con los problemas de ingresos recurrentes que se generarían, o podrían cobrar poco por el dispositivo, pero cobrar suscripciones por el contenido. Sin embargo, prefirieron una tercera vía: obtener los ingresos justos para cubrir costes con las ventas, y obtener beneficios a través de la publicidad.

Para garantizar este nuevo campo de negocio, Chumby no es 100% abierto, estableciendo limitaciones en cuanto a su uso, tanto a nivel de hardware como a nivel de la "red Chumby", para garantizar su modelo de negocio.

#### Acceso a la "red Chumby"

Tras adquirir un Chumby, el usuario debe registrarse en la web de la empresa para tener acceso a los *widjets*, aceptando unos términos de uso. Estos términos permiten que cualquiera incorpore nuevos *widjets*, con cualquier tipo de información que deseen, dando permiso para su distribución a cualquier dispositivo conectado a la red. Se establecen, sin embargo, unos límites en cuanto al contenido permitido, prohibiendo contenido inadecuado (racista, violento, sexista, *spam*...), pero también contenido comercial:

"**Prohibited Content** includes Content that: (...) except as expressly approved by Chumby, involves commercial activities and/or promotions such as, without limitation, contests, sweepstakes, barter, advertising, or pyramid schemes." (<http://www.chumby.com/pages/terms>)

Los contenidos publicitarios, por lo tanto, necesitarán pagar para obtener la autorización. Por otro lado, los términos también avisan de que al conectarse a la red Chumby, se recibirá publicidad.

Aunque la incorporación de *widjets* al margen de la red Chumby es técnicamente posible a través de dispositivos USB, la empresa confía en que la mayor parte de las aportaciones se mantendrán en el seno de su red, consiguiendo atraer suficiente contenido como para cobrar valor por el número de personas y de contribuciones en torno a ella.

#### El dispositivo

Chumby permite el acceso a los esquemáticos y PCB de su dispositivo, sin embargo, los fabricantes que quieran aprovechar sus diseños e incorporarlos a sus productos, tendrán que pagar a la empresa por licenciar su nuevo producto, y tendrán que aceptar que, además de otras redes a las que se conecte, también cuente con la red Chumby.

En resumen, Chumby reconoce que el valor de su dispositivo está en el contenido, de una manera análoga a la de O'Reilly en "Open Source Paradigm Shift", y otros. Su estrategia, además de caracterizar a su producto por su apertura, es atraer al mayor número de personas a su red, intentando convertirla en red de referencia para pequeños dispositivos móviles de este tipo. Sin embargo, en lugar de vender el contenido a través de suscripciones, ha decidido capitalizar el valor a través de la publicidad.

Para la empresa, el uso de software y hardware abierto supone una estrategia clave de cara a la difusión y adopción, no sólo de su dispositivo, sino de la red que ha creado para proporcionarle contenido. Por otra parte, su carácter abierto le proporciona una clara diferenciación y ventaja comercial frente a productos similares como el iPod Touch y el iPhone de Apple.

### 3.4. Otros mercados auxiliares

La propia generalización del software libre, tanto por su forma de desarrollo como por su uso, genera en sí misma otros mercados relacionados que han sido explotados por distintas empresas:

- **Comunidad y desarrollo:** quizás los ejemplos más obvios sean los que proporcionan servicios de alojamiento y herramientas de trabajo colaborativo para proyectos de software, como SourceForge, Collabnet, o Fresh-Meat. También podemos destacar la proliferación de buscadores de código como Google Code, Koders, Krugle o Codase.
- **Certificaciones legales:** también crecen en importancia las empresas dedicadas a este tipo de certificaciones, que aseguren que un software o una combinación concreta será posible legalmente, y conozca los problemas derivados de las licencias que puede tener. Este servicio lo están ofreciendo



empresas que hemos visto antes, como las creadoras de plataformas y "paquetes", como SpikeSource, pero también han surgido algunas dedicadas íntegramente al trabajo legal, como Black Duck, y Palamida.

- **Venta de libros:** O'Reilly con sus libros es otro de los ejemplos más mencionados de estas categorías.
- **Merchandising:** tampoco debemos olvidar la importancia del *merchandising* como forma de financiación complementaria, o incluso principal. Podemos destacar el papel que juega ThinkGeek, filial de SourceForge, en la aportación de ingresos, a través de la venta por Internet de diversos tipos de productos para *geeks*: desde camisetas y tazas de té, hasta diversos dispositivos o *gadgets*.

## Resumen

Este módulo se ha dedicado a estudiar detalladamente los diferentes modelos de negocio válidos y viables basados en el software libre. El crecimiento de empresas dedicadas íntegramente a su explotación, así como la reorientación de la estrategia de multinacionales del software, es una prueba concluyente.

En un primer momento, se han presentado diferentes clasificaciones propuestas por distintos autores a lo largo del tiempo:

- Las clasificaciones de Hecker y de Raymond, basadas en la observación de empresas que usaban el software libre como parte de sus modelos de negocio.
- La clasificación del Grupo de Trabajo Europeo de Software Libre, basada en el modelo de financiación empresarial.
- La clasificación de Daffara, basada en estudios empíricos.

Finalmente, se ha propuesto y desarrollado detalladamente una propuesta propia y particular de modelos de negocio:

- Especialistas/verticales, centradas principalmente en el producto de software libre y que eventualmente pueden adoptar modelos mixtos de doble licencia, de accesorios propietarios, de venta distribuida del producto, o bien modelos de comercialización de servicios sobre el producto, como el software como servicio.
- Servicios asociados, como los desarrollos a medida, la selección, instalación, integración, certificación técnica, formación, soporte y mantenimiento de productos; que eventualmente se pueden orientar a la distribución de plataformas, a la integración a gran escala o al servicio de pequeñas empresas y microempresas.
- Mercados auxiliares de hardware, que con el software libre complementan una orientación principal de negocio a la venta de productos físicos o bien directamente al negocio de los contenidos accesibles desde un determinado hardware.
- Otros mercados auxiliares, como las herramientas de colaboración, las certificaciones legales, la venta de libros o el *merchandising*.

## Bibliografía

**Augustin, L.** (2007). "A New Breed of P&L: the Open Source Business Financial Model". Open Source Business Conference (OSBC) *Metcalfe, Randy*. <[http://www.osbc.com/live/images/13/presentation\\_dwn/A\\_New\\_Breed\\_of\\_P\\_and\\_L.pdf](http://www.osbc.com/live/images/13/presentation_dwn/A_New_Breed_of_P_and_L.pdf)> [Consulta: febrero 2009]

**Mickos, M.** (2007). "Open Source: why freedom makes a better business model". *Open Source Business Conference (OSBC)*. <[http://www.osbc.com/live/images/13/presentation\\_dwn/Keynote-Open\\_Source\\_Why\\_Freedom.pdf](http://www.osbc.com/live/images/13/presentation_dwn/Keynote-Open_Source_Why_Freedom.pdf)> [Consulta: junio 2008]

**West, J. y Gallagher, S.** (2006). "Patterns of Open Innovation in Open Source Software". En: Henry Chesbrough; Wim Vanhaverbeke; Joel West (eds.). *Open Innovation: Researching a New Paradigm* (pág. 82-106). Oxford: Oxford University Press. <<http://www.openinnovation.net/Book/NewParadigm/Chapters/index.html>> [Consulta: junio 2008].

**Capiobanco, F.; Onetti, A.** (julio, 2005). "Open Source and Business Model Innovation. The Funambol case". En: M. Scotto; G. Succi (eds.). *Proceedings of First International Conference on Open Source (OSS2005)* (pág. 224-227). Génova. <<http://oss2005.case.unibz.it/Papers/4.pdf>> [Consulta: junio 2008].

**Riehle, D.** (2007). "The Economic Motivation of Open Source Software: Stakeholder perspectives". *IEEE Computer* (vol. 4, núm. 40, págs. 25-32). <<http://www.riehle.org/computer-science/research/2007/computer-2007-article.html>> [Consulta: febrero 2009]

**Comino, S.; Manetti, F. M.** (2007). *Dual licensing in open source markets*. Univerista Degli Studi di Trento, Departamento de economía. <[http://www-econo.economia.unitn.it/new/pubblicazioni/papers/18\\_07\\_comino.pdf](http://www-econo.economia.unitn.it/new/pubblicazioni/papers/18_07_comino.pdf)> [Consulta: junio 2008].

**Daffara, C.** (2007). *Business models in FLOSS-based companies*. Conecta Research, 2007. <<http://opensource.mit.edu/papers/OSSEMP07-daffara.pdf>> [Consulta: junio 2008]

**Pal, M.** (julio, 2005). "Participatory Testing: The SpikeSource Approach". O'Reilly Network. <<http://www.oreillynet.com/pub/a/network/2005/04/07/spikesource.html>> [Consulta: junio 2008]

**Kelsey, J.; Schneier, B.** (junio, 1999). "The Street Performer Protocol and Digital Copyrights". *First Monday* (vol. 4, núm. 6). <[http://www.firstmonday.dk/issues/issue4\\_6/kelsey/](http://www.firstmonday.dk/issues/issue4_6/kelsey/)> [Consulta: junio 2008]

**Rasch, C.** (junio, 2001). "The Wall Street Performer Protocol". *First Monday* (vol. 6, núm. 6). <[http://www.firstmonday.org/issues/issue6\\_6/rasch/index.html](http://www.firstmonday.org/issues/issue6_6/rasch/index.html)> [Consulta: junio 2008]

**Daffara, C.; Barahona, J. B. y otros** (2000). "Free Software/Open Source: Information Society Opportunities for Europe?" *Working paper*. <<http://eu.conecta.it/paper/>> [Consulta: febrero 2009]

**Lucero, Alejandro.** "Seminario UAM: Linux en Sistemas Empotrados". <[www.os3sl.com/Documents/Seminario\\_UAM\\_I.pdf](http://www.os3sl.com/Documents/Seminario_UAM_I.pdf)> [Consulta: Junio 2008]

**Raymond, E.** (1999). *The Magic Cauldron* <<http://catb.org/~esr/writings/magic-cauldron/>>

Traducción al español en: <<http://gnuwin.epfl.ch/articles/es/magiccauldron/es-magic-cauldron/es-magic-cauldron.html>> [Consulta: febrero 2009]

**Hecker, F.** (1998). *Setting Up Shop. The Business of Open Source Business* <<http://hecker.org/writings/setting-up-shop>> [Consulta: febrero 2009]

**Metcalfe, Randy** (2006). *Open Source Business: Differentiation and Success* <<http://www.oss-watch.ac.uk/resources/businessmodels.xml>> [Consulta: febrero 2009]

### Casos de estudio

50 *Open Source Success Stories in Business, Education, and Government* <<http://www.blogcrm.com/50-open-source-success-stories-in-business-education-and-government.php>>

Red Hat and J. Boss. "Is Open Source viable in Industry? The case of Red Hat and JBoss". <<http://www.whyfloss.com/es/conference/madrid08/getpdf/68>>

Avanzada7. "Business models based on Asterisk: The case of Avanzada7". <<http://www.whyfloss.com/es/conference/madrid08/getpdf/64>>

Openbravo. "Openbravo: keys to success in free software application development". <<http://www.whyfloss.com/es/conference/madrid08/getpdf/49>>

Liferay. "Liferay Enterprise Portal: The project, the product, the community and how to extend it". <<http://www.whyfloss.com/es/conference/madrid08/getpdf/66>>

Varios casos. <<http://www.opensourceacademy.gov.uk/solutions/casestudies>>