

# Software como negocio

Irene Fernández Monsalve

PID\_00145051



Universitat Oberta  
de Catalunya

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



# Índice

|  |    |
|--|----|
| <b>Introducción</b> .....  | 5  |
| <b>Objetivos</b> .....   | 6  |
| <b>1. Posibilidades de negocio en torno al software</b> .....      | 7  |
| 1.1. Empresas prestadoras de servicios .....                       | 8  |
| 1.1.1. Especialización vertical .....                              | 8  |
| 1.1.2. Especialización horizontal .....                            | 9  |
| 1.2. Empresas desarrolladoras: ¿crear productos o servicios? ..... | 10 |
| 1.2.1. Necesidad de inversión inicial .....                        | 12 |
| 1.2.2. Mantenimiento del flujo de ingresos .....                   | 13 |
| 1.3. Modelos híbridos .....  | 16 |
| 1.4. Software como servicio .....                                  | 16 |
| <b>2. Empresas dominantes en el sector</b> .....                   | 19 |
| <b>3. Marketing en la empresa: ¿A quién vender?</b> .....          | 22 |
| 3.1. Mercados de nicho y mercados de masas .....                   | 22 |
| 3.2. Patrones de adopción tecnológica y el "abismo" .....          | 24 |
| <b>4. Función del producto: ¿Qué vender?</b> .....                 | 27 |
| <b>Resumen</b> .....   | 29 |
| <b>Bibliografía</b> .....  | 31 |



## Introducción

En este módulo vamos a prestar atención a la visión más "clásica" del software como negocio. Nos centraremos en el punto de vista del software propietario, dejando para un módulo posterior el estudio de las posibilidades adicionales que presenta el software libre en este escenario. Aunque algunos de los aspectos que trataremos se vuelven irrelevantes al aplicar estrategias de software libre, otros seguirán vigentes en gran medida.

Repasaremos algunos de los factores clave a la hora de diseñar un negocio en torno al software, como **la elección de actividad principal** y el enfoque general de la empresa (vender productos o servicios), aspectos de **comercialización** y **marketing** (cómo elegir un mercado, y cómo dirigirse a él), y la **definición de sus productos o servicios** (qué tipo de productos o servicios se van a desarrollar, y cómo se posicionarán).

## Objetivos

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

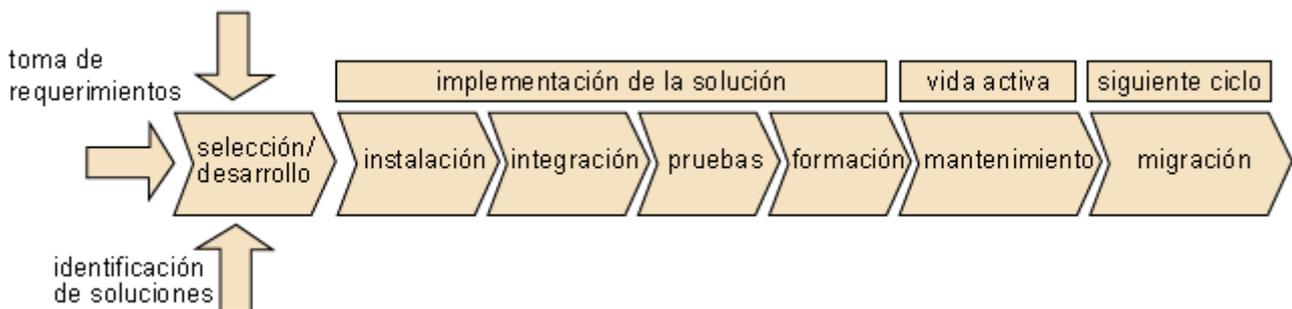
- 1.** Adquirir una visión global de las posibilidades de negocio en torno al software.
- 2.** Conocer los modelos tradicionales de empresas de software.
- 3.** Entender las características económicas y las diferencias entre las empresas de productos y las de servicios.
- 4.** Identificar los factores clave que debe tener en cuenta una empresa de software para posicionar sus productos en el mercado.

## 1. Posibilidades de negocio en torno al software

Tanto las personas individuales como los entornos corporativos cuentan con unas necesidades de software que generan múltiples oportunidades de negocio.

La tarea central para satisfacer estas necesidades será la creación de ese software, el trabajo puro de desarrollo. Sin embargo, ahí no acaban las necesidades a explotar, sino que no hacen más que empezar. Una vez se dispone de un producto, surgen una serie de necesidades relacionadas, de consultoría, instalación, configuración, mantenimiento, soporte y formación, por las que ciertos clientes (principalmente otras empresas) estarán dispuestos a pagar.

A lo largo de todo el proceso de adopción de una tecnología, desde la identificación de necesidades, la decisión de construir o comprar, hasta el fin de su vida útil, se generan múltiples necesidades a las que distintas empresas pueden dar respuesta:



**Proceso de adopción de una tecnología** (elaborado a partir de Carlo Daffara. "Sustainability of FLOSS-Based economic models". II Open Source World Conference. Málaga. Disponible en: <http://www.cospa-project.org/Assets/resources/daffara-OSWC2.pdf>)

Por otra parte, el propio proceso de creación del software se puede entender de dos maneras: como la creación de un producto, o como la prestación de un servicio. La elección entre ambas será determinante a la hora de definir el funcionamiento de la empresa, y el potencial de generación de ingresos que presente, dando lugar a modelos de negocio muy distintos.

Esta disyuntiva –desarrollar software como producto o como servicio– refleja a su vez la primera cuestión que una empresa consumidora de software tendrá que evaluar a la hora de adoptar una solución tecnológica: comprar un producto estándar, empaquetado, u obtener un desarrollo a medida.

Podemos distinguir, por lo tanto, las siguientes actividades empresariales centradas en el software:

- Desarrollo de aplicaciones
  - Como producto: soluciones estándar (*shrink-wrapped*)

- Como servicio: desarrollos a medida
- Prestación de servicios en torno a 1 o más aplicaciones
  - Asesoría
  - Selección
  - Instalación
  - Integración
  - Formación
  - Mantenimiento y soporte
  - etc.
- Software como servicio (*software as a service*)

Esta clasificación no pretende ser exhaustiva ni excluyente, es decir, muchas empresas implementarán modelos híbridos que les permitan ofrecer soluciones integrales a sus clientes.

Las características de las empresas de software, así como sus dinámicas de negocio, variarán mucho dependiendo de las actividades en las que se centren, como veremos a continuación, pero cualquiera de los modelos tienen potencial para generar tanto empresas viables como altos beneficios.

### 1.1. Empresas prestadoras de servicios

Tal como se ha comentado anteriormente, las empresas podrán implementar diferentes actividades al mismo tiempo, especializándose en unos o más aspectos de la cadena del proceso de adopción de una tecnología.

En este sentido, para las empresas que incluyen diversos servicios entre su oferta comercial podemos distinguir dos tipos de especializaciones: vertical y horizontal.

#### 1.1.1. Especialización vertical

En general, las empresas que tengan como principal actividad el desarrollo tenderán a presentar una **especialización vertical**. Si su estrategia de negocio se centra en los desarrollos a medida, de forma natural sus actividades incluirán el resto de servicios relacionados, como **instalación, integración, y formación**. Pero, como veremos más adelante, aquellas empresas que hayan escogido una estrategia de software como producto harán bien en explotar también el resto de servicios asociados, como manera de asegurarse un flujo constante de ingresos.

|                         | Paquete 1 | Paquete 2 | Paquete 3 | Etc. |
|-------------------------|-----------|-----------|-----------|------|
| Desarrollo              | X         | X         |           |      |
| Instalación             | X         | X         |           |      |
| Integración             | X         | X         |           |      |
| Certificación           | X         | X         |           |      |
| Formación               | X         | X         |           |      |
| Mantenimiento y soporte | X         | X         |           |      |
| Migración               | X         | X         |           |      |

Especialización vertical (basada en Daffara. "Sustainability of FLOSS-Based economic models". II Open Source World Conference. Málaga. Disponible en: <http://www.cospa-project.org/Assets/resources/daffara-OSWC2.pdf>)

Resulta interesante destacar que una empresa que invierte una cierta cantidad en licencias de software espera invertir otro tanto en concepto de servicios relacionados como mantenimiento y soporte, así como en actualizaciones. De esta manera, la venta de productos a un cliente empresarial abrirá las puertas a obtener contratos de servicios con ese mismo cliente, y por lo tanto, un flujo de ingresos más constante a lo largo del tiempo.

### 1.1.2. Especialización horizontal

Por otro lado, las empresas que explotan las necesidades derivadas del uso general de productos de software a menudo abarcarán servicios en torno a varios paquetes, centrándose en una o varias de las fases de la adopción de una tecnología.

|                                  | Paquete 1 | Paquete 2 | Paquete 3 | Etc. |
|----------------------------------|-----------|-----------|-----------|------|
| Selección / Desarrollos a medida |           |           |           |      |
| Instalación                      |           |           |           |      |
| Integración                      |           |           |           |      |
| <b>Certificación</b>             | X         | X         | X         | X    |
| <b>Formación</b>                 | X         | X         | X         | X    |
| Mantenimiento y soporte          |           |           |           |      |
| Migración                        |           |           |           |      |

Especialización horizontal (basada en Daffara. "Sustainability of FLOSS-Based economic models". II Open Source World Conference. Málaga. Disponible en: <http://www.cospa-project.org/Assets/resources/daffara-OSWC2.pdf>)

Aunque existen empresas especializadas en la formación o en el soporte, a menudo las empresas de servicios abarcarán varias de las fases descritas, generando tipologías como, por ejemplo, la de la consultoría (centrándose en la selec-

ción, la asesoría, y/o certificación), o los proveedores de soluciones completas, que abarcan todas las categorías, incluyendo desarrollos a medida, e incluso la provisión de hardware.

Las empresas creadoras de distribuciones GNU/Linux siguen un modelo de **prestación de servicios con especialización horizontal**.

Estas empresas orientadas a servicios a menudo evidenciarán que sus clientes prefieren recibir soluciones completas y tratar con un único proveedor de soluciones tecnológicas. Para poder ofrecer este tipo de servicios integrales, a menudo es necesario contar con una potente infraestructura y capacidad técnica, lo que limitará la entrada de empresas medianas y pequeñas, que por sí mismas no serán capaces de dar respuesta a todas las necesidades.

Una solución común es que la empresa de servicios subcontrate las partes que no puede gestionar por sí misma. Otra solución que resulta bastante interesante es el "modelo de consultoría piramidal" que propone Daffara (*sustainability of FLOSS-Based economic models*) y que explicamos a continuación.

Como norma general, se puede decir que en el campo del soporte y el mantenimiento informáticos se cumple la regla 80/20: el 80% de las consultas son fáciles, y pueden solucionarse de forma inmediata. El 20% restante, en cambio, presenta problemas importantes, y requerirá el 80% del esfuerzo. Por lo tanto, una pequeña o mediana empresa de servicios podría abarcar un alto número de clientes, haciéndose cargo del 80% de sus incidencias, y cobrando por el servicio una cantidad moderada. Para solucionar el 20% restante, requerirá los servicios técnicos de las empresas creadoras del software, a las que previsiblemente tendrá que pagar más de lo que cobra por cada uno de sus clientes, pero menos de lo que ingresa por el conjunto de ellos.

Este modelo generará una cooperación sustentable entre las empresas desarrolladoras, con especialización vertical y las empresas proveedoras de soluciones integrales. Las primeras podrán alcanzar a más usuarios gracias a las consultoras horizontales, que supondrán además una fuente importante de ingresos. Las segundas lograrán manejar una amplia base de clientes y ofrecerles soporte de alta calidad sobre una gama de productos, manteniendo un negocio rentable siempre que su base de clientes sea suficientemente alta.

## 1.2. Empresas desarrolladoras: ¿crear productos o servicios?

Como hemos dicho anteriormente, una empresa que quiera centrarse en la actividad de desarrollo tendrá dos grandes líneas entre las que elegir: podrá generar **productos estándar**, empaquetados para vender a un mercado de masas (los llamados *shrink-wrapped*), o podrá generar **desarrollos a medida**, específicos a cada cliente según sus necesidades.

### Ejemplo de especialización horizontal

Canonical, creadora de la distribución basada en **Debian Ubuntu**, realiza un trabajo de selección e integración horizontal, abarcando un sistema operativo completo junto con diversas aplicaciones, con el objetivo principal de proporcionar una distribución fácil de usar, instalar y configurar bajo el lema, "linux para seres humanos". Sin embargo, y dado el carácter libre de Ubuntu, los ingresos de Canonical provienen de servicios relacionados, como soporte, formación y certificación.

### Web recomendada

Para más información acerca del "modelo de consultoría piramidal", podéis consultar: <http://www.cospa-project.org/Assets/resources/daffara-OSWC2.pdf>

La primera opción tiene el potencial de generar grandes márgenes de beneficios, aunque estos serán difíciles de mantener en el tiempo, y presenta unas barreras de entrada que pueden resultar infranqueables. La segunda supone una opción mucho más intensiva en cuanto a mano de obra, y con márgenes de beneficios mucho menores, aunque presenta más posibilidades de generar fuentes de ingresos constantes en el tiempo, y ser menos sensible a cambios en el entorno macroeconómico.

Vamos a analizar en detalle los aspectos diferenciadores de estas dos opciones.

### Economías de escala y posibilidad de grandes márgenes de beneficios

El proceso económico de creación de software tiene unas particularidades que no vemos en otros sectores, que le llevan a tener unos **retornos positivos de escala** enormes.

Por un lado, las compañías comerciales deben invertir grandes cantidades de dinero en desarrollo antes de tener una versión comercial de un producto para lanzar, y a menudo deben reinvertir otro tanto cada dos o tres años para mantener su flujo de ingresos constante. Al intentar generar un producto estándar, este desarrollo implica un alto riesgo, ya que no hay seguridad alguna de poder recuperar la inversión a través de ventas posteriores. Sin embargo, cuando se dispone de un producto finalizado, el coste marginal de cada copia adicional vendida es cercana a cero. La primera copia del software creada es muy cara, pero el resto no cuesta prácticamente nada.

Esto proporciona unas economías de escala, en el lado de la oferta, enormes, que se unen a unas economías de escala también importantes en el lado de la demanda: tanto por el tiempo invertido en adquirir destrezas en el uso de una aplicación, como por la posible incompatibilidad de formatos, cambiar de un producto a otro supone una tarea difícil y costosa. Como consecuencia, cuanto más grande sea la base de usuarios de un producto, más fácil es que esta base crezca y perdure en el tiempo. De esta manera, en el mercado del software se llega a situaciones de "el ganador se lleva todo" (*the winner takes it all*), pudiendo generar enormes beneficios, e impidiendo al mismo tiempo la entrada de nuevas empresas a estos mercados.

### Ejemplos de empresas generadoras de productos estándar

Entre las empresas que han aprovechado estas fuertes economías de escala, encontramos algunas de las principales empresas del sector de software, como **Microsoft** copando los mercados de sistemas operativos de escritorio, y **Oracle** con su adquisición de **PeopleSoft** en el 2005. Sin embargo, también existen pequeñas empresas, Independent Software Vendors (ISV), que explotando nichos específicos han conseguido establecer negocios viables. Como ejemplos podemos citar el "Pretty Good Solitaire" desarrollado por la microempresa de dos trabajadores **Goodsol Development Inc.** (uno de los juegos de solitario con más éxito), o "HomeSite" un editor de HTML desarrollado por la microempresa Bradbury Software en 1995, que fue comprado por Allaire Corp. (Más tarde Allaire fue adquirida por Macromedia, que en el 2005 fue a su vez absorbida por Adobe.)

#### Lectura obligatoria

M. Cusumano (2004). *The Business of Software* (cap. 1, "The Business of Software, a Personal View").

En contraste, una empresa centrada en el desarrollo a medida no disfrutará de las economías de escala del software estándar. Cada nuevo cliente requerirá un desarrollo específico, y por lo tanto una inversión en tiempo y esfuerzo costosa, a pesar de que este tipo de empresas tenderán a reutilizar sus desarrollos siempre que les sea posible.

### 1.2.1. Necesidad de inversión inicial

A la hora de crear una empresa centrada en la idea tradicional de producto surge un problema importante: la **necesidad de una inversión inicial**. Durante las primeras fases de la empresa, dedicadas al desarrollo, no habrá un flujo de ingresos, pero sí de gastos, hasta que las primeras versiones del software estén listas para ser comercializadas. Además de los gastos directamente derivados del desarrollo, habrá que tener en cuenta los gastos necesarios ligados al marketing y a la comercialización. Ante este problema, hay dos tipos de soluciones: **atraer inversión externa, o empezar con otro tipo de actividad empresarial** que genere ingresos suficientes para permitir el desarrollo simultáneo del producto.

Las **empresas de desarrollos a medida** implicarán **mucho menos riesgo** y podrán comenzar su actividad con una inversión mucho más pequeña (el desarrollo sólo se inicia después de tener un contrato firmado), pudiendo evitar la búsqueda de inversores externos.

En la literatura financiera se suele discutir más sobre aquellas empresas que financian sus desarrollos a partir de inversiones de capital riesgo, ya que resultan más atrayentes. Este tipo de financiación permitirá un crecimiento más rápido, siendo éste uno de los factores importantes para el éxito según Cusumano. (Michael Cusumano, *The Business of Software*)

#### Reflexión

En este punto, resulta interesante hacer la siguiente reflexión: ¿Con qué parámetros contamos a la hora de juzgar el éxito de una iniciativa empresarial? Los inversores, así como las publicaciones financieras, considerarán exitosas aquellas empresas que consigan presentar beneficios año tras año, y probablemente, aquellas que demuestren crecimiento. Una compañía cuyo tamaño se mantenga en el tiempo, y su cuenta de resultados no muestre beneficios no atraerá la atención de la literatura financiera ni de inversores. Sin embargo, una empresa de este tipo puede haber sido muy exitosa a la hora de crear puestos de trabajo de calidad, y a la hora de mantenerlos en el tiempo. Para muchas personas emprendedoras, éste puede suponer su objetivo principal.

Conseguir suficiente inversión externa puede suponer un escollo infranqueable, pero aun cuando se hace posible, presentará ciertas desventajas que será necesario valorar. La presencia de inversores supondrá una presión sobre las decisiones de gestión de la empresa, y requerirá que la empresa genere beneficios suficientes para poder devolver la inversión y proporcionar ganancias. Esta situación limitará la autonomía y capacidad de decisión de las personas fundadoras.

La otra opción tampoco resultaría fácil. La empresa tendría que orientar su actividad a los servicios, intentando que éstos generasen suficientes ingresos como para permitir el desarrollo simultáneo del producto. Como veremos más adelante, conseguir esto con la prestación de servicios será difícil, ya que el margen de beneficios será menor, y tanto la ausencia de economías de escala como la presencia de competencia limitan la capacidad de mantener precios suficientemente altos.

En este sentido, el software libre irrumpe con nuevas características que modifican este escenario. La posibilidad de **recortar los costes** gracias a la colaboración de personas voluntarias, así como los nuevos esquemas de difusión y comercialización que brinda esta colaboración suponen una importante disrupción de estos escenarios, teniendo el potencial de disminuir de forma considerable la inversión inicial necesaria.

Veremos más sobre estos aspectos en los módulos siguientes.

### **1.2.2. Mantenimiento del flujo de ingresos**

Sin duda, una pregunta fundamental para cualquier empresa será no sólo cómo conseguir ingresos de forma puntual, sino como mantenerlos en el tiempo. Mientras que para las empresas centradas en la prestación servicios, la continuidad será la norma (en general, si los clientes están satisfechos, continuarán necesitando los servicios de forma recurrente), en aquellas empresas centradas en la producción de soluciones estándar, el mantenimiento de un flujo constante de ingresos encontrará diferentes problemas.

#### **1) Ciclos del software**

Cusumano, en *The Business of Software*, compara el proceso de escribir un producto de software exitoso con el de escribir un *best seller*. Conseguirlo genera enormes beneficios, pero además de resultar muy difícil, sólo los genera de forma puntual. El ciclo de vida natural de un producto de software comercial lo llevará finalmente a perder su capacidad de generar ingresos.

En un principio, las primeras versiones tendrán varios fallos, y su funcionalidad no se ajustará finamente a las necesidades de sus usuarios. Esto permitirá a la compañía creadora mantener sus ingresos en el tiempo mediante el lanzamiento de nuevas versiones que incorporen progresivamente mejoras en el producto, tanto por la solución de errores como por disponer de mucha más información de requerimientos obtenida como *feed-back* de sus usuarios y clientes.

Previsiblemente, si las nuevas versiones del producto presentan suficientes mejoras y resultan más atractivas que las anteriores, continuarán generando ingresos. Sin embargo, una vez los usuarios perciben que tienen una aplicación

suficientemente buena, desaparece la motivación por pagar una nueva versión. De forma análoga, intentar mantener los ingresos de un *best seller* mediante secuelas sólo tiene una efectividad limitada.

Existen estrategias para combatir estas tendencias y mantener un flujo de ingresos a través de licencias de versiones sucesivas, las más comunes a costa del consumidor. La incompatibilidad total o parcial entre versiones sucesivas del propio producto, unidas a intensas campañas de difusión de éste, llevarán a una nueva situación de economías de escala en el lado de la demanda a favor de la versión más reciente, que obligará a muchos usuarios a cambiar aunque el producto anterior satisfaga sus necesidades.

Sin embargo, por la naturaleza de algunos productos de software, la actualización constante se hace necesaria, por las necesidades cambiantes de los propios usuarios.

**Un ejemplo ilustrativo: las aplicaciones de contabilidad, laborales y de manejo de impuestos.**

Dado que la legislación sobre impuestos y materia laboral cambia a menudo, los usuarios necesitarán una actualización de su aplicación cada vez que esto ocurra, por lo que los ingresos se pueden mantener constantes en el tiempo gracias a los ajustes coyunturales en el sistema financiero y al marco legislativo.

Por otra parte, una vez la idea inicial ha sido explotada y analizada, se abrirá el camino para que otras empresas empiecen a producir un software análogo, sin necesidad de dedicar tiempo a I+D ni a la toma de requerimientos. Si consiguen hacer el producto más rápidamente, quizás simplificándolo y manteniendo sólo las funcionalidades justas, serán capaces de competir por el mismo mercado a un mejor precio. Una vez entren suficientes empresas en ese mercado, generando productos intercambiables entre sí (*commoditization*), se llega a una situación particular: en ausencia de otros factores de diferenciación, los consumidores comprarán el producto más barato, generando una situación de alta competencia.

Este fenómeno es común a cualquier tipo de producto, y en el software debería ser posible también. Sin embargo, ciertos factores protegen las empresas dominantes de este proceso, que en una situación ideal contribuiría a una mayor difusión tecnológica, y reportaría beneficios a los usuarios (aunque haga más difícil para las empresas el logro de grandes márgenes de beneficios). Como hemos apuntado antes, existen unas fuertes economías de escala en el lado de la demanda, por lo que no será tan fácil que los usuarios perciban los productos competidores como realmente sustituibles. Por otro lado, el uso de formatos propietarios genera una importante situación de cautividad de la que es complicado salir.

En este sentido, el software libre aparece como una fuerza impulsora de una situación de **bienes libres perfectamente intercambiables**: la aparición de un producto similar que se distribuye de forma libre, o eventualmente gratuita

hará más difícil mantener altos los ingresos derivados de licencias, y puede ser una de las pocas formas de lograr romper las inercias de cautividad que genera el software propietario.

### Software libre como tecnología disruptiva

El término *tecnología disruptiva*, acuñado en 1999 por Clayton M. Christensen, se refiere a innovaciones que por su bajo precio y prestaciones, o por enfocarse a un nuevo tipo de clientes, consigue desplazar soluciones anteriores del mercado. El software libre podría suponer, de esta manera, una tecnología disruptiva, dada la eventual posibilidad de obtenerlo de forma gratuita, y dada su capacidad de contribuir a la generalización del uso del software a través de las brechas tecnológicas actuales.

La transformación del sector del software en un escenario de bienes intercambiables (*commoditization*), aunque limitaría en gran medida las posibilidades de mantener altos beneficios mediante licencias, podría abrir nuevos mercados, generando un ecosistema de necesidades en torno al nuevo producto intercambiable y de amplia adopción.

## 2) Dependencia de ciclos económicos

Las compañías tradicionales de software, centradas en productos, pueden generar enormes beneficios, pero también pueden sufrir enormes caídas y pérdidas en ciclos económicos desfavorables. A pesar de contar con negocios y productos asentados, del 2000 al 2002 muchas compañías de software perdieron del 80 al 90% de su valor, incluso Microsoft sufrió una pérdida de dos tercios de su valor (Michael Cusumano, *The Business of Software*).

Durante periodos económicos desfavorables, el consumo caerá, y los productos de software serán de los primeros en notar estos efectos. Los usuarios simplemente dejarán de comprar software, pudiendo afectar en gran medida a las empresas de productos que dependen sólo de esta fuente de ingresos. Como consecuencia, es difícil encontrar una empresa de productos pura de este tipo, ya que la seguridad de sus ingresos será demasiado discrecional y arriesgada y pasará de forma inevitable por periodos bajos.

Aunque cualquier actividad económica se verá afectada en este tipo de escenarios, las empresas centradas en servicios serán más capaces de mantener sus ingresos, dados los contratos a largo plazo con los que puedan contar, y dado que sus clientes serán principalmente otras empresas, que, aunque sea en menor medida, seguirán necesitando el mantenimiento de sus infraestructuras informáticas. En muchas ocasiones, estas infraestructuras permiten a la

empresa cliente funcionar con mayor eficiencia, aumentando así sus posibilidades de supervivencia en periodos bajos, por lo que mantendrán los gastos destinados a los servicios en torno a las nuevas tecnologías.

### 1.3. Modelos híbridos

En realidad, existen multitud de modelos híbridos, que combinan en distinto grado la venta de productos estándar, y la prestación de servicios, intentando compatibilizar ambas tendencias. Se puede considerar que el grado en el que una empresa se incline hacia los productos o servicios es indicativo de su propio ciclo de vida, y existe una tendencia generalizada de transición hacia los servicios.

#### Ejemplo de empresa de modelo híbrido

Pensemos en una empresa que empieza con un modelo puro de producto, logrando unas elevadas ventas, y unos grandes beneficios, pero que comprueba que será difícil mantener ese nivel de ingresos. Para asegurarse la continuidad, o como respuesta a periodos económicos bajos, podrá empezar a establecer contratos de servicios con algunos de sus clientes, viendo una reducción considerable en el ritmo de crecimiento de la empresa, pero obteniendo mayor estabilidad a largo plazo. Finalmente, puede que la empresa acabe por poner todo su peso sobre los servicios, habiendo saturado ya el mercado de su producto original.

Por supuesto, éste no es más que un ejemplo teórico, y muchas empresas ni completarán este ciclo, ni lo empezarán en el mismo sitio.

Por otro lado, la transición hacia los servicios no es fácil, y puede acarrear consecuencias negativas si no se hace con cuidado. Adoptar un modelo híbrido como respuesta a un momento de crisis, sin considerar detenidamente su estrategia de negocio, puede traer a una empresa de productos muchos problemas.

En momentos de falta de ingresos, la empresa puede ceder a presiones de distintos clientes para desarrollar adaptaciones del producto muy concretas, y de difícil integración en el producto estándar principal. Si esta práctica se generaliza, y la empresa pretende mantener sus ingresos mediante la venta del producto estándar, puede encontrar serias dificultades para mantener la compatibilidad entre las nuevas versiones lanzadas y las adaptaciones particulares de distintos clientes. El trabajo de depuración y desarrollo se multiplica, y en ocasiones puede llevar a la empresa a generar más gastos que ingresos.

### 1.4. Software como servicio

El concepto de "*software as a service*" (SaaS), o "**software como servicio**", se originó en 1999, como una nueva manera de implementación del software enfocada en la funcionalidad.

La visión básica de esta idea se centra en que, para los usuarios, el software se hace importante en la medida en que les permite solucionar un problema, es decir, en la medida en que les presta un servicio.

La necesidad de adquirir un producto de software, de contar con una infraestructura de hardware y software relacionado, y el trabajo necesario de instalación y soporte que conlleva, no serían, bajo este paradigma, más que un estorbo para el usuario final, que debe sobrellevarlas para poder obtener la funcionalidad deseada.

Bajo un modelo de software como servicio, todos estos inconvenientes desaparecen, y el software pasa de ser un producto que puede ser adquirido, a ser un servicio que puede ser prestado. En este sentido, es importante distinguir entre las empresas de servicios que hemos comentado anteriormente, que se dedican a **prestar servicios sobre un software** (instalación, mantenimiento, etc.), a este nuevo modelo que se enfoca sobre la **prestación del software como un servicio** (prestación de la funcionalidad concreta de ese software).

Para llevar a cabo esta idea, la empresa proveedora se haría cargo de toda la infraestructura necesaria, alojando el software requerido, y ofreciendo el servicio vía web a través de un navegador. La presencia de una infraestructura de comunicaciones suficientemente potente es necesaria, pero el resto de requerimientos tecnológicos en el lado del receptor del servicio disminuyen, permitiéndole centrar su atención sólo en la funcionalidad ofrecida.

El modelo de *software como servicio* supone una opción de bajo coste para proporcionar software a empresas, frente a la venta de productos tradicional. Por un lado, los clientes se ahorrarán una cantidad importante en concepto de mantenimiento de sus infraestructuras tecnológicas. Por otra parte, las empresas proveedoras podrán ofrecer precios más económicos, al combinar los ingresos recurrentes derivados de la prestación de un servicio y aprovechar al mismo tiempo una única instancia de su aplicación para dar servicio a un gran número de clientes.

La presencia tanto de software libre como de ofertas SaaS está amenazando a los vendedores de software tradicionales, que están sintiendo una fuerte presión con la entrada de estos nuevos competidores, y tendrán dificultades a la hora de mantener los precios de sus productos.

Los proveedores de software como servicio, además, tienen mucho que ganar del empleo de software libre. Por un lado, al usarlo en sus infraestructuras de software conseguirán un ahorro considerable de costes en licencias o en desarrollo. Por otro lado, algunas están aprovechando aplicaciones libres, con licencia GPL, como base para desarrollar sus aplicaciones críticas de negocio, manteniendo sus modificaciones cerradas, como forma de proteger la diferen-

#### Prestación de software como un servicio

Cada vez más empresas están usando este modelo para proporcionar software corporativo, como 37signals con Basecamp (herramienta de gestión de proyectos), y el popular Salesforce.com (CRM, *customer relationship management*) que permite la personalización del software en función de las necesidades de los clientes.

#### Software vía web

También encontramos numerosos ejemplos de software vía web orientado a consumidores individuales, aunque esta tendencia se ha denominado "Web 2.0". Muchas han tenido un gran éxito, como las numerosas aplicaciones de Google, o e-bay.

ciación de su negocio. Aprovechan así un vacío que contiene la GPL en este sentido: las modificaciones del código sólo deben ser redistribuidas si se redistribuye el programa. En el caso del software como servicio, el código no se llega a redistribuir, sólo lo hace la funcionalidad, por lo que la empresa no tiene la obligación de compartir sus mejoras.

## 2. Empresas dominantes en el sector

Como hemos visto, orientar una empresa hacia productos o servicios generará dinámicas empresariales muy distintas, pero ambas aproximaciones pueden dar lugar a modelos de negocio lucrativos. Sin embargo, mantener activas empresas de productos puras será muy difícil, y las barreras de entrada, enormes.

En la encuesta "Software 500" del "Software Magazine" ([www.softwaremag.com](http://www.softwaremag.com)), que cada año elabora un ranking de las primeras 500 empresas de software comercial según ingresos, podemos ver que entre las compañías más lucrativas se encuentra una representación de los dos tipos de empresas comentadas.

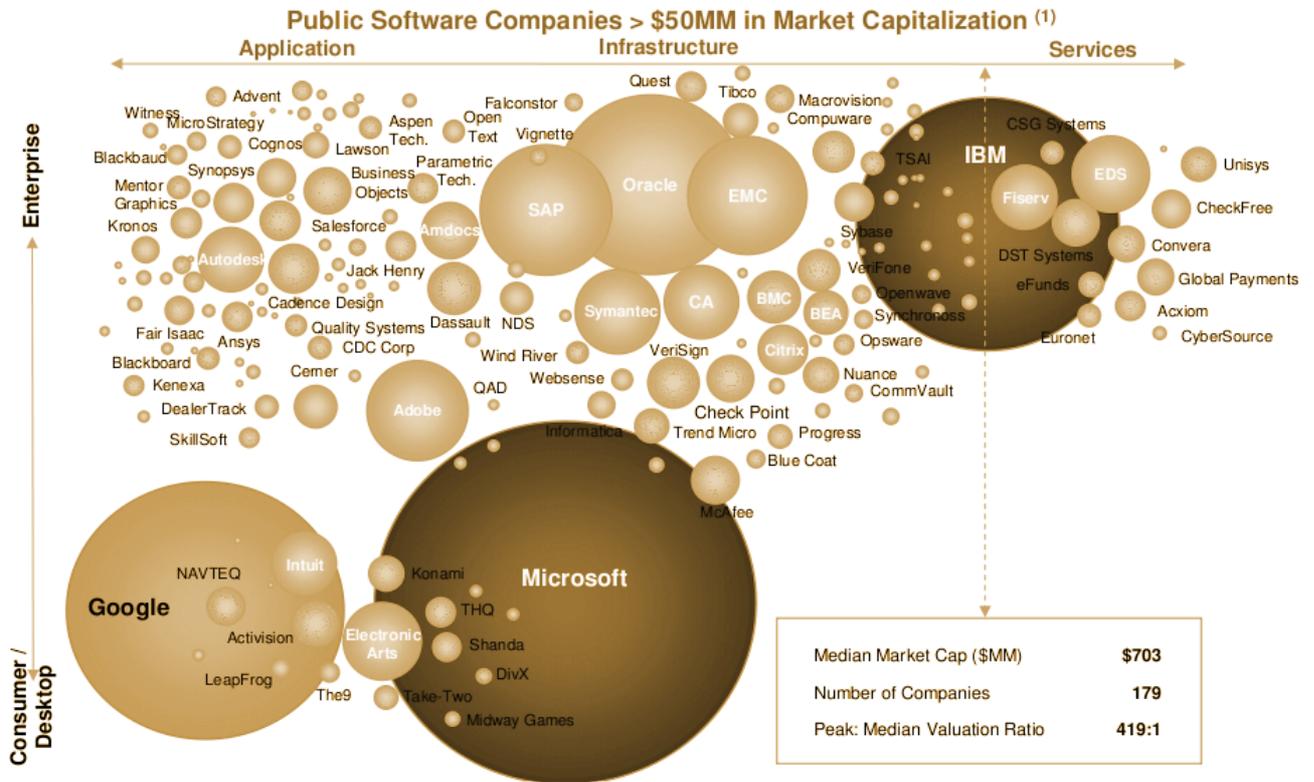
Sin embargo, de las veinte primeras, sólo cuatro de ellas mantienen un enfoque de producto marcado, con menos del 30% de servicios como porcentaje total de su actividad: Microsoft Corporation, Oracle, SAP y Symantec, representando productos líderes en sus sectores, orientados a clientes corporativos y a mercados de masas (sistemas operativos de escritorios, bases de datos, ERP y seguridad, respectivamente).

Dos empresas presentan un balance al 50% entre productos y servicios, Lockheed Martin Corporation y EMC Corporation. De las restantes, diez declaran como su sector de negocio principal la integración, la consultoría, y los servicios de subcontratación, y el resto, a pesar de dedicarse al desarrollo de productos concretos, obtienen sus ingresos principalmente de la prestación de servicios relacionados.

|    | Empresa   | Sitio web              | Ingresos derivados de software/ servicios (millones \$) | Crecimiento de ingresos (%) | Servicios como % | Nº de personas empleada | Sector de software                                  |
|----|---|------------------------|---|-----------------------------|------------------|-------------------------|---|
| 1  | IBM   | www.ibm.com            | \$66,451.00   | 3.0%                        | 72.6%            | 394,540                 | Middleware/Servidor de aplicaciones/Servidor web    |
| 2  | Microsoft Corporation                                 | www.microsoft.com      | \$39,317.00   | 9.0%                        | NA               | 71,000                  | Sistemas operativos                                 |
| 3  | EDS   | www.eds.com            | \$21,268.00   | 8.0%                        | 100%             | 118,500                 | Servicios de outsourcing                            |
| 4  | Hewlett-Packard Company                               | www.hp.com             | \$16,918.00   | 2.0%                        | 92.3%            | 156,000                 | Servicios de integración de sistemas/Consultoría IT |
| 5  | Accenture   | www.accenture.com      | \$16,646.40   | 7.0%                        | 100.0%           | 140,000                 | Servicios de integración de sistemas/Consultoría IT |
| 6  | Computer Sciences Corporation                         | www.csc.com            | \$14,615.60   | 4.0%                        | NA               | 79,000                  | Servicios de integración de sistemas/Consultoría IT |
| 7  | Oracle Corporation                                    | www.oracle.com         | \$14,380.00   | 22.0%                       | 19.7%            | 56,133                  | Bases de datos                                      |
| 8  | SAP   | www.sap.com            | \$12,309.70   | 23.0%                       | 29.2%            | 39,355                  | ERP ( <i>enterprise resource planning</i> )         |
| 9  | Cap Gemini  | www.capgemini.com      | \$10,158.60   | 23.0%                       | NA               | 67,889                  | Servicios de integración de sistemas/Consultoría IT |
| 10 | Hitachi   | www.hitachi.com        | \$9,019.20  | 5.0%                        | 85.4%            | 356,000                 | <i>Storage management</i>                           |
| 11 | Lockheed Martin Corporation                           | www.lockheedmartin.com | \$8,992.00  | 10.0%                       | 51.2%            | 140,000                 | Aplicaciones verticales industriales                |
| 12 | Science Applications International Corporation (SAIC) | www.saic.com           | \$7,775.00  | 8.0%                        | NA               | 43,600                  | Servicios de integración de sistemas/Consultoría IT |
| 13 | NTT Data Corporation                                  | www.nttdata.co.jp      | \$6,685.80  | 4.0%                        | 7.9%             | 21,308                  | Servicios de integración de sistemas/Consultoría IT |
| 14 | EMC Corporation                                       | www.emc.com            | \$6,014.50  | 16.0%                       | 51.2%            | 31,100                  | Gestión de la información                           |
| 15 | Affiliated Computer Services, Inc.                    | www.acs-inc.com        | \$5,353.70  | 23.0%                       | NA               | 58,000                  | Servicios de outsourcing                            |
| 16 | LogicaCMG plc   | www.logicacmg.com      | \$5,221.40  | 65.0%                       | NA               | 40,483                  | Servicios de integración de sistemas/Consultoría IT |
| 17 | Unisys Corporation                                    | www.unisys.com         | \$4,917.20  | 3.0%                        | NA               | 31,500                  | Servicios de integración de sistemas/Consultoría IT |
| 18 | Sun Microsystems, Inc.                                | www.sun.com            | \$4,697.00  | 19.0%                       | 100.0%           | 34,400                  | Middleware/Application Server/Web Server            |
| 19 | SunGard Data Systems, Inc. Pvt                        | www.sungard.com        | \$4,212.00  | 8.0%                        | 91.9%            | 16,600                  | Aplicaciones financieras                            |
| 20 | Symantec Corporation                                  | www.symantec.com       | \$4,143.40  | 60.0%                       | 3.3%             | 17,396                  | Herramientas de seguridad/Sistemas                  |

Primeras 20 empresas en el sector del software y su actividad principal (elaborada a partir del estudio "Software 500" de 2007. <http://www.softwaremag.com/SW500/>)

En la siguiente figura podemos ver el posicionamiento actual de éstas y otras empresas de software, en cuanto a su enfoque (aplicación, infraestructura, servicios), y el tipo de clientes a los que se dirigen (empresas o consumidores individuales).



Posicionamiento de las principales empresas de software (con capitalización de mercado superior a \$50 millones, y cotización en bolsa). John Prendergast (2008). "Can Xensource, MySQL or Jboss tell you anything about your company's prospects?". *Open Source Business Conference*. Disponible en: <http://akamai.infoworld.com/event/osbc/08/docs/CEO-CMO-Prendergast.pdf>

### 3. Marketing en la empresa: ¿A quién vender?

Hasta ahora hemos examinado varios aspectos sobre la naturaleza principal de una empresa de software, y la definición de sus actividades principales. Sin embargo, otro aspecto fundamental que debe plantearse en cualquier empresa es qué vender, y a quién dirigirse.

#### 3.1. Mercados de nicho y mercados de masas

Para cualquier empresa que cuente con unas fuertes economías de escala, como es el caso de las empresas de productos de software, cuanto mayor sea su base de usuarios, mayor será su margen de beneficios. Por lo tanto, la situación aparentemente más lucrativa sería la de orientar sus productos hacia los mercados de masas.

Sin embargo, una estrategia como ésta puede presentar muchas dificultades: el mercado de masas estará más estudiado, controlado y saturado por grandes empresas. Para una compañía que esté abriéndose camino, será extremadamente difícil llegar a competir con las ya establecidas y dominantes del sector, que además cuentan con una gran capacidad de marketing y difusión.

Resultará más fácil cubrir las necesidades detectadas en **mercados de nicho**, que por su tamaño no son atractivos para grandes empresas. Para las grandes compañías, los retornos potenciales de estos mercados son demasiado bajos, dado el reducido número de clientes, pero para una pequeña empresa serán más que suficientes. El número de nichos posibles es enorme, contando con numerosos factores sobre los que segmentar y concretar un mercado. La pregunta clave en este sentido será cuántos consumidores potenciales proporcionará este nicho, lo que permitirá calcular el volumen de negocio y por lo tanto, qué volumen de gastos podrá permitirse la empresa.

El mundo del software presenta posibilidades más interesantes que otros productos tangibles en los mercados de nicho, dada la ausencia de barreras geográficas que proporciona Internet. Un nicho detectado en un área geográfica dada podrá extrapolarse de manera relativamente fácil a otras zonas con necesidades similares, o incluso se ampliará solo, sin necesidad de un esfuerzo concreto por parte de la empresa comercializadora.

A la hora de crear productos para mercados de nicho, es fundamental conocer en profundidad ese entorno particular. Además de competencias técnicas, es necesario contar con un amplio conocimiento sobre las actividades, prioridades y forma de funcionamiento de ese nicho en particular. Siguiendo la regla de Eric Raymond, "todo buen trabajo de software comienza a partir de las necesidades personales del programador" ("*Every good work of software starts by scratching a developer's personal itch*") resulta útil partir de un nicho del que se forma parte, para entender mejor qué necesidades y problemas existen en él.

Otro factor importante a tener en cuenta es si se va a vender el producto a entornos corporativos, a pequeñas empresas o a personas individuales.

Las empresas de servicios deberán orientarse hacia entornos corporativos, administraciones públicas u otras organizaciones, ya que las personas consumidoras raramente pagarán por servicios orientados al software. Las empresas de productos, sin embargo, podrán elegir, de acuerdo a las características de sus productos y a su estrategia empresarial, los clientes potenciales del mercado objetivo. Los clientes corporativos pueden resultar más atractivos, ya que estarán más dispuestos a pagar por un producto de software, y además contribuirán a generar ingresos a través de servicios.

En un entorno corporativo, se pagará por un producto de software, pero también por el soporte sobre ese producto, la formación, la instalación y la integración en el resto de sus sistemas. Las empresas que compran software, en general pagarán del 15% al 25% del precio de la licencia en concepto de mantenimiento anual (Dan Woods, Gautam Guliani, "Open source for the enterprise"). A menudo, también solicitarán desarrollos a medida para adaptar el producto a sus propias necesidades. De esta manera, el cliente corporativo ayudará a una empresa de software a desarrollar ingresos a partir de servicios, dándole más garantías de continuidad. Sin embargo, estos nuevos ingresos serán más intensivos en cuanto a mano de obra, y será necesario llevar una gestión de la empresa cuidadosa para asegurar que los costes de la prestación del servicio no superan los ingresos derivados de ellos.

Por otro lado, los servicios de soporte a menudo se ofrecen sobre versiones concretas del producto, por lo que mantener una relación de servicios también puede ayudar a generar ingresos en forma de licencias de sucesivas versiones: aunque el cliente no tuviera interés por adquirir la nueva versión, se verá obligado dado que el soporte sobre la versión antigua ya no se realiza.

Como desventaja, un cliente corporativo de importancia será reacio a contratar servicios de una pequeña y nueva empresa. Uno de los factores esenciales a la hora de la contratación será la reputación y la confianza que le genere la compañía proveedora de servicios, por lo que las empresas menores o de reciente creación encontrarán clientes más fácilmente en su mismo entorno, es decir, entre pequeñas y medianas empresas.

#### Conocimiento del entorno

Es el caso del nicho de los desarrolladores de software: está bastante explorado y explotado, ya que todas las personas programadoras son a la vez creadoras y usuarias, contando con un conocimiento íntimo de las necesidades y problemas del sector.

#### Lectura recomendada

D. Woods; G. Guliani (2005). *Open source for the enterprise: managing risks, reaping rewards*. O'Reilly Media, Inc.

### 3.2. Patrones de adopción tecnológica y el "abismo"

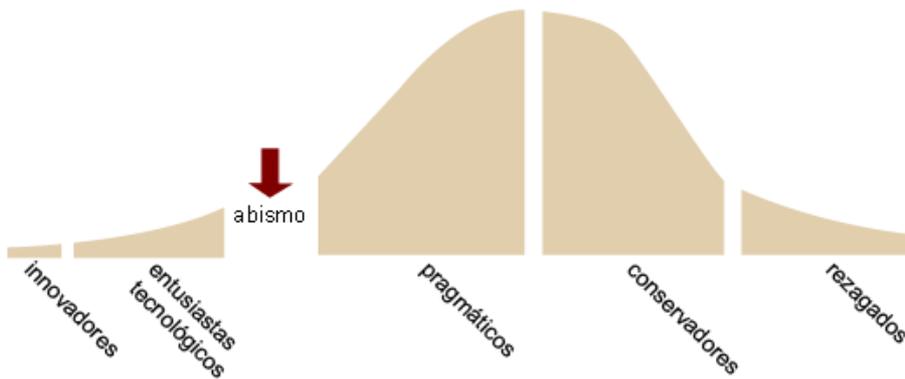
Detectar un nicho de mercado y elaborar un buen producto que satisfaga las necesidades del grupo de usuarios potenciales no es suficiente para lograr su aceptación. Para lograr introducir un nuevo producto o servicio, será fundamental tener en cuenta cómo son los patrones de adopción de tecnología entre un grupo de personas.

Los libros de marketing tradicionalmente han dibujado un modelo de adopción sobre la base de una curva de Gauss con cuatro grupos de usuarios:

- **Innovadores y primeros en probar (*early adopters*):** les gusta la tecnología y la innovación. A menudo adoptarán cierto producto sólo porque es nuevo.
- **Mayorías tempranas:** adoptarán una tecnología sólo si les ayuda a solucionar un problema concreto.
- **Mayorías tardías:** intentan evitar las nuevas tecnologías.
- **Rezagados:** serán los últimos en probar algo nuevo, o puede que nunca lleguen a probarlo.

La curva representa dos ideas clave: las dos categorías intermedias agrupan a la gran mayoría de los **clientes potenciales**, y sólo se puede conseguir captar a los grupos en orden, de izquierda a derecha (los "primeros en probar" la adoptarán si lo han hecho los innovadores, las mayorías tempranas, si lo han hecho los innovadores, las mayorías tardías, si lo han hecho las tempranas, y los rezagados, si lo han hecho las mayorías tardías).

Geoffrey Moore, en su libro *Crossing the Chasm* renombra los grupos, refiriéndose a ellos como **entusiastas tecnológicos, visionarios, pragmáticos, conservadores y escépticos**, y argumenta que esta teoría tiene un fallo, ya que la transición entre los entusiastas y la mayorías pragmáticas no es un continuo, y será difícil de conseguir. Las mayorías tempranas no adoptarán soluciones que no hayan sido ampliamente probadas, y lo harán de las que puedan obtener buenas referencias de otros pragmáticos, por lo que a veces alcanzarlas puede parecer una tarea imposible. Para Moore, existe un abismo (*chasm*) entre ambos grupos, por lo que redibujó la curva de la siguiente manera:



Curva de adopción tecnológica según Moore

Los innovadores y los entusiastas tecnológicos tienen una **alta tolerancia al riesgo** y a los defectos de la nueva tecnología, ya que cuentan con una importante habilidad técnica. Estos usuarios adoptarán una tecnología concreta sobre la base de la funcionalidad pura que presenten buscando la innovación. Las mayorías tempranas y tardías (pragmáticos y conservadores) tienen una **tolerancia baja al riesgo**, y estarán interesadas en adquirir un producto que les permita aumentar su productividad, pero que cuente con una alta estabilidad y madurez.

De esta manera, un producto innovador podrá tener un éxito importante entre los innovadores y los entusiastas tecnológicos, pero si la compañía creadora quiere ampliar su base de clientes, tendrá que poner en marcha una campaña de marketing distinta, poniendo énfasis no en las funcionalidades concretas y mejoras que ofrece el producto, sino en generar confianza sobre él, contando casos de éxito, implementaciones previas y número de usuarios.

Conseguir los primeros clientes en el grupo de los pragmáticos y mantenerlos contentos se hace fundamental, pero resulta muy difícil, dado el círculo vicioso que se crea: ninguno de ellos adoptará una solución que otros pragmáticos no hayan probado antes.

Se puede construir confianza ofreciendo soluciones completas, que incluyan mantenimiento, soporte y formación, para lograr atraer a clientes sensibles a la estabilidad y facilidad de uso del producto. Los primeros clientes de este grupo tendrán que ser tratados con cuidado, sin escatimar en tiempo o coste, ya que serán el punto de referencia para el resto. Una vez conseguidos unos pocos pragmáticos de referencia, atraer al resto será mucho más fácil, y una vez los pragmáticos hayan adoptado la solución, los conservadores seguirán sin requerir esfuerzos de marketing especiales.

Centrarse en los innovadores y entusiastas –por suponer que a pesar de ser un mercado potencial reducido, será suficiente para una pequeña empresa– puede ser peligroso, ya que este grupo es inestable por naturaleza, y abandonará un producto en cuanto deje de ser novedoso.

Esta curva de adopción también marcará el ciclo de vida del producto, sus dinámicas de desarrollo y sus prácticas de marketing. La empresa comercializadora debe tener claro en qué fase está y quiénes son sus clientes en ese momento, ya que cada grupo se siente atraído por muy distintos factores. Mientras que incluir muchas funcionalidades nuevas, y mantener un producto cambiante atraerá a los innovadores, los conservadores necesitarán que el producto simplemente funcione en unos escenarios concretos, y que lo haga siempre igual. Cada cambio supondrá una dificultad que sólo estarán dispuestos a superar si ello conlleva la solución de algún problema con el que se encuentren.

## 4. Función del producto: ¿Qué vender?

Considerar con cuidado la naturaleza del producto a desarrollar es muy importante. Una de las preguntas que habrá que resolver es si se pretende que el producto sea líder en el sector, un seguidor o un producto complementario.

Aunque en principio ser el líder del sector puede parecer lo más atractivo, puede no ser lo más efectivo. Al detectar la ausencia de una funcionalidad en un producto con una amplia adopción, una empresa podría tomar dos caminos: desarrollar su propia versión, incluyendo la funcionalidad ausente e intentar competir con el líder, o construir un complemento al líder que complete sus posibilidades.

La primera opción resultará muy complicada, y puede fracasar fácilmente, necesitando una inversión considerable no sólo en el nuevo desarrollo, sino en la campaña de marketing y ventas posterior. En la segunda, además de poder desarrollar el producto en menos tiempo, el trabajo de marketing estará ya hecho, en gran medida, por el líder, por lo que será mucho más fácil que el complemento gane adopción. Además, los usuarios conservadores (las mayorías) estarán mucho más dispuestos a incorporar un complemento a una solución conocida y probada, que a cambiar de tecnología y de proveedor. Un peligro común será que la empresa líder decida incorporar la funcionalidad desarrollada a su producto central, eliminando la necesidad de adquirir el complemento. En este sentido, la relación que se tenga con la empresa desarrolladora del producto central será fundamental.

Resulta importante, por lo tanto, definir el papel que tendrán otras empresas activas en el sector. Cuáles van a actuar como competidoras directas, cuáles como colaboradoras y cuáles, aun estando en el mismo sector, no entrarán a competir con nuestro producto por contar con una especialización concreta. Segmentando nichos y ofreciendo diferenciación, se puede evitar la competencia directa de empresas fuertes, y la existencia de empresas que elaboren productos o servicios relacionados puede ser un factor importante para el éxito.

A la hora de posicionar un producto, también será importante tener en cuenta para qué plataforma se va a desarrollar, es decir, qué conjunto de software básico es necesario para el funcionamiento del producto. Pensemos, por ejemplo, en la elección del sistema operativo y tecnología relacionada con el que operará la aplicación. Esta decisión afectará a la definición del nicho de mercado que se va a explotar, y a qué tipo de clientes se podrá dirigir, pero también será importante a la hora de definir su relación con aliados y competencia.

Una aplicación diseñada para funcionar en una plataforma determinada será una aplicación complementaria de dicha plataforma. Si se trata de un conjunto de software ya consolidado en el mercado y de amplia aceptación, se amplía también el mercado potencial de clientes, pero se reducen las posibilidades de encontrar aliados en los desarrolladores de la plataforma. El valor de estas plataformas vendrá dado, en gran medida, por el número y diversidad de aplicaciones que corran sobre ella, por lo que una empresa que esté intentando establecerse como líder de plataforma tendrá bastante interés en que se desarrollen aplicaciones relacionadas, y será, por lo tanto, una aliada más dispuesta.

Sin embargo, aunque más difícil, puede ser más interesante posicionarse como líder en un sector determinado. La pregunta, en este caso, será si se pretende crear una categoría de producto nueva para un nicho sin explotar, o si se va a intentar desplazar a otro producto existente.

#### **La segmentación y los clientes potenciales**

Para una empresa modesta, la única posibilidad puede ser la de ir segmentando el mercado, hasta encontrar un nicho concreto donde posicionarse. Puede ser difícil posicionarse como líder en las aplicaciones de planificación de recursos empresariales (ERP, *enterprise resource planning*), pero puede ser más fácil si se desarrolla un ERP para pymes, o un ERP para pymes hosteleras. Desde luego, al aumentar la segmentación, disminuirá la competencia, aunque también la base de clientes potenciales.

Ser los primeros en un mercado determinado sin duda proporcionará ventajas a la hora de posicionarse como líder, y a la hora de definir los estándares sobre los que se asentará esa tecnología, pero no proporciona ninguna garantía. La primera empresa en desarrollar una tecnología no siempre llega a convertirse en líder en el sector. A veces, llegar primeros y capturar a los entusiastas tecnológicos puede dar falsas indicaciones de éxito, ya que el producto deberá alcanzar a las mayorías para convertirse en líder. Las decisiones estratégicas y técnicas posteriores serán decisivas para determinar si la empresa es capaz de capitalizar las economías de escala de la demanda para posicionar a su producto como número uno en su sector.

Para conseguir irrumpir en un mercado que ya cuenta con un líder, harán falta campañas de marketing y ventas que a menudo no están al alcance de empresas de reciente creación. Sin embargo, el uso de un producto de software libre, que entre a competir con precio cero, puede ser un agente disruptor suficientemente potente. En próximos módulos veremos ésta y otras estrategias que proporciona el software libre para competir en distintos mercados.

## Resumen

Las necesidades en torno al software generan múltiples oportunidades de negocio a lo largo de su ciclo de vida, desde el propio desarrollo hasta servicios conexos como la instalación, la migración o la formación de los usuarios.

El posicionamiento empresarial resulta clave para determinar las posibilidades del negocio:

- La orientación a servicios proporciona un marco económico más estable al cabo del tiempo.
- La orientación al desarrollo de productos genera una economía de producto, más difícil de mantener en periodos largos.
- Los modelos híbridos pretenden ofrecer garantías de equilibrio a los dos modelos anteriores.
- La irrupción del software como servicio amenaza los modelos más tradicionales, ofreciendo una variación más versátil para los clientes potenciales.

Por otra parte, la explotación de vetas de mercado que son próximas y conocidas puede ayudar a la estrategia empresarial de un nuevo negocio, así como la adecuación del producto a los patrones de adopción tecnológica del mercado objetivo.

Finalmente, también será necesario establecer claramente la relación del negocio con sus competidores, así como la del producto con la de sus competidores. Estas relaciones pueden eventualmente favorecer la introducción del producto en el mercado objetivo.



## Bibliografía

**Christensen, C. M.** (1997). *The innovator's dilemma*. Harvard University Press <[http://books.google.es/books?id=Slaxi\\_qgq2gC](http://books.google.es/books?id=Slaxi_qgq2gC)> [Fecha de consulta: abril 2009]

**Christensen, C. M.; Raynor, M. E.** (2003). *The innovator's solution*. Harvard University Press. <<http://books.google.es/books?id=ZUsn9ulgkAUC>> [Fecha de consulta: abril 2009]

**Cusumano, M.** (2004). *The Business of Software Free Press*. Cambridge: Cambridge University Press. <<http://books.google.com/books?id=7KAW-ToDnBAC&dq=the+business+of+software&hl=es>> [Consulta: febrero 2009]

**Daffara, C.** (marzo, 2006). "Sustainability of FLOSS-based economic models". *II Open Source World Conference*. Malaga. <<http://www.cospa-project.org/Assets/resources/daffara-OSWC2.pdf>> [Fecha de consulta: abril 2009]

**McKenna, R.; Moore, G.** (2006). *Crossing the chasm* Capstone. <<http://books.google.com/books?id=GTwFAQAACAAJ&dq=crossing+the+chasm&hl=es>> [Consulta: febrero 2009]

**Sink, E.** (2006). *Eric Sink on the Business of Software* Apress. New Jersey: Princeton University Press <<http://books.google.com/books?id=h5IQengOGIC&dq=eric+sink+business+of+software>> [Consulta: febrero 2009]

