

MÁSTER EN ADMINISTRACIÓN Y DIRECCIÓN DE EMPRESAS

El sector del software libre en España: aspectos estratégicos y operativos

Centro Internacional de Postgrado
Universidad de Oviedo



Universidad de Oviedo

Universidá d'Uviéu

University of Oviedo

Autor: Tamara Bermejo Fidalgo

Tutor: Jesús García García

Julio 2020

Agradecimientos

En primer lugar, quiero dar las gracias a mi tutor, Jesús García García, por brindarme la oportunidad de realizar este Trabajo Fin de Máster y por su orientación, colaboración y entrega durante todo su desarrollo.

Por otro lado, no puedo olvidarme de todas aquellas empresas que, de forma desinteresada, han dedicado unos minutos de su tiempo a colaborar con la encuesta online que se les ha enviado, mostrando un gran interés por el objetivo del estudio.

Y por último, agradecer a mis familiares y amigos toda la ayuda y apoyo que me han dado a lo largo de estos meses de trabajo.

Índice General

| | |
|---|-----------|
| Introducción | 1 |
| Estructura del documento..... | 1 |
| Capítulo 1. El software libre | 3 |
| 1.1. Proyecto GNU..... | 4 |
| 1.2. Qué es el Software Libre..... | 6 |
| 1.2.1. Financiación..... | 8 |
| 1.2.2. Documentación Libre..... | 10 |
| 1.2.3. Open Source vs Software Libre..... | 11 |
| 1.2.4. Distribuir software libre por un precio..... | 12 |
| 1.2.5. Free Software Foundation..... | 13 |
| 1.2.6. The Linux Foundation..... | 15 |
| 1.3. Licencias..... | 17 |
| 1.3.1. Copyleft..... | 17 |
| 1.3.2. La Licencia Pública General (GPL)..... | 18 |
| 1.3.3. Licencia Apache 2.0..... | 19 |
| 1.3.4. Licencia Pública General Affero (AGPL)..... | 19 |
| 1.3.5. Creative Commons..... | 20 |
| 1.4. Modelo de negocio..... | 24 |
| 1.5. Motivaciones de los desarrolladores para su participación..... | 26 |
| 1.5.1. Peligros de la coexistencia del software libre y el privativo..... | 27 |
| 1.6. La extensión de la filosofía del software libre..... | 28 |
| 1.7. Debilidades, Amenazas, Fortalezas y Oportunidades (DAFO)..... | 29 |
| Capítulo 2. Estimación de costes en empresas de software | 35 |
| 2.1. El valor del conocimiento..... | 37 |
| 2.2. Algunos datos sobre innovación..... | 39 |
| 2.3. El Coste Total de Propiedad..... | 42 |
| 2.4. Modelos de estimación de costes..... | 47 |
| 2.4.1. La experiencia y madurez en proyectos similares..... | 47 |
| 2.4.2. Estimación por juicio de expertos..... | 48 |
| 2.4.3. Técnica Delphi..... | 48 |
| 2.4.4. Estimación por tres puntos..... | 50 |
| 2.4.5. Estimación de punto de función..... | 50 |
| 2.4.6. Modelo COCOMO..... | 51 |

| | |
|--|-----------|
| Capítulo 3. Metodología del estudio..... | 55 |
| 3.1. Objetivo de la encuesta | 55 |
| 3.2. Diseño | 56 |
| 3.2.1. Software utilizado para la encuesta | 56 |
| 3.2.2. Formato de los ítems | 57 |
| 3.2.3. Ítems de elección..... | 57 |
| 3.2.4. Ítems de construcción..... | 58 |
| 3.2.5. Redacción de los ítems..... | 59 |
| 3.2.6. Tipo de preguntas..... | 59 |
| 3.3. Recogida de los datos..... | 61 |
| 3.4. Envío de la encuesta..... | 62 |
| 3.5. Tasas de respuesta..... | 63 |
| 3.6. Problemas encontrados..... | 65 |
| Capítulo 4. Análisis de los datos..... | 66 |
| 4.1. Clasificación genérica de las empresas | 66 |
| 4.2. Financiación e ingresos | 68 |
| 4.3. La estrategia de posicionamiento | 69 |
| 4.4. Las funciones dentro de la empresa | 70 |
| 4.5. Planificación estratégica..... | 71 |
| 4.6. Sistema de información..... | 72 |
| 4.7. La información contable..... | 72 |
| 4.8. Subcontratación de servicios..... | 73 |
| 4.9. Contabilidad de costes..... | 73 |
| 4.10. Reutilización de código..... | 74 |
| 4.11. Donación al software libre | 75 |
| Capítulo 5. Conclusiones..... | 76 |
| Capítulo 6. Bibliografía..... | 80 |
| Anexo I: Encuesta..... | 86 |
| Anexo II: Listado de empresas participantes en el estudio | 91 |
| Anexo III: Glosario..... | 93 |

Índice de gráficos

| | |
|---|----|
| Gráfico 1. Clasificación de las empresas según el número de empleados | 66 |
| Gráfico 2. Clasificación de las empresas según el software comercializado | 67 |
| Gráfico 3. Clientes prioritarios de los servicios ofrecidos por la empresa | 67 |
| Gráfico 4. Fuentes de financiación más habituales | 68 |
| Gráfico 5. Principal método utilizado para generar ingresos | 69 |
| Gráfico 6. Funciones de la empresa | 70 |
| Gráfico 7. Cómo fluye la información en la planificación estratégica de la empresa..... | 71 |
| Gráfico 8. Principal sistema de información utilizado por la empresa..... | 72 |
| Gráfico 9. La información contable..... | 72 |
| Gráfico 10. Porcentajes de código libre reutilizado por la empresa..... | 74 |
| Gráfico 11. Porcentajes de código propio reutilizado por la empresa..... | 74 |

Índice de ilustraciones

| | |
|--|----|
| Ilustración 1. Modalidades de donaciones la Free Software Foundation | 15 |
| Ilustración 2. Ejemplo de uso de <i>Creative Commons</i> – Cultura Libre | 22 |
| Ilustración 3. Ejemplo de uso de <i>Creative Commons</i> - Ayuntamiento de Gijón | 22 |

Índice de tablas

| | |
|--|----|
| Tabla 1. Indicadores uso TIC en las empresas en 2019..... | 39 |
| Tabla 2. Gasto en innovación tecnológica y empresas innovadoras en 2017..... | 40 |
| Tabla 3. Software libre en empresas con menos de 10 empleados en 2017-2018 | 40 |
| Tabla 4. Razones por las que las empresas no utilizan software libre..... | 41 |

Introducción

En primer lugar, se inicia este Trabajo Fin de Máster con las **motivaciones** que me han llevado a la elección y desarrollo de este proyecto. De entre ellas, la principal razón ha sido conseguir unir lo aprendido durante el desarrollo del máster con mi formación anterior como Ingeniera Técnica en Informática de Gestión, ya que gracias a este trabajo podía unificar ambas disciplinas.

Mediante este estudio, he puesto en práctica diferentes aspectos estudiados en el Máster en Administración y Dirección de Empresas de entre los cuales cabría destacar el uso, mediante un caso real, de una de las técnicas de recogidas de datos analizadas durante el curso, la **encuesta**. También he podido afianzar y ampliar los conocimientos sobre **estimación de costes**, un campo explorado al inicio del curso y que quería desarrollar más en profundidad.

Además, otra de las finalidades del presente trabajo es la de ahondar en las características, usos y peculiaridades del **software libre** del que disponía de una pequeña aproximación adquirida a través de mis estudios técnicos, pero no conocía aspectos más amplios y su influencia en la economía.

Por último, los objetivos principales que se persiguen con el trabajo empírico a través de la encuesta *online* son analizar los aspectos estratégicos y operativos de las compañías dedicadas al desarrollo de software libre en España. De esta forma, se pretende conocer los objetivos que desean lograr estas empresas, así como las actividades que llevarán a cabo para alcanzarlos. Se busca averiguar cuáles son la suma de las capacidades internas de los encuestados y cómo han llevado a cabo la adaptación al cambio constante en el entorno. Todo ello, a través de diferentes cuestiones, al inicio más generales, para situar a la empresa, y ya más tarde, las propias de la materia que se desea investigar.

Estructura del documento

Analizando la estructura del documento se puede observar que, el primer capítulo está destinado a un **análisis teórico sobre el software libre**, incluyendo, desde algunos rasgos característicos de su fundador, hasta su filosofía y objetivos a alcanzar, para que alguien profano en la materia obtenga una primera toma de contacto con este tipo de software. Entre otros, también se abordan aspectos como la financiación, el tipo de licencias a elegir

para los diferentes proyectos desarrollados con código abierto, los modelos de negocio más habituales o las motivaciones que impulsan a los desarrolladores a participar en los proyectos. Por último, para cerrar esta parte teórica sobre el software libre, se realiza un análisis **DAFO** del mismo, donde descomponer sus debilidades, amenazas, fortalezas y oportunidades.

En el segundo bloque se lleva a cabo un **estudio sobre las empresas de base tecnológica**, principales participantes de la encuesta. Se detallan aspectos de estas compañías como: sus características comunes, fórmulas mediante las que obtienen su financiación o la importancia que en ellas tiene el conocimiento. Pero este capítulo aborda, principalmente, **la estimación de costes en empresas de software** a través de los diferentes modelos de estimación de costes y del cálculo del coste total de la propiedad.

El tercer bloque queda reservado para abordar todos los aspectos relativos a la **encuesta**, un instrumento de recogida de datos que ha sido fundamental para este trabajo. Al inicio, se abarcan puntos como los objetivos o su diseño, para posteriormente explicar un poco más la metodología seguida describiendo el software utilizado, el tipo de preguntas formuladas, cómo se ha realizado el envío o la tasa de respuesta obtenida finalmente.

En la siguiente sección se **analizan los datos adquiridos** mediante la encuesta construyendo una serie de gráficos de apoyo y calculando los porcentajes obtenidos para las diferentes cuestiones planteadas a los encuestados, comparando las respuestas recibidas.

Por último, tras el análisis es necesario finalizar este trabajo fin de máster con las **conclusiones** alcanzadas, tanto con el estudio teórico, como con el caso práctico mediante la encuesta. También se incluye la **bibliografía** con los libros, artículos, documentos y webs utilizados a lo largo del desarrollo del presente estudio.

Como **anexos** al documento se adjuntan la **encuesta**, con sus dieciocho preguntas y sus respectivas posibles respuestas, el **listado de las empresas** que han colaborado con sus opiniones y un **glosario** para la definición o explicación de diferentes términos y expresiones ordenados de forma alfabética.

Capítulo 1. El software libre

La industria del software es una de las más dinámicas, y en ella han ido surgiendo numerosas tendencias tecnológicas que pueden tener gran impacto en el desarrollo del mercado de los servicios relacionados con el software y del mercado en general, aumentando significativamente la competitividad industrial. Analizando su influencia en Europa, según el estudio *The Economic and Social Impact of Software and Services on Competitiveness and Innovation*, realizado por la Comisión Europea, este mercado representó en 2009 una cifra de negocios de 229 mil millones de euros y desde entonces ha experimentado un crecimiento anual medio del 1,5%, hasta el año 2016, este crecimiento deberá estabilizarse en el 2,9% entre 2015 y 2020. Según el mismo estudio, se prevé que para 2020 los servicios basados en software generarán una facturación de 290 millones de euros. Esta industria del software ofrece nuevas oportunidades, pero también presenta muchos desafíos que las empresas de software y la industria europea en general deben afrontar. Analizando el empleo en el sector del software en la Unión Europea este creció un 16,1% entre 2008 y 2013, en oposición a una disminución en el empleo en la economía empresarial total de alrededor del 3,4% y además, la alta productividad (medida en valor agregado por empleado) caracteriza a las empresas de servicios de software.

En cuanto a los segmentos altamente dinámicos en los que actualmente se está invirtiendo en esta industria del software, se encuentran ejemplos como: el **big data**, en el que las empresas encuentran grandes beneficios analizando los datos para ayudarles en la toma de decisiones; la **ciberseguridad**, aumentando la demanda de estas soluciones por la creciente proliferación de servicios digitales basados en la web, el uso intensificado de dispositivos móviles y el creciente número de ataques de delincuentes cibernéticos; el **internet de las cosas**, que ofrece un enorme potencial comercial para las empresas aunque este mercado está en sus primeras etapas, por lo que los nuevos modelos de negocio están aún en evolución; y por supuesto, el **software libre**, modelo productivo del que se hablará en profundidad a lo largo de este trabajo.

1.1. Proyecto GNU

En sus inicios, en la comunidad informática los programadores estaban acostumbrados a cooperar entre sí, ayudándose y compartiendo su trabajo, incluso las compañías informáticas distribuían habitualmente código, sin embargo, al inicio de los ochenta casi todo el software se había convertido en privativo, impidiendo esta cooperación entre programadores y poniendo de manifiesto la necesidad de que, en 1983, **Richard Stallman** comenzase con el **Proyecto GNU**, para devolver así esa libertad a los usuarios.

Cualquier usuario que necesite realizar tareas a través de un ordenador precisa disponer de un sistema operativo, por ello, la primera tarea en la lista del proyecto GNU fue implementar un sistema operativo libre, para ya más tarde, ir agregando el resto de funcionalidades necesarias para ofrecer este software como una alternativa al privativo. En su desarrollo el movimiento GNU decidió hacer el sistema operativo compatible con Unix, para así facilitar el paso a los usuarios de Unix a GNU, además de que era un sistema operativo portable y ya probado. El núcleo se creó a principios de los 90 por **Linus Torvalds**, con la ayuda, de entre otros muchos programadores de software libre, de Richard Stallman y basándose en Unix, de ahí que adquiriese el nombre de **Kernel Linux**. Este núcleo es, por supuesto, libre y gratuito y su código fuente está disponible para ser utilizado y modificado por cualquier usuario. En algunos casos, se llega a considerar que Unix y Linux son lo mismo, pero no es así, como se ha dicho, el *Kernel Linux* está basado en Unix y comparten algunos aspectos, pero difieren en otros muchos:

- Origen: Unix nació en los 70 en los Laboratorios Bell, de la famosa compañía AT&T, creando un sistema operativo para manejar servidores; el *Kernel Linux*, como ya se ha comentado antes, fue creado por Linus Torvalds a principios de los 90, generándose a partir de entonces muchas distribuciones basadas en Linux.
- Propiedad y derechos de autor: Unix es un sistema propietario que no se puede modificar, propiedad de la compañía AT&T que es la única que tiene permiso de modificarlo y actualizarlo; sin embargo, Linux está bajo licencia GNU y por lo tanto, es completamente libre y gratuito y cualquier persona puede modificar el código fuente, el cual está disponible para todos.
- Aplicaciones: Unix se utiliza principalmente en servidores, con la excepción del sistema operativo MacOS X, que es un sistema operativo de escritorio; Linux tiene

sistemas operativos tanto para servidores como para clientes. Dentro del mundo Linux hay muchas distribuciones, muchas de escritorio y gran variedad de herramientas creadas para ellos.

Para aclarar toda la confusión generada surgió la famosa frase “*Linux is not Unix*” y el acrónimo recursivo de GNU que significa “*GNU is Not Unix*”.

En 1985 se fundó la **Free Software Foundation** con el objetivo inicial de recaudar fondos precisamente para ayudar al Proyecto GNU, en apartados posteriores se analizará más en profundidad esta fundación y sus objetivos y logros.

Muchos programadores han querido colaborar en este movimiento del software libre, todos aquellos que estaban descontentos con la comercialización por grandes empresas del software. Según formula Stallman, desarrollar y utilizar GNU para él y sus seguidores es motivo de felicidad, mayor que la que les puede otorgar el dinero, que es el que suele mover a desarrollados privativos. Las aportaciones a este movimiento no son cosa del pasado, actualmente se puede encontrar en la página del Proyecto GNU diferentes formas de participación, como son:

- Ayudando en el desarrollo de algunos de los proyectos que tienen listados como prioritarios.
- Escribiendo manuales y otra documentación libre para software de GNU.
- Mejorando la accesibilidad del software libre.
- Ofreciendo paquetes de software que los usuarios han desarrollado por su cuenta con anterioridad, incluyéndolos como paquetes de GNU. Para ello, deben superar una evaluación de idoneidad comprobando que encaje perfectamente con el resto de paquetes GNU, se evalúa tanto técnica como filosóficamente, así como su calidad, utilidad y otras características. Basándose en el informe de los evaluadores, Richard Stallman es quién decide si aceptar o no el software propuesto.
- Incluso hay opción para que los estudiantes que tengan que realizar un proyecto de desarrollo de software para finalizar sus estudios participen en esta iniciativa.

1.2. Qué es el Software Libre

Tras describir la necesidad que algunos usuarios observaron en la creación del Proyecto GNU, se debe continuar con una introducción sobre qué es concretamente el software y cuáles son sus principios y peculiaridades.

Con software libre, Richard M. Stallman, se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software. Distingue cuatro clases de libertad que este software debe cumplir para poder calificarse como libre:

- **Libertad 0:** la libertad para ejecutar el programa sea cual sea el propósito del usuario.
- **Libertad 1:** la libertad para estudiar el funcionamiento del programa y adaptarlo a las necesidades de cada individuo, el acceso al código fuente es condición dispensable para esto.
- **Libertad 2:** la libertad para redistribuir copias y ayudar así al resto de usuarios.
- **Libertad 3:** la libertad para mejorar el programa y luego publicarlo para el bien de toda la comunidad, el acceso al código fuente vuelve a ser condición indispensable para ello.

Todo esto implica que los usuarios son libres para redistribuir copias con o sin modificaciones, de forma gratuita o cobrando por su distribución, a cualquiera y en cualquier lugar. Asimismo, el individuo es libre para introducir modificaciones y utilizarlas de forma privada, ya sea en el trabajo o en el tiempo libre, sin siquiera tener que mencionar su existencia. Esta libertad para utilizar un programa significa que cualquier individuo u organización podrá ejecutarlo desde cualquier sistema informático, con cualquier fin y sin obligación de comunicárselo ni al desarrollador ni a ninguna entidad en concreto. Se debe destacar que la accesibilidad del código fuente es una condición necesaria para el software libre, el usuario puede pagar o no por obtener copias de software libre, pero independientemente de la manera en que las obtenga, siempre tendrá la libertad para copiar, modificar e incluso vender esas copias.

El software libre no significa que sea “no comercial”. **Cualquier programa libre estará disponible para su uso, desarrollo y distribución comercial.** El desarrollo comercial del software libre ha dejado de ser excepcional y de hecho el software libre comercial es muy importante. No debe caerse en el error de pensar en el software libre como software gratis, porque entonces se caerá en el error de interpretarlo como una mera cuestión de precio y

no de libertad. Un buen ejemplo de “esta libertad sin restricciones comerciales” se encuentra en **Linux**, considerado como el proyecto de software de código abierto más importante del mundo y sin embargo, con una aceptación masiva en casi todos los sectores, incluidos los servicios financieros, las administraciones públicas o la educación entre otros. Linux también es el sistema operativo elegido para utilizar tecnologías de vanguardia como el internet de las cosas, la computación en la nube o el *big data*, que actualmente mueven gran cantidad de dinero. La propia **Linux Foundation**¹ ofrece algunos datos interesantes sobre la influencia de este sistema operativo en proyectos de carácter comercial donde se puede apreciar algunos ejemplos de la diversidad de sectores en los que se ha utilizado:

- Más del 80% de los nuevos teléfonos inteligentes que se comercializan actualmente funcionan con **Android**, que se basa en el *kernel* de Linux.
- Las 500 mejores supercomputadoras del mundo, utilizadas para cálculos muy específicos, funcionan con Linux.
- La mayoría de los mercados mundiales se ejecutan en Linux, incluida la Bolsa de Nueva York, NASDAQ, la Bolsa de Londres o la Bolsa de Tokio.
- Más del 75% de las empresas habilitadas para la nube utilizan Linux como su plataforma principal en la nube.
- Linux es la infraestructura de acceso de los líderes mundiales de comercio electrónico, incluyendo empresas como **Amazon, eBay y PayPal**.

Otro ejemplo de éxito a través de Linux son aquellas empresas que se han dedicado a **comercializar distribuciones personalizadas de este sistema operativo**. Una de ellas es **Red Hat, quien ofrece a sus usuarios sus servicios mediante el modelo de suscripción** a través de la cual los clientes tienen la capacidad de descargar e instalar productos de software libre a nivel empresarial, utilizarlos de manera eficiente y segura y recibir soporte técnico continuado en cada etapa del proceso. Según un estudio realizado por la empresa **International Data Corporation (IDC)** en el año 2019 las organizaciones de Tecnología de la Información (TI, o IT en inglés) que utilizan Red Hat Enterprise Linux pudieron prever ahorros del orden de los 6.700 millones de dólares sólo durante ese año 2019. Los ejecutivos de estas empresas colaboradoras en la encuesta destacan que este sistema:

- Disminuyó en un 52% el costo anual de software.

¹ The Linux Foundation es una organización sin ánimo de lucro que promueve el uso del código abierto con fines comerciales, www.linuxfoundation.org Se ampliará la información en el apartado correspondiente.

- Redujo en un 25% el tiempo que el personal de TI dedica a realizar tareas informáticas estándar o rutinarias.
- Recortó en un 5% los costes asociados con el tiempo improductivo imprevisto.

La IDC, tras su investigación, estima para el período 2018 – 2023 una tendencia con unos ingresos netos superiores a las 135.000 millones para los *partners* de Red Hat, de entre los cuales se pueden encontrar algunas multinacionales y otras que no lo son. La IDC concluye que todo esto significa que para 2023 las empresas, gracias al ecosistema creado por Red Hat, podrían invertir más de 35.000 millones en sus economías locales, provocando un gran impacto económico a largo plazo.

Por último, para cerrar este apartado, debe tenerse en cuenta que el término conlleva un **problema de ambigüedad** para las personas de habla inglesa, evocando un significado equivocado: software que se puede obtener de forma gratuita, a coste cero. La comunidad ha estudiado la posibilidad de modificar este término por otro correcto y menos ambiguo, pero finalmente las alternativas en inglés encontradas conllevan sus propios problemas. Cualquier sustituto para *free software* tiene un problema semántico parecido o peor, razón por la cual se ha decidido mantener el término. Más adelante se hablará de cómo a raíz de este dilema de ambigüedad ha nacido también el término *open source*.

1.2.1. Financiación

La financiación de proyectos de software libre puede realizarse mediante diferentes fórmulas, variando los mecanismos para conseguir recursos de un caso a otro, se pueden encontrar proyectos desarrollados íntegramente por programadores voluntarios, utilizando en este caso recursos cedidos altruistamente, hasta proyectos llevados a cabo por una empresa que factura el 100% de los costes a una entidad interesada en el desarrollo correspondiente.

Se debe destacar la **financiación pública** de proyectos de software libre, en las que la entidad financiadora puede ser directamente un gobierno (local, regional o nacional) o una institución pública, como una fundación. En estos casos, la financiación suele ser similar a la de los proyectos de investigación y desarrollo, y de hecho es muy habitual que provenga de entidades públicas promotoras de I+D. Normalmente, la entidad financiadora no busca recuperar la inversión, o al menos no de forma directa, aunque suele tener objetivos claros,

como: favorecer la creación de tejido industrial e investigador, promover cierta tecnología o tipo de aplicaciones, etc.

Con un mecanismo similar se puede encontrar también la **financiación privada sin ánimo de lucro**, donde la financiación la realizan normalmente fundaciones u organizaciones no gubernamentales. La motivación directa en estos casos suele ser producir software libre para su uso en algún ámbito que la entidad financiadora considera especialmente relevante, pero también puede encontrarse la motivación indirecta de contribuir a resolver un problema, por ejemplo, una fundación dedicada a promover una investigación puede financiar la construcción de un programa estadístico que ayude al análisis de los datos obtenidos y mejorar así sus conclusiones.

Otra modalidad es la **venta de bonos** para financiar un proyecto, la cual consiste en que el desarrollador (individual o empresa) tiene una idea para un nuevo producto software, tras analizar todas sus especificaciones, realiza la estimación del coste y emite bonos por dicho valor para su construcción. Cuando se hayan vendido suficientes bonos se comenzará el desarrollo, una vez finalizado y certificado por un tercero, el desarrollador “ejecuta” los bonos que había vendido pagando sus deudas, y lo que queda son los beneficios que obtiene por el desarrollo.

En algunos casos, los desarrolladores de software libre se unen en diferentes tipos de asociaciones, comúnmente **cooperativas de software**, funcionando de una forma similar a una empresa pero con un mayor compromiso ético con este movimiento, pudiendo combinar el trabajo voluntario con el remunerado. En el anexo dedicado a las empresas participantes en el estudio se pueden ver algunos ejemplos de esta vertiente de financiación.

Por último, se debe analizar la **financiación privada por empresas con ánimo de lucro** destacando entre todas ellas a **Microsoft, con Satya Nadella como CEO de la compañía**, que aunque pueda resultar extraño, es actualmente el **mayor aportante de código fuente libre**. Su presidente, **Brad Smith**, ha afirmado, en un evento en 2020 de la Massachusetts Institute of Technology (MIT)², que “*Microsoft estaba en el lado equivocado de la historia cuando el código abierto explotó a principios de siglo*”, Smith lleva más de 25 años en

² Massachusetts Institute of Technology (MIT) es una universidad privada localizada en Cambridge, Massachusetts (Estados Unidos) considerada por numerosos rankings como una de las mejores y más prestigiosas universidades a nivel mundial.

Microsoft, y fue de hecho uno de los principales abogados de la compañía durante sus batallas con el software libre. Para el presidente de la multinacional *“la buena noticia es que, si la vida es lo suficientemente larga, puedes aprender que necesitas cambiar”*. El desarrollo del software libre desde los años 80 y principios de los 90 se encontró con una fuerte oposición por parte de Microsoft. Todavía en 2008, el entonces consejero delegado de la compañía, Steve Ballmer, aseguró que *“los tres grandes competidores de Microsoft eran en publicidad Google, en hardware Apple y en nuestro negocio tradicional, el código abierto”*. Las cosas han cambiado, y el gigante del software, de hecho, es ahora el **mayor contribuyente a proyectos de código abierto en el mundo, superando a Facebook o Google** entre otros. Microsoft, además, con este cambio de filosofía, entró como **miembro platino de la Linux Foundation**, compartiendo este privilegiado grupo con sólo otras trece compañías, entre las cuales se encuentran ejemplos como: Intel, Google, Samsung o Huawei. Microsoft también participa en otras iniciativas relacionadas, como ser **patrocinador premium de la Open Source Initiative (OSI)**, organización encargada de la promoción del código abierto. Las circunstancias han propiciado este cambio por parte de Microsoft, viendo que el software libre se volvía imparable, y que disponía de una gran comunidad de desarrolladores y usuarios, decidiendo unirse a él, y aliarse con aquel que había considerado enemigo. Analizándolo desde la postura del software libre, esta colaboración de la gran multinacional debe valorarse también de forma positiva ya que estas aportaciones beneficiarán a toda la comunidad.

1.2.2. Documentación Libre

Dentro del movimiento del software libre se debe destacar la importancia de la documentación libre. La documentación en cualquier tipo de programa es un aspecto sumamente importante, tanto en el desarrollo de la aplicación como en el mantenimiento de la misma, de ahí que deba tenerse muy en cuenta dentro de la filosofía *free software*. La fase de documentación debe iniciarse a la vez que se comienza con la implementación del código y prolongarse hasta la finalización del mismo.

En el caso del software libre se puede apreciar aún mejor la necesidad de una correcta documentación, ya que personas distintas están estudiando y modificando un código desarrollado por terceros, una buena documentación puede ahorrar una gran cantidad de

tiempo en un nuevo desarrollo, y por lo tanto, mejorar la productividad de forma considerable.

Como ocurre con el software, la documentación libre no es un asunto de precio sino de libertad, el criterio es bastante parecido en un caso y en otro. El permiso de modificación es crucial, se debe tener en cuenta que cuando un usuario ejerce su derecho a modificar el software y añade o cambia sus características, si es riguroso deberá cambiar también su correspondiente manual, para suministrar información útil y precisa, sino se estaría distribuyendo un software con una documentación obsoleta y errónea.

Sin embargo, Stallman formula que, aunque es inaceptable la prohibición plena de esta modificación, si pueden establecerse ciertos límites sin ningún tipo de problema. El *free software* acepta mantener la nota de *copyright* original del autor, las condiciones de distribución o la lista de autores. Tampoco es un problema obligar a que en las versiones modificadas haya constancia de que han sido modificadas, o incluso tener secciones enteras que no pueden ser borradas o cambiadas, siempre que esas secciones traten sobre asuntos no técnicos. Estas restricciones no invalidan la esencia de que la documentación sea realmente libre, es decir, que dicha documentación se adapte a las modificaciones que haya sufrido el código. Un software libre necesita inequívocamente una documentación que también lo sea.

1.2.3. Open Source vs Software Libre

En 1998, parte de la comunidad de software libre decidió abandonar este término y empezó a hablar de código abierto. Con ello, algunos de sus defensores, trataban de evitar la confusión entre libre y gratuito que se ha explicado anteriormente. Sin embargo, otros intentaban abandonar los principios que habían impulsado la creación del software libre y el proyecto GNU. El término *open source* se asoció rápidamente con un enfoque distinto, una filosofía distinta, e incluso diferentes criterios para decidir qué licencias son aceptables.

Stallman explica que estos dos movimientos están en desacuerdo en los principios básicos, pero más o menos de acuerdo en las recomendaciones prácticas. Por ello, han trabajado juntos en muchos proyectos, sin considerar al *open source* como un enemigo, ya que en todo caso, el enemigo del software libre se encontraría en el software propietario. Como

Richard explica en su libro “*Software libre para una sociedad libre*”, la diferencia fundamental entre los dos movimientos está en sus valores, en su visión del mundo. Para el movimiento *open source*, la cuestión de si el software debe ser de fuente abierta es una cuestión práctica, no ética. El *open source* es un método de desarrollo, mientras que el software libre es un movimiento social. Stallman formula que para el movimiento *open source*, el software no libre es una solución ineficiente, para el movimiento de software libre el software no libre es un problema social y precisamente el software libre es la solución. Se podría concluir que la diferencia entre ambas serían cuestiones deontológicas.

1.2.4. Distribuir software libre por un precio

Aunque ya se ha mencionado ligeramente en la definición del software libre, en este apartado se pretende aclarar si es lícito o no el pago por software libre, ya que es un punto conflictivo para la perfecta comprensión del término.

Existe la creencia generalizada de que la distribución de copias de software libre debe realizarse gratis o a bajo coste (menor que el software propietario). Sin embargo, la comunidad de este tipo de software defiende que cada individuo cobre cuanto quiera o cuanto puedan, esto implica que en ocasiones estos programas se distribuyen gratis y en otras en cambio a un precio elevado. Debe puntualizarse que, el programa es libre sin que importe su precio, porque los usuarios lo pueden utilizar libremente.

Tampoco debe confundirse el hecho de conseguir una copia de un programa de forma gratuita con que sólo por esto sea software libre. Con o sin precio, el programa no es libre porque los usuarios no tienen libertad. Por lo tanto, se puede concluir que redistribuir software libre es una forma totalmente válida de conseguir beneficios con dicha actividad.

También es importante destacar que desde este movimiento se promueve la necesidad de que una parte de los beneficios que los distribuidores generan vuelvan a ser donados para el crecimiento de la comunidad, bien a través de la Free Software Foundation (FSF) o a cualquier otro proyecto que desarrolle software libre.

En resumen, Stallman defiende que la filosofía del software libre rechaza una práctica empresarial concreta y muy generalizada, pero no rechaza el negocio en general.

1.2.5. Free Software Foundation

El movimiento del software libre es uno de los movimientos sociales más exitosos de los últimos años, por ello, necesitaba una infraestructura que le aportase soporte logístico, legal y financiero, de ahí el nacimiento, en 1985, de la Free Software Foundation, una organización sin ánimo de lucro dedicada al desarrollo del software libre.

El número de personas que utilicen el software libre dependerá principalmente de las tareas que puedan desarrollar con él y su facilidad de manejo, ya que muchos usuarios seguirán usando software propietario si con el software libre no pueden realizar las labores que hasta ahora llevaban a cabo, tanto para su actividad profesional como en su tiempo libre. De ahí que uno de los mayores objetivos de esta comunidad es la conseguir mayor número de seguidores y así aumentar el desarrollo del software libre. Esta organización asegura la tan preciada libertad de los usuarios, mantiene artículos que cubren la filosofía del software libre y su definición, para mostrar claramente qué debe cumplir un programa de software concreto para que se le considere como tal. Piden el apoyo de su misión a los usuarios a través del uso del software libre en sus ordenadores y motivando a otros miembros de su comunidad para que también lo adopten en sus casas, trabajos, negocios etc. Además, disponen del boletín del *Promotor del Software Libre*, donde se enuncian las diferentes formas de participar con esta fundación.

La FSF tiene derechos de autor sobre una gran parte del sistema operativo GNU y muchos otros programas libres, llevan a cabo estas acciones para defender el software libre de los esfuerzos que se hacen con el fin de convertirlo en privativo. Cada año reúnen miles de asignaciones de derechos de autor de los desarrolladores de software individuales y empresas que trabajan en software libre. Registran estos derechos de autor con la oficina de derechos de autor de Estados Unidos y hacen cumplir la licencia bajo la cual distribuyen software libre, típicamente la Licencia Pública General GNU, GPL. Con ello lo que pretenden es asegurar que los distribuidores de software libre respeten sus obligaciones de transmitir a todos los usuarios la libertad de compartir, estudiar y modificar el código fuente de los programas.

La FSF utiliza la Licencia Pública General GNU (GNU GPL), la licencia de software libre más popular del mundo, y la única licencia escrita con el objeto expreso de promover y preservar la libertad del software. Otras licencias importantes que incluyen son la Licencia Pública General Menor GNU (GNU LGPL), la Licencia Pública General Affero GNU

(GNU AGPL) y la Licencia de Documentación Libre GNU (GNU FDL), más adelante se hablará sobre las licencias más utilizadas.

Según datos de la FSF disponen de más **3000 miembros activos repartidos por 48 países** representando un conjunto diverso de miembros, como: usuarios de ordenadores, artistas, ingenieros de software, *hackers*, estudiantes y activistas, ofreciendo a cada uno de ellos una razón por la que formar parte de esta organización:

- Como desarrollador de software, el software libre le permite construir y mejorar el trabajo de otros, como parte de una comunidad social construida sobre el principio de compartir.
- Como artista, con el software libre es posible hacer cosas que el software propietario no permite. Todo el software libre permite ser utilizado para cualquier fin.
- Como usuario, el software libre le saca de la lucha por el poder del software propietario, de manera que así es capaz de ayudarse a sí mismo y no depender de un solo desarrollador o empresa para ayudarle.
- Como estudiante, puede estudiar y modificar el software que utiliza, aprendiendo de él y mejorando las herramientas que utiliza para su propia educación.

En cuanto a su financiación, la FSF acepta donaciones de cualquier individuo o empresa, pero gran parte de sus ingresos proceden de la venta de copias de software libre, libros o manuales y de otros productos relacionados, ya que en su web se pueden adquirir productos como: camisetas, polos, gorras, llaveros, incluso un peluche del ñu, logo del proyecto GNU. Además, en esta web se puede encontrar una sección dónde realizar directamente una donación con diferentes modalidades en función de la cantidad monetaria aceptada, como se muestra en la *Ilustración 1*.

| | | |
|--|---|--|
| Supporting Member (with ThankGNU) - \$ 1,000.00 | Contributing Member (with ThankGNU) - \$ 500.00 | Go above & beyond - \$ 350.00 |
| Give a little more - \$ 200.00 | Annual Associate Member - \$ 120.00 | Annual Student Associate Member - \$ 60.00 |
| Monthly Associate Member - \$ 10.00 | Monthly Student Associate Member - \$ 5.00 | |

Please renew my membership automatically.

Increase your one-time or monthly contribution by

Total Amount \$ 1,000.00

Ilustración 1. Modalidades de donaciones la Free Software Foundation

1.2.6. The Linux Foundation

Aunque en el apartado anterior se han mencionado las donaciones que cualquier empresa o usuario puede realizar al movimiento del software libre a través de la FSF, la finalidad principal de esta fundación no es la de obtener financiación sino la de ejercer de guía a todos aquellos que deseen participar en la comunidad. Sin embargo, en el año 2007, de la mano de **Linus Torvalds**, nace la figura de la **Linux Foundation** un consorcio tecnológico sin ánimo de lucro que entre otras funciones, tiene la de lograr el crecimiento de Linux mediante aportaciones de grandes multinacionales.

Esta organización se dedica a construir ecosistemas sostenibles en torno a proyectos de código abierto para acelerar el desarrollo tecnológico. Además, ofrece soporte para las comunidades de código abierto a través de recursos financieros e intelectuales, infraestructura, servicios y eventos formativos. La Fundación Linux alberga muchos de los proyectos de código abierto más importantes del mundo, de entre los que se debe incluir al propio núcleo de Linux. Con más de **mil compañías respaldando a decenas de miles de desarrolladores activos**, sus proyectos aprovechan el poder del desarrollo de código abierto para impulsar la innovación a una velocidad y escala inigualables. **Los miembros corporativos de la Fundación Linux abarcan todos los sectores de la industria**, ejemplos de algunos de estos miembros son: **Fujitsu, IBM, Huawei, Accenture, Sony, Toyota, Uber, Adobe o Panasonic**. También disponen de simpatizantes individuales

cuyas donaciones financian programas que ayudan a proporcionar oportunidades para diferentes grupos. El **valor colectivo del código** en los proyectos de la Linux Foundation, según la propia fundación, se estima en aproximadamente **16 mil millones de dólares**, y crece cada día.

Se debe señalar que su Junta Directiva está compuesta por 22 líderes senior de toda la industria de la Tecnología de la Información, representando a la totalidad de los miembros de la fundación y a la comunidad de desarrolladores de Linux, estableciendo, además, la dirección estratégica de la organización. **Los componentes de esta junta proceden de empresas tan conocidas como: IBM, Samsung, Fujitsu, Hitachi, Facebook o Microsoft.**

Ejemplo de la influencia de esta organización a nivel empresarial son las declaraciones de Todd Moore, Vicepresidente de Tecnología Abierta en IBM, quien afirma que gracias a su participación en la Fundación Linux han realizado cosas que sin dicha asociación no hubieran sido posibles. Moore ha manifestado que si tuviesen que coger un proyecto, abrirlo ellos mismos y esperar que la gente fuese a colaborar en él, el camino sería muy difícil, sin embargo, gracias a la colaboración con la Linux Foundation ese recorrido se suaviza. De esta forma, se pone de manifiesto uno de los aspectos más sorprende y llamativo de este movimiento del software, y es que sea financiado por grandes empresas, algo que en un inicio puede parecer contradictorio, sin embargo, se consigue una simbiosis en la que tanto la comunidad como las empresas salen beneficiadas, con Linux y su filosofía colaborativa como base de todo. Se produce lo que podría definirse como una **transferencia de tecnología sin fricciones**, es decir, la Linux Foundation sirve de foro de encuentro entre diferentes compañías rivales en su sector pero que colaboran para crear tecnología al alcance de todos ellos, evitando los costosos acuerdos de colaboración y transferencia tecnológica que de otra manera serían necesarios para ello.

1.3. Licencias

El código desarrollado debe disponer de una licencia de software que permita establecer un **acuerdo legal que gobierne la relación entre el usuario y el software**, sin la cual nada le estaría permitido al usuario. Son un conjunto de permisos que un desarrollador le puede otorgar a un usuario en los que tiene la posibilidad de distribuir, usar o modificar el producto bajo una licencia determinada. Las licencias de software pueden establecer entre otras cosas: la cesión de determinados derechos del propietario al usuario final sobre una o varias copias del programa informático, los límites en la responsabilidad por fallos, el plazo de cesión de los derechos, el ámbito geográfico de validez del contrato e incluso pueden establecer determinados compromisos del usuario final hacia el propietario, tales como la no cesión del programa a terceros o la no reinstalación del programa en equipos distintos al que se instaló originalmente.

Dentro del software libre se pueden encontrar diferentes tipos de licencias, tanto para el software como para su documentación, siendo éste uno de los puntos más conflictivos y difíciles de abordar, ya que surgen diversas dudas sobre las ventajas e inconvenientes de unas licencias frente a otras, cuál es la más adecuada para un proyecto concreto, conocer si la licencia seleccionada se ajusta realmente a las necesidades del programa creado, etc. Por ello, desde la FSF se ofrece asesoramiento para seleccionar la mejor licencia para la obra realizada por el usuario, poniendo a su disposición diferentes artículos e incluso contacto directo para plantear las dudas a través de correo electrónico. En este apartado se expondrán las características de las licencias más utilizadas para despejar algunas de las incertidumbres habituales a la hora de licenciar los productos obtenidos a través del software libre.

1.3.1. Copyleft

En primer, lugar para comenzar a hablar de licencias debe introducirse el término de *copyleft*. Para asegurar que un software libre lo siga siendo en sus sucesivas versiones, surgió la necesidad de crear el concepto de *copyleft*. La forma más sencilla de conseguir que un programa sea libre sería ponerlo en el dominio público sin ningún tipo de derechos reservados. Esto permitiría a la gente compartir el programa y sus mejoras, pero también podría dar lugar a que dicho software se acabase convirtiendo en un programa propietario, con lo que finalmente no se estarían respetando las libertades que inicialmente había

otorgado el autor. De ahí **la necesidad del uso del *copyleft* para conservar la filosofía inicial del software desarrollado y elegida por el autor.**

De acuerdo con el *copyleft*, cualquiera que distribuya software, con o sin modificaciones, debe traspasar con él la libertad para copiarlo y modificarlo. El *copyleft* garantiza que cada usuario disfrute de esta libertad. Además, permite que los programadores contribuyan a mejorar el software libre, al otorgarles el permiso de modificar y distribuir el código.

Para aplicar el *copyleft* a un programa, primero se reservan los derechos declarando que dispone de *copyright*; luego se añaden los términos de distribución, un instrumento legal que otorga a todo el mundo el derecho a utilizar, modificar y redistribuir el código del programa o cualquier programa derivado del mismo, siempre que no se alteren los términos de distribución. De esta forma, el código y las libertades se convierten en elementos legalmente inseparables.

Los desarrolladores de software propietario usan el *copyright* para restar libertad a los usuarios; el software libre recurre a los derechos reservados para garantizar dicha libertad. Por eso han invertido el nombre, convirtiendo los derechos reservados, *copyright*, en *copyleft* siendo una forma de usar el *copyright* en un programa, por lo que no significa abandonar el *copyright*; de hecho, si así se hiciera, el uso del *copyleft* sería imposible.

Al tratarse de un concepto general no es posible utilizarlo directamente, sino una aplicación específica de dicho concepto y es cuando surgen diferentes tipos de licencias detallando a continuación algunas de las más usadas.

1.3.2. La Licencia Pública General (GPL)

Es la licencia de software libre más conocida y más utilizada, entre un 50% y un 70% del software libre se encuentra bajo esta licencia, incluso el propio núcleo de Linux. Es la primera de todas ellas y por ello, la que sirvió de modelo para todas las licencias posteriores, siendo derivadas de la GPL. Garantiza a los usuarios la libertad de usar, estudiar, compartir (copiar) y modificar el software. Su propósito es doble: declarar que el software cubierto por esta licencia es libre, y protegerlo (mediante el ya citado *copyleft*) de intentos de apropiación que restrinjan esas libertades a nuevos usuarios cada vez que la obra es distribuida, modificada o ampliada. Las obras cubiertas por esta licencia se pueden

comercializar por un precio establecido, permitiéndose su distribución comercial, o distribuirlo gratuitamente.

1.3.3. Licencia Apache 2.0

Cuando el programa que se desea licenciar no supera las 300 líneas de código, no merece la pena realizar el esfuerzo que requiere una licencia con *copyleft* por no aportar grandes beneficios debido a ese pequeño tamaño, por ello, desde el movimiento de software libre se aconseja utilizar la licencia Apache 2.0. En principio fue creada por la Apache Software Foundation para publicar los paquetes del proyecto Apache, sin embargo en la actualidad es usada para otros muchos proyectos. Se considera una licencia permisiva porque no requiere que los trabajos derivados sean publicados bajo la misma licencia y tampoco exige la liberación del código fuente. Los cambios a la versión original deben reflejarse en un archivo en el código fuente y pueden ser publicados usando cualquier otra licencia. Lo más importante de esta licencia es que los derechos de autor deben conservarse tanto en el código fuente como en los binarios. El movimiento GNU afirma que entre las licencias permisivas, la Apache 2.0 es la mejor. Así que, si por cualquier motivo se va a emplear una licencia permisiva, esta es la más recomendada.

1.3.4. Licencia Pública General Affero (AGPL)

Los términos de la licencia AGPL son prácticamente idénticos a los de la GPL; la única diferencia sustancial es que incluye una cláusula adicional para garantizar que las personas que utilizan el software en red puedan obtener el código fuente correspondiente, razón por la cual es recomendada para el software para servidores.

Además, se debe tener en cuenta, que cuando se contribuye a un proyecto existente, la versión modificada se debe publicar bajo la misma licencia que la obra original, facilitando la cooperación con quienes mantienen el proyecto, ya que usar una licencia distinta para las modificaciones puede dificultar dicha colaboración. Podría plantearse un cambio de licencia si se han realizado cambios significativos en la obra que estaba bajo una licencia sin *copyleft* y si la nueva versión creada es considerablemente más útil que el original, entonces merecería ser licenciada con *copyleft*. Si se opta por publicar contribuciones bajo una licencia distinta, el creador se debe asegurar de que la licencia original permite el uso

del material bajo la licencia que se ha elegido y siendo honesto con la filosofía de este movimiento se debería indicar qué partes del trabajo se encuentran bajo qué licencia.

1.3.5. Creative Commons

Por último, se debe destacar que el software libre ha influido también en otras organizaciones y corrientes con los que compartía ciertos puntos en común, ejemplo de ello son las llamadas *Creative Commons*.

Fruto de la aparición de Internet y del uso masivo de las nuevas tecnologías cada día se generan millones de datos produciendo, a su vez, la necesidad de proteger los derechos de los autores de toda esa información. Por ello, en el año 2001, con **Lawrence Lessig** como uno de sus principales actores, aparece una organización sin ánimo de lucro, muy ligada al software libre, denominada **Cultura Libre**. Su objetivo es el de apoyar y proteger a los creadores e innovadores mediante una serie de licencias dando un salto extensivo y cualitativo a través de nuevas fórmulas, para dar cabida, ya no solo al software, sino además a obras culturales (literarias, artísticas y científicas). Estas licencias han pasado a llamarse *Creative Commons*, naciendo para precisamente ayudar a los autores de obras de diferentes naturalezas a proteger sus derechos decidiendo qué pueden realizar los usuarios con su trabajo.

Lawrence Lessig, en su libro *Cultura Libre. Cómo los grandes medios usan la tecnología y las leyes para encerrar la cultura y controlar la creatividad*, formula los principios de este movimiento, él mismo explica en esta obra la influencia que, Richard Stallman y la Fundación de Software Libre, tienen en ella, calificándola incluso de obra derivada, inspirando su título y gran parte de la argumentación del libro. Lessig manifiesta que la cultura libre no es una cultura sin propiedad, no es una cultura en la que no se pague a los artistas por sus creaciones, sino que una cultura libre apoya y protege a creadores e innovadores. Lo hace directamente concediendo derechos de propiedad intelectual. Pero lo hace también indirectamente limitando el alcance de estos derechos, para garantizar que los creadores e innovadores que vengan más tarde sean tan libres como sea posible del control por parte del pasado. Sin embargo, Lawrence también puntualiza que cultura libre no es sinónimo de cultura gratis, producir un libro conlleva costes de derechos de autor, traducción, edición, corrección, maquetación, diseño e impresión, e indica cómo colaborar

en este movimiento a través de donaciones, aspectos que también son comunes a los que se encuentran en la Fundación de Software Libre.

Las *Creative Commons* ofrecen una versión más moderna del tradicional *copyright*. Por ello, si se hace uso de una obra protegida con estas licencias debe tenerse en cuenta que existen cuatro alternativas, pudiendo incluirse todas o solo algunas de ellas en la obra a la que se le desea otorgar la licencia y las normas que implican cada una de ellas. Esta licencia no significa que no disponga de derechos de autor, sino que el propio autor ha decidido cómo gestionar ese derecho, decidiendo entre las siguientes opciones:

- *Reconocimiento*: se permite copiar, reproducir, distribuir, comunicar públicamente, realizar obras derivadas y hacer uso comercial de ella, siempre y cuando, se mencione al autor.
- *No comercial*: no se puede utilizar con fines comerciales.
- *Sin obras derivadas*: se puede utilizar la obra pero no transformarla para crear una obra derivada de la original.
- *Compartir igual*: se puede realizar una obra derivada siempre y cuando se mantenga la misma licencia para su divulgación.

En las siguientes imágenes, se puede observar el uso de la licencia *Creative Commons* en el libro antes mencionado de Lawrence Lessig, donde se pueden ver las diferentes opciones que ha elegido el autor para autorizar su correcta utilización y en la página web del Ayuntamiento de Gijón, para visualizar de esta forma ejemplos en diferentes formatos:

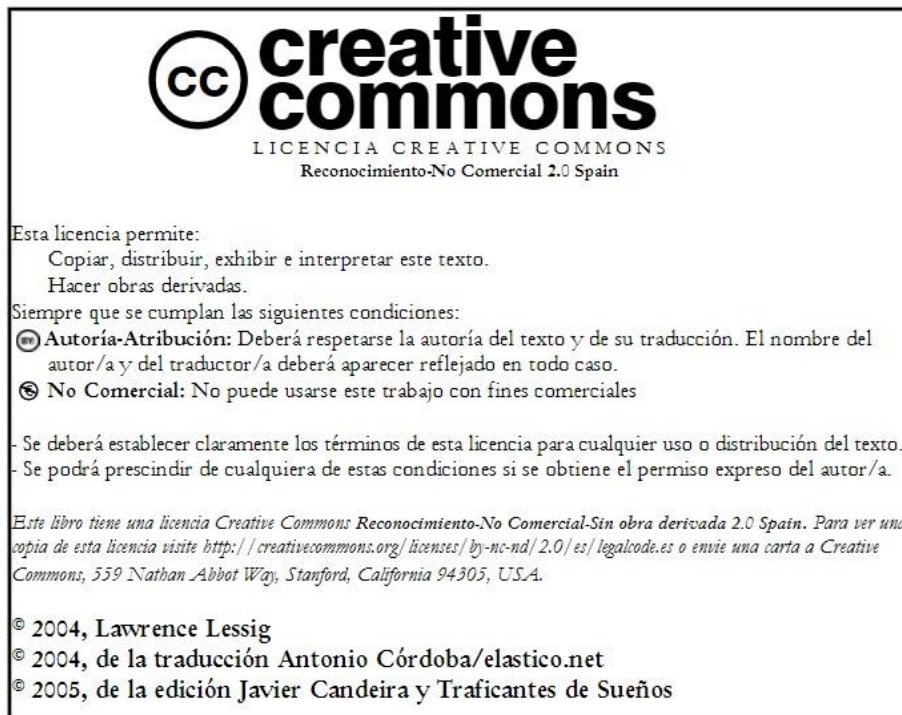


Ilustración 2. Ejemplo de uso de *Creative Commons* – Cultura Libre



Ilustración 3. Ejemplo de uso de *Creative Commons* - Ayuntamiento de Gijón

A su vez, también se pueden encontrar tres formas diferentes de expresar estas licencias: *Commons Deed*, consiste en un resumen del texto legal; *Legal Code*, se refleja el código legal completo; y por último, *Digital Code*, incluyendo el código digital que pueden leer los ordenadores para que motores de búsqueda y otras aplicaciones identifiquen el trabajo.

Por último, indicar que, aunque la duración de estas licencias se supone en principio indefinida, pueden producirse ciertas modificaciones como se puede ver en el siguiente fragmento del texto legal de las licencias:

“El licenciador se reserva el derecho a divulgar o publicar la obra o la prestación en condiciones distintas a las presentes, o de retirar la obra o la prestación en cualquier momento. No obstante, ello no supondrá dar por concluida esta licencia que continuará vigente y con efectos completos.”

1.4. Modelo de negocio

Tras comentar algunas de las licencias más conocidas y utilizadas dentro del movimiento del software libre, es hora de averiguar cómo las empresas pueden obtener beneficios a través de esta industria, cuáles serían las diferentes propuestas de valor que pueden ofrecer, olvidando la errónea idea de software libre igual a software gratuito.

La más obvia de todas las alternativas, es la **comercialización del software mediante el pago de licencias**. Como ya se ha dejado claro este software no tiene que ser gratuito, cobrar por su distribución no hace que sea menos libre, por lo tanto, la empresa puede obtener beneficios de su comercialización, utilizando la licencia más apropiada para ello. Un ejemplo de este modelo es el de las distribuciones GNU/Linux, que se venden a un precio inferior si se compara con el de sus homólogos privativos, pero superior al coste de una copia (incluso existiendo la posibilidad de bajarlo directamente desde nuestro PC a través de internet), participando entonces otros factores como la imagen de marca o la comodidad que supone para el consumidor.

Una variante del caso anterior sería la **doble licencia o freemium**, en la que se ofrece una versión gratuita y otra de pago, con el objetivo de que el usuario pruebe la opción sin coste, y le sea tan útil que decida adquirir la versión de pago, obteniendo así herramientas y funcionalidades adicionales.

Otra opción de negocio sería la **instalación y configuración** del software, obtenido el software de forma gratuita pero abonando la correcta implantación en la empresa.

Algunas compañías distribuyen su software de forma gratuita pero ofrecen el **soporte y servicio técnico** para cualquier fallo o consulta a través de una **suscripción** mensual o anual de pago.

Otras empresas se especializan en prestar servicios de **consultoría y asesoría** para elegir las aplicaciones que mejor se adapten a las necesidades observadas, incluso pasando a la siguiente fase, y si es necesario, **desarrollar aplicaciones hechas a medida** y totalmente personalizadas para la compañía. En este caso, el *target* o mercado objetivo suele ser el pequeño cliente, por lo que el crecimiento se basará en un gran volumen.

En algunos casos se ha apostado por construir una **imagen de marca** con buena reputación que aporte calidad y prestigio a los productos de la empresa, de tal forma que pueda comercializar sus productos a un precio más elevado, ya que los consumidores están

dispuestos a pagar ese sobrepago por el respaldo que esa imagen de calidad les ofrece, además de aportarles ciertos servicios adicionales.

Otra oportunidad de negocio dentro de esta industria sería la de ofrecer **formación** a los usuarios, bien a través de clases presenciales, tutoriales o plataformas de *e-learning*. Mediante esta formación se eliminaría uno de los puntos débiles que algunas compañías pueden detectar antes de invertir en software libre.

Como se puede ver en las diferentes alternativas planteadas, el modelo de negocio basado en licencias cada es más pequeño orientándose al mercado de venta de diferentes servicios, necesarios a su vez para el crecimiento de esta industria en general. Aunque en este documento se ha realizado una clasificación independiente, son muchas las empresas que utilizan **modelos de negocio mixtos**, que son combinaciones de las alternativas anteriormente descritas, aunando una parte más tradicional con otro más innovadora y arriesgada, ofreciendo a sus clientes mayor valor añadido a sus productos.

Gracias a estas opciones de negocio las empresas serán capaces de ofrecer una propuesta de valor clara, para así llegar más fácilmente al cliente, diferenciarse, y establecer con él una fuerte fidelización y relación de confianza y compromiso.

1.5. Motivaciones de los desarrolladores para su participación

El fenómeno del software de código abierto ha atraído la atención de investigadores y profesionales, sobre todo de campos como la economía o la psicología, para intentar revelar las motivaciones que los desarrolladores individuales tienen para participar en estos proyectos, que habitualmente no disponen de compensación monetaria para ellos como colaboradores. Enunciando algunas de las conclusiones a las que han llegado diferentes investigadores encontramos que Ghosh (1998) identifica que el disfrute y la creatividad son los motivos principales para la participación en proyectos de código abierto, Raymond (1999) argumenta que la reputación les da un incentivo para su colaboración. Otra de las motivaciones puestas en valor por los investigadores ha sido el propio deseo de los programadores de resolver el problema al que se enfrentan según Franke y von Hippel (2003) o las oportunidades futuras de conseguir un buen trabajo gracias a su contribución.

Parte la literatura dividió esta motivación para participar en proyectos de código abierto en dos grandes categorías: **motivaciones intrínsecas y extrínsecas**. La primera de ellas se define como la realización de una actividad por sus satisfacciones inherentes más que por alguna consecuencia separable. Específicamente, los psicólogos y sociólogos explican dos fuentes de motivación intrínseca: la participación en la tarea y la identificación de objetivos (Galbraith 1977, Staw 1989). En este caso, los desarrolladores disfrutaban más de hacer la tarea que de cualquier otra recompensa. Algunos economistas los denominan como agentes motivados u orientados a la misión (Besley y Ghatak 2005), otros ejemplos de ello pueden ser los policías promoviendo la justicia, los bomberos salvando vidas o los soldados defendiendo su país. También se debe destacar dentro de esta motivación intrínseca la satisfacción de pertenencia como miembro de una comunidad, el altruismo, la reciprocidad generalizada o la adopción de una actitud de entrega de regalos (Rossi 2004). Por otro lado, la motivación extrínseca se refiere a la que deriva de factores externos al propio individuo, recompensas como la reputación o las posibles ofertas de trabajo futuras, son algunos ejemplos de este tipo de motivación.

Lee y Kim (2012) estudiaron la decisión interna de los programadores entre el código abierto y los proyectos de software comercial desde una perspectiva económica, encontrando que el nivel óptimo de esfuerzo realizado por los programadores en el proyecto de código abierto aumenta con los niveles de motivación y habilidad. Con respecto al proyecto comercial, concluyeron que un programador de alta capacidad ejerce más esfuerzo que un programador de baja capacidad, pero se les paga por igual, lo cual

puede generar desmotivación y desánimo para el empleado. Con esto respaldan el resultado obtenido en su estudio afirmando que el proyecto de código abierto, a diferencia de lo que pueda parecer, es un mecanismo de compensación efectivo.

Sin embargo, a pesar de estos beneficios que las compañías pueden encontrar a través de estas nuevas motivaciones no se debe olvidar que también hay una serie de desafíos para las empresas que adoptan el enfoque de innovación abierta (Chesbrough, 2006). Las personas que participan en estas comunidades están más allá de los ámbitos jerárquicos de las empresas. Los desarrolladores de software libre pueden decidir dónde trabajar, con quién trabajar y en qué trabajar, lo que dificulta que las empresas dirijan la dirección del desarrollo (Dahlander y Wallin, 2006).

1.5.1. Peligros de la coexistencia del software libre y el privativo

Aunque los participantes en el desarrollo del software libre tienen muy claro sus principios, valores y motivaciones para llevarlo a cabo quizás con el paso del tiempo puedan encontrarse con ciertas dudas para continuar con su labor con la misma motivación y se puedan sentir tentados por la modalidad privativa, de ahí el peligro de la coexistencia del software libre y del privativo. Por ejemplo, puede ocurrir que mientras trabajan en un proyecto de código abierto dejen de contribuir libremente si creen que tienen una idea que podría generar una gran recompensa desarrollándola como software comercial, y quizás no tanto dentro del movimiento libre. También puede suceder que la motivación inicial vaya desapareciendo y que esa recompensa, en muchos casos solo emocional, deje de ser lo suficientemente fuerte como pago por su trabajo buscando otras alternativas para su desarrollo.

1.6. La extensión de la filosofía del software libre

Dada la evolución que ha seguido el software libre desde sus inicios puede existir la posibilidad de querer instaurar la filosofía seguida por dicho movimiento en otro tipo de industrias que no sea la del software, y así beneficiarse de toda la productividad y rentabilidad que se ha alcanzado gracias a este trabajo colaborativo. Se pueden encontrar algunas compañías en las que sí trabajan conjuntamente empresas con ánimo de lucro con otras sin fines lucrativos o que utilizan la experiencia del usuario para alcanzar ciertos puntos de innovación o que realizan intercambios de información con otras entidades para beneficio mutuo, todos ellos ejemplos de ingredientes utilizados e imprescindibles en el software libre. Sin embargo, otros aspectos serían difíciles de emular en otros modelos productivos, por ejemplo, habrá grandes proyectos que sean muy complicados de fragmentar en pequeños módulos manejables e independientes, o no exista tanto personal especializado para la colaboración. También puede darse como inconveniente que no sea suficiente sólo con ofrecer un buen funcionamiento y una interfaz amigable, sino que ciertos productos necesitan de costosos procesos de pruebas o regulaciones por parte de entidades externas para poder salir al mercado. Por último, otro de los aspectos que pueden frenar el uso de la filosofía libre es si se trata de un mercado masivo con muchos usuarios y poco especializados, que no van a ofrecer un reconocimiento por los servicios adquiridos, algo que los colaboradores del software libre necesitan para sentirse motivados y continuar con sus proyectos. Por todo ello, se puede apreciar que, en la mayoría de los casos, existen demasiadas barreras de entrada para poder instaurar la ideología libre en otros modelos productivos.

1.7. Debilidades, Amenazas, Fortalezas y Oportunidades (DAFO)

A continuación, se realizará un análisis DAFO del software libre, aplicando esta metodología de estudio para analizar sus características internas (debilidades y fortalezas) y su situación externa (amenazas y oportunidades).

DEBILIDADES

- Desconocimiento sobre qué es el software libre y qué ventajas aporta y es que, al hecho de pensar que el software libre debe ser gratuito, se extienden otra serie de falsos mitos, como: que no es seguro, está desarrollado por programadores *amateur* o que es perjudicial para la innovación, afirmaciones que, aunque ficticias, perjudicial gravemente que nuevos usuarios se animen a probarlo, y por lo tanto, al crecimiento de este tipo de software.
- Falta de formación a nivel de usuario. Al realizar la migración desde un sistema de software privativo hacia uno libre, la formación de los usuarios es un aspecto fundamental y básico para el éxito del cambio, lo cual supone un gasto considerable. Esta formación puede realizarse a través de:
 - Tutoriales distribuidos por la red.
 - Cursos o clases presenciales, suelen ser las más efectivas pero también las más costosas.
 - Plataformas de formación *e-learning*.
- Necesidad de gastos adicionales para la migración. Cuando se realice un cambio hacia la filosofía del software libre, se deben asumir unos costes adicionales, como el tiempo en el que los empleados o usuarios no disponen del sistema informático por dicha migración, la pérdida de productividad al inicio de su formación en el nuevo sistema, o la inactividad producida por algún fallo o error preliminar.
- En algunos casos, carencia de planes de soporte y mantenimiento, aunque en muchas ocasiones estos servicios se están añadiendo como una nueva fuente de ingresos para las empresas pero aún se observa esta insuficiencia en la actividad.

- Defectos de usabilidad que disminuyen el grado de facilidad de este tipo de soluciones. En algunos casos, esto suele deberse a que los desarrolladores de software construyen las aplicaciones pensando en otros desarrolladores cuyos cánones estéticos y de usabilidad varían respecto al resto de usuarios, dejando de lado en los proyectos y aplicaciones este aspecto e incluso no incorporando a diseñadores que se encarguen de ello.
- Pueden presentarse problemas de compatibilidad con algunas aplicaciones protegidas por *copyright*. En algunos casos, esta situación se puede resolver con algún software intermediario.
- Desconocimiento de los modelos de negocio. Muchas empresas se olvidan de las diferentes posibilidades que ofrece el software libre para obtener beneficios para sus compañías, al hecho de poder cobrar por la distribución del mismo, se suman actividades tan necesarias como el servicio técnico o el mantenimiento, en muchos casos, aunque el software se distribuya gratuitamente, se establece un sistema de suscripción que garantiza un pago mensual o anual obligatorio; otra versión es la llamada doble licencia o *freemium* que consiste en ofrecer una versión gratuita del producto y otra de pago, con la esperanza de que el cliente pruebe la gratuita y al resultarle útil decida actualizarla a la versión de pago, obteniendo así nuevas funcionalidades y herramientas; otra posibilidad es la de ofrecer un servicio de asesoría mediante el cual indicar qué aplicaciones son las más adecuadas al tipo de negocio del cliente o un análisis para un futuro desarrollo a medida construido con software libre.

FORTALEZAS

- Gracias a la reutilización el desarrollo se inicia desde unos mínimos y no desde cero, lo cual permite reducir tanto el tiempo de desarrollo como los costes del producto final, incrementando la productividad.
- Ahorro de tiempos de desarrollo. En el modelo de software propietario precisamente ese “secretismo” tecnológico es que el frena y ralentiza el desarrollo de nuevos productos o sus actualizaciones, algo que no ocurre en el software libre.

- Gran flexibilidad y personalización de las soluciones implementadas con software libre. El acceso libre al código permite realizar una personalización completa de la aplicación desarrollada, ajustándose así a las necesidades concretas de los usuarios.
- Sistemas como servidores web, servidores de correo o sistemas anti *spam*, desarrollados con software libre, están considerablemente implantados, debido a su robustez y seguridad. El acceso al código permite que tanto *hackers* como empresas de seguridad puedan auditar los programas, por lo que la existencia de agujeros de seguridad es mucho más difícil, aumentando la fiabilidad de estos sistemas.
- Compatibilidad y soporte de los productos a largo plazo. En ocasiones en la industria del software privativo una vez que un producto ha alcanzado un número de ventas determinado a la compañía no le interesa que los usuarios continúen con él, y sí con su nuevo producto, de ahí que dejen de dar soporte y conviertan en obsoleto el anterior. Sin embargo, esto no ocurrirá con el software resultante del movimiento de Richard Stallman.
- Detección y corrección rápida de errores. La comunidad de software libre a demostrado su rápida acción para subsanar los problemas encontrados gracias a la colaboración de sus desarrolladores, por ello, Linus Torvald afirmaba “*Dado un número suficientemente elevado de ojos, todos los errores se convertirán en obvios.*” Sin embargo, si se compara con el caso del software privativo, en este el tiempo de respuesta suele ser mayor e incluso, en ocasiones, llegando a negar la existencia del fallo inicialmente.

AMENAZAS

- Venta vinculada: cuando se adquiere un nuevo equipo éste viene provisto de software privativo, generalizando su uso, llevando incluso a que el cliente en el futuro se vea obligado a adquirir esas licencias mediante pagos adicionales.
- La resistencia al cambio, el esfuerzo adicional que supone un cambio de tecnología es una de las principales causas de las alternativas al software privativo sean

descartadas en muchos casos. Numerosos estudios han demostrado que la resistencia al cambio no es provocada por un cambio en el aspecto técnico (el cambio en sí), sino el cambio social que éste trae consigo. Los expertos de desarrollo organizacional, Ronken y Laurence (1952), en su artículo publicado para el Harvard Business Review, explican que la resistencia por parte del equipo operativo viene del miedo a lo desconocido, a las nuevas tecnologías, a que cambien las relaciones laborales del empleado, y su posición dentro de la empresa, incluyendo el miedo a perder su trabajo.

- Imagen mediática poco favorable, considerando el software libre como algo raro o extraño, vinculado sólo a ciertos grupos.
- La prevalencia de ciertos valores imperantes en la sociedad, relacionados con el consumismo, el atractivo de ciertas marcas donde su imagen es muy fuerte, ofreciendo un aval que garantiza su calidad y que los consumidores confíen en su producto, por lo que, se puede observar una falta de estrategia de marketing por parte del software libre.
- Falta de apoyo institucional, aunque actualmente se pueden encontrar iniciativas por parte de colectivos, como ayuntamientos, universidades y diferentes organismos, tanto públicos como privados, aún es necesario un mayor amparo del software libre dentro de las instituciones. Analizando el estudio realizado por el Centro Nacional de Referencia de Aplicación de las TIC basadas en fuentes abiertas (CENATIC) sobre *El software libre en los organismos públicos de ámbito estatal* en el año 2011, se puede observar que sólo el 40% disponen de software libre desplegado en sus servidores y que, en el caso de soluciones de escritorio, el porcentaje baja hasta el 15%, por lo tanto, pone de manifiesto la necesidad de un mayor empujón en el uso de esta tecnología dentro de las instituciones.

OPORTUNIDADES

- Ahorro de costes, si una empresa decide utilizar software libre en su organización puede observar un incremento de gastos al inicio, si tiene que formar a sus

empleados y adecuar el nuevo software a su organización, pero a la larga los costes se verán muy reducidos.

- Ventaja competitiva en cuanto al precio o a la velocidad de evolución de las soluciones, como ya se ha mencionado con anterioridad, se pueden reducir considerablemente la inversión destinada al software y gracias a la reutilización las soluciones se desarrollan a un mayor ritmo.
- Fomento de la innovación: el desarrollo de este tipo de software permite explorar nuevas tecnologías y procesos y metodologías innovadoras.
- Nuevas oportunidades de negocio, menos orientadas al producto y más a servicios.
- Independencia tecnológica de proveedores, permitiendo elegir la solución más adecuada a cada situación. Al evitar la dependencia de forma exclusiva posibilita que otras empresas del sector puedan proporcionar servicios de soporte sobre una aplicación, y por lo tanto, si el proveedor desaparece, el programa siempre podrá ser mantenido por otra empresa.
- Cambio social orientado hacia la solidaridad y el beneficio comunitario de los usuarios, favoreciendo el trabajo en equipo. Este modelo tiene similitudes con el utilizado tradicionalmente en el que campo científico, en el que la innovación y el conocimiento pertenecen al conjunto de la comunidad, no a una empresa concreta, llegando a obtener resultados brillantes que de otra forma sería imposible conseguir.
- Las empresas usuarias apuestan por la tecnología como valor diferencial respecto de sus competidores, integrando y aplicando el *know-how* adquirido a su sistema informático, un activo intangible muy valioso para la empresa y difícil de transmitir o copiar por otras compañías, de ahí su eficacia y utilidad para diferenciarse del resto.

- Fomento de la industria local. La utilización de software libre permitirá a las compañías no depender de las empresas extranjeras que comercializan software privativo, y sí recurrir a las empresas locales para el desarrollo y posteriores servicios relacionados, repercutiendo de esta manera en la industria local.

Capítulo 2. Estimación de costes en empresas de software

La naturaleza de las empresas que participan en el estudio se engloban dentro de las llamadas empresas de base tecnológica (EBT), estas compañías son aquellas que convierten el conocimiento científico o tecnológico en nuevos productos, procesos o servicios que se pueden introducir en el mercado. Algunas de las características en común que se puede encontrar entre ellas son:

- Pueden ser tanto empresas productoras de bienes como de servicios.
- Su competitividad se basa en que aplican su conocimiento a una innovación tecnológica.
- Cuentan con personal científico y/o técnico cualificado, con formación superior.
- En general son empresas con poco personal y que producen bienes y servicios con alto valor añadido.
- Su activo más importante es el *know-how* (conocimiento).
- Su gestión se apoya en nuevas tecnologías.
- Han desarrollado innovaciones muy recientes que han supuesto nuevos productos, procesos o servicios, o bien la mejora significativa de los ya existentes.
- Son empresas con capacidad para un crecimiento rápido pero al mismo tiempo tienen mayores dificultades en su gestión y una necesidad constante de innovación.

Una parte de las EBT son las llamadas *spin-off*, que surgen en el ámbito de otras entidades. Una *spin-off* es una empresa de nueva creación, con una estructura jurídica propia, pero surgida en el ámbito de otra organización o entidad mayor; por ejemplo, una empresa que surge en el ámbito de una Universidad o de un Centro Tecnológico. Las *spin-off* contribuyen a la transferencia del conocimiento y los resultados de la investigación al ámbito empresarial y por tanto al desarrollo económico de las regiones.

En la fase de captación de capital inicial este tipo de empresas obtienen, en muchos casos, financiación de agentes externos y relativamente desconocidos para el emprendedor, como:

- *Business Angels*: es un individuo que toma sus propias decisiones de inversión y que aporta su propio dinero, y en ocasiones su tiempo, a empresas no cotizadas promovidas por personas que le son ajenas.

- **Aceleradoras:** programas de ayuda al desarrollo de estas empresas que suelen abarcar financiación, mentoría y contactos. Además, pueden incluir un espacio físico en el que desarrollar la actividad durante el período de aceleración, el cual es limitado.
- **Capital semilla:** es un tipo de capital riesgo que se utiliza para apoyar la creación de nuevas empresas que surgen a través de los proyectos de emprendedores.
- **Aportación en especie:** compañías complementarias, proveedores normalmente, ofrecen sus servicios a cambio de una compensación en acciones en lugar de dineraria.
- **El *crowdfunding* o micromecenazgo:** mecanismo de financiación colectiva con el que una empresa recibe fondos para su proyecto por parte de multitud de inversores (en su mayoría particulares, aunque también pueden participar institucionales). Para ello, el emprendedor debe ponerse en contacto con alguna de las múltiples plataformas *online* que existen en el mercado y presentar su plan de negocio. Si supera los filtros de viabilidad, la plataforma hará pública su campaña. Estas campañas tienen un plazo definido y suelen incorporar un objetivo de captación mínimo. Si este no se alcanza, el dinero no se cobra (modelo *allornothing*). No obstante, algunas plataformas permiten que el emprendedor reciba el dinero, aunque no alcance el objetivo (modelo *keepitall*).
- **Subvenciones y créditos blandos gubernamentales.**

2.1. El valor del conocimiento

En este tipo de compañías se debe hacer un pequeño apunte sobre el valor del conocimiento, este bien que, aun siendo intangible, es algo que se puede potenciar y sobre todo valorar por las empresas. En los años 60, ya se adelantaba la importancia que tendría este concepto de conocimiento, a través de Peter Drucker, a quien se podría acuñar como uno de los padres de la gestión del mismo. El **conocimiento**, se entiende como aquellos datos procesados por las personas a los que se les agrega la experiencia para ser aplicados con sabiduría para resolver problemas. Por tanto, se puede entender la vital importancia que tiene la información para que este proceso tenga éxito ya que para que la sabiduría y la experiencia puedan gestionar los datos se debe contar con toda la información posible y esta debe moverse por la organización con libertad, a la vez que debe ser fácilmente accesible por todos.

Se pueden encontrar dos tipos de conocimientos. Por un lado, el **conocimiento implícito**, individual y subjetivo, de difícil difusión, codificación o formalización, y por otro lado, el **conocimiento explícito**, se puede extraer, difundir, clasificar y debe ser recompensado y valorado, siendo vital estimularlo y entrenarlo, se trata de todo aquello transmisible.

A continuación, es importante incluir la definición de la **gestión del conocimiento** en la empresa. En primer lugar, el término “gestión” se define como el proceso de planear, organizar, dirigir y controlar el uso de los recursos para lograr los objetivos organizacionales (Chiavenato, 2004, p. 91). Desde este punto de vista, la gestión del conocimiento debe cumplir con este concepto, entendiendo como recursos al conocimiento. Existen un sinnúmero de definiciones, por lo que es necesario tener presentes algunas de ellas para entender y establecer en forma práctica el significado de este término:

- Es el proceso sistemático de buscar, organizar, filtrar y presentar la información con el objetivo de mejorar la comprensión de las personas en una específica área de interés (Lavenport y Klahr, 1998, p. 195).
- Es la habilidad de desarrollar, mantener, influenciar y renovar los activos intangibles llamados capital de conocimiento o capital intelectual (Saint-Ouge, 1996, p. 45).

- Es el arte de crear valor con los activos intangibles de una organización (Sarvary, 1999, p. 107).
- Es el proceso sistemático de detectar, seleccionar, organizar, filtrar, presentar y usar la información por parte de los participantes de la empresa, con el objeto de explotar cooperativamente el recurso de conocimiento basado en el capital intelectual propio de las organizaciones, orientados a potenciar las competencias organizacionales y la generación de valor (Harvard Business Review, 2003, p. 7).

Esta gestión del conocimiento aporta a la empresa una serie de **beneficios**, como:

- Transferir de la forma más rápida y efectiva el conocimiento adquirido en la empresa, para que la información no pierda su valor.
- Utilizar al empleado como motor de la innovación; que la fuente principal sea interna.
- Identificar nuevas estrategias de experimentación y solución a los problemas.
- Transformar el conocimiento del personal en competencias corporativas y ventajas competitivas.
- Medir y compartir el efecto de cada uno de los empleados en la empresa.
- Aprender más rápidamente y sacar mejor provecho de la experiencia.
- Capitalizar el conocimiento creado, y lograr un retorno en la inversión en formación del personal, y en el conocimiento de nuestros clientes y competidores.
- Crear en la organización una cultura de mejora y aprendizaje continuos.

Por todo ello, la investigación científica deja de ser una externalidad, formando cada vez más parte de la actividad cotidiana de las empresas, razón por la cual, se internalizan sus costes. Requieren de mayor y creciente calificación, motivación, y creatividad en los trabajadores, para que éstos sean realmente productivos en dicha economía basada en el conocimiento.

2.2. Algunos datos sobre innovación

Dada la naturaleza de las empresas participantes, se ha incluido en este apartado algunos datos referidos a la innovación dentro de las empresas para que se sitúe un poco la importancia que se le está dando durante los últimos años, la cual cabe esperar que seguirá en aumento con el paso del tiempo. Además, se añade también información relativa al software libre en la empresa. Como fuente de datos se han empleado: la encuesta sobre *el uso de TIC y Comercio Electrónico (CE) en las empresas para el año 2018 y primer trimestre de 2019*, la encuesta sobre *Innovación en las empresas para el año 2018* y la encuesta *de uso de TIC y Comercio Electrónico (CE) en las empresas en 2017-2018*, todas ellas del Instituto Nacional de Estadística (INE).

En primer lugar, a través de la siguiente tabla (*Tabla 1*) se puede analizar información general sobre el uso de las Tecnologías de la Información y las Comunicaciones (TIC), comparando su influencia en las empresas de menos de 10 empleados y en las que superan esta cifra, se observa que el porcentaje de empresas que disponen de ordenadores y conexión a Internet es muy alta en ambos casos, existiendo mayores diferencias si se analizan otros aspectos como el uso de redes sociales o del comercio electrónico.

| | | Empresas con menos de 10 empleados | Empresas con más de 10 empleados |
|---|---|------------------------------------|----------------------------------|
| Disponen de ordenadores | 1 | 79,91 | 99,26 |
| Tiene conexión a internet | 1 | 76,31 | 98,39 |
| Tiene conexión a internet y página web | 2 | 30,21 | 78,16 |
| Utilizan medios sociales | 2 | 32,67 | 52,94 |
| Realizan ventas por comercio electrónico | 1 | 5,81 | 20,36 |
| Realizan compras por comercio electrónico | 1 | 17,52 | 33,90 |

1. Datos medidos en porcentaje sobre el total de empresas de cada tipo

2. Datos medidos en porcentaje sobre el total de empresas con conexión a internet de cada tipo

Tabla 1. Indicadores uso TIC en las empresas en 2019

Por otro lado, el INE afirma que el 92,8% de las empresas ha utilizado alguna medida de seguridad de las Tecnologías de la Información y las Comunicaciones (TIC) en el primer trimestre de 2019 y el 8,3% de las empresas realizaron análisis de *big data* durante el año 2018, el cual es 2,9 puntos inferior al del año anterior.

Con datos publicados en diciembre de 2019, también se ofrece el gasto en innovación en las empresas en 2018:

| | Valor |
|--|-------------------------|
| Gasto en actividades innovadoras | 18.688.949 ¹ |
| Porcentaje de empresas innovadoras | 20,0 |
| Porcentaje de empresas con innovaciones de producto | 9,8 |
| Porcentaje de empresas con innovaciones de procesos de negocio | 17,5 |

Tabla 2. Gasto en innovación tecnológica y empresas innovadoras en 2017

1.- Miles de euros

Por último, ya centrandó el análisis en el software libre en concreto, se ha analizado la encuesta realizada por el INE sobre *el uso de TIC y Comercio Electrónico (CE) en las empresas 2017-2018* extrayendo, entre todos sus resultados, los obtenidos sobre el uso del software libre en las empresas, en este caso para aquellas con menos de 10 empleados, obteniendo lo siguiente:

| Uso del software | Porcentaje |
|---|-------------------|
| Utilizan alguna tipología de código abierto | 64,80 |
| Sistemas operativos | 26,76 |
| Navegadores de Internet | 61,67 |
| Aplicaciones ofimáticas | 42,11 |
| Servidores web/Internet | 12,80 |
| ERP o CRM | 4,77 |
| Software de seguridad, plataformas de aprendizaje, servidores de correo electrónico | 19,99 |

Tabla 3. Software libre en empresas con menos de 10 empleados en 2017-2018

A través de esta misma fuente se pueden observar también las razones por las que las empresas encuestadas no utilizan software de código abierto:

| Razones para no utilizar software de código abierto | Porcentaje |
|---|-------------------|
| Por los problemas que puede conllevar la migración | 6,67 |
| Por la inercia del mercado | 6,52 |
| Por la percepción de falta de calidad de este tipo de soluciones | 5,42 |
| Por falta de apoyo o desconocimiento del mismo por parte de organismos públicos o las empresas para la adopción de software libre | 8,28 |

Tabla 4. Razones por las que las empresas no utilizan software libre

2.3. El Coste Total de Propiedad

En 1987, gracias al trabajo de **Bill Kirwin**, consultor con más de veinticinco años de experiencia en investigación y consultoría de TI, y otros analistas del Grupo Gartner, el término *Total Cost Ownership* (TCO) comenzó a ganar popularidad, como una metodología para estimar el coste total del ciclo de vida de un producto del sector de la Tecnología de la Información, llegando a convertirse con los años en un auténtico estándar en la industria, utilizándose para el cálculo de costes de casi cualquier equipo, y por tanto, vigente para la estimación del coste de la propiedad del software libre para una empresa y así conocer cuáles son esos costes si la compañía decide trasladar su sistema a esta filosofía. La metodología es utilizada tanto por quien toma la decisión de la inversión como por quien oferta los productos como argumento de venta (Echeverri, 2009).

El Coste Total de Propiedad (CTP), como también se conoce al TCO, puede ser calculado tanto antes como después de la adquisición o uso de un producto, resultando **muy útil para tomar decisiones que reduzcan los costes a largo plazo**. No siempre obtener el producto más económico resultará la mejor opción para la empresa o para el usuario, en muchas ocasiones comprar un paquete de mayor coste puede ser una opción óptima a largo plazo, por ello, a través del TCO, se podrá mejorar el estado de las compras internas y no caer en ese error inicial. Los compradores obtendrán un mayor conocimiento de qué costes son necesarios y cuáles no, consiguiendo una visión más amplia de las inversiones y de los costes que a posteriori se pueden generar de esa compra.

El principio básico del TCO es que los costes de propiedad de cualquier bien que se adquieran, **tienen componentes más allá de los estipulados en el precio de compra del mismo**, dado que a lo largo de la vida del activo se cometerán gastos adicionales, costes en los cuales se debe incurrir para garantizar el funcionamiento correcto, así como protegerlo, mantenerlo y administrarlo, durante la vida útil del mismo. Calcular el coste total de propiedad del software es muy complejo, requiere que se consideren muchos factores, incluida su compra, el mantenimiento y costes de actualización, compra de hardware y costes de mantenimiento, capacitación de personal y costes legales y administrativos (Russo et al. 2005). El concepto de TCO ofrece un análisis de todos los costes que supone la implantación de una tecnología, lo que refleja **una medida de los riesgos tecnológicos** a los que se expone una organización a la hora de realizar la inversión en su sistema de información. Constituye una visión del impacto de la tecnología en la empresa, desde un punto de vista financiero y organizativo.

Una vez comprendido el término de TCO es necesario conocer las diferentes **variables** que influyen en su cálculo, sin olvidar que el objetivo final de este análisis es la implantación en una compañía de un sistema desarrollado mediante software libre. Dichas variables pueden ser divididas en:

- **Costes de adquisición:** se incluirían todos los costes de compra tanto de hardware como de software, cuyas necesidades dependerán del tipo de organización, ya que no todas las empresas dispondrán de las mismas exigencias de información para su funcionamiento. Dentro del **hardware** se incluirán: los servidores, los dispositivos terminales con los que trabajan los usuarios, aparatos de impresión, de almacenamiento, etc. Si se analiza el **software** se hablará del sistema operativo, los motores de bases de datos, las aplicaciones de administración, de comercio electrónico o *Business Intelligence*, entre otras. Por último, dentro de esta categoría también se deben contemplar los llamados **costes suaves u ocultos** que son aquellos que no son imputables, ni directa ni indirectamente a ningún elemento que genere valor para la organización, un ejemplo de ellos pueden ser la adaptación de los usuarios al nuevo sistema de software implantado en la organización. Este tipo de costes están en todas las actividades y afectan a todas las áreas de la empresa. La identificación y la cuantificación de los mismos es determinante en la toma de decisiones. Identificar los costos ocultos puede incrementar en gran medida la rentabilidad y la productividad de una empresa o usuarios.
- **Costes de mantenimiento:** en este punto se deben incluir todos los gastos derivados de la administración diaria de los sistemas operativos, servidores, aplicaciones, bases de datos etc.
- **Costes de soporte técnico:** por último, se añaden los costes del soporte que el sistema necesitará para hacer frente a posibles fallos o incidencias durante el periodo de tiempo para el cual se esté calculando la inversión. Este soporte puede ser realizado por empleados propios de la organización o por personal externo.

Además, si se desea plantear una posible **fórmula para el cálculo del Coste Total de Propiedad** podría plantearse la siguiente:

$$\text{TCO} = \text{Costes directos} + \text{Costes indirectos}$$

Costes directos = Costes de licencias de software + Costes de hardware + Costes de soporte + Costes de formación

Costes indirectos = Costes de mantenimiento + *Downtime*

Detallando los costes incluidos en las anteriores fórmulas, se obtienen los siguientes **costes directos**:

- **Licencias de software:** se incluyen todas las licencias necesarias para desempeñar las labores diarias de la compañía. Como ya se ha mencionado anteriormente, en el caso del software libre las licencias pueden variar de coste, desde ser gratuitas hasta el precio que el autor haya decidido otorgarle a su obra.
- **Hardware:** todos los equipos para el correcto funcionamiento del sistema a implantar. Se debe tener en cuenta que generalmente un sistema basado en software libre será menos exigente, en cuanto a recursos hardware, que uno propietario, y por lo tanto, los costes en este caso serán menores.
- **Coste de soporte:** se incluyen aquellos derivados de la instalación, configuración y mantenimiento resultante. Al respecto, algunos autores argumentan que en la misma comunidad se encuentra la principal fuente de soporte (Yalta, 2007; Dahlander, 2008), y es que una posibilidad, si desean disminuir estos costes, es solicitar colaboración a la comunidad de software libre a través de sus foros y ayuda *online* y continuar así con la filosofía de colaboración promovida Richard Stallman.
- **Formación:** se trata de un coste imprescindible para que los usuarios puedan actuar perfectamente con el sistema, sacándole el máximo partido, sin dicha formación se puede llegar incluso al abandono o rechazo del sistema por parte de los usuarios.

Avanzando ahora hacia los **costes indirectos**, éstos son los más difíciles de identificar y calcular, ya que son los que se producen por fallos del sistema o falta de productividad de los usuarios, por ello, suelen planificarse como un porcentaje. Entre ellos están:

- **Costes de mantenimiento:** no deben confundirse con los de soporte, son los que surgen al solucionar un error o problema no previsto.

- **Costes de seguridad:** aquellos provocados por fallos de seguridad producidos por el tiempo de caída del sistema por un virus o una intrusión de terceros, o por la aplicación de parches y actualizaciones no previstas. En este caso también se debe tener en cuenta otro de los puntos fuertes del software libre, como es su alto grado de seguridad, proporcionando un marco de trabajo más eficiente y seguro que si lo comparamos con su equivalente privativo.
- **Costes de escalabilidad:** si se experimenta un crecimiento considerable de la productividad de la empresa, puede que su sistema necesite adecuarse a la nueva situación, bien a través de nuevo hardware, de servicios de soporte adicional o incluso de una reestructuración del sistema de información y es ahí donde se pone de manifiesto este tipo de coste.
- **Costes de disponibilidad:** son aquellos producidos por la falta de disponibilidad del sistema debido a la carencia de robustez y estabilidad, produciendo un retraso en las tareas diarias a desarrollar. Como otra de las grandes fortalezas del software libre es precisamente su estabilidad, minimizando de esta forma este tipo de costes indirectos.
- **Costes de inoperatividad del sistema:** son los contemplados durante el cambio desde el sistema antiguo al nuevo, es decir, si se realizase una migración de un sistema privativo a otro de software libre, en el cual se producirá un período en el que los usuarios no dispondrán de sistema informático a utilizar, además del coste producido por la pérdida de productividad del usuario al enfrentarse a un nuevo sistema. Aunque estos costes pueden reducirse en gran medida mediante una correcta planificación de la migración de sistema y de formación de empleados.

Por tanto, partiendo de que cada sistema es distinto, el TCO dependerá del tipo de proceso de la empresa y de la cantidad de tecnología necesaria para llevarlo a cabo. Por este motivo en ciertas ocasiones el cálculo puede verse afectado por la propia experiencia, el tipo de solución, de las negociaciones y de acuerdos puntuales con las empresas proveedoras. Con el Coste Total de Propiedad se intenta medir el impacto de los sistemas de información en la organización, procurando mejorar la rentabilidad a largo plazo minimizando los costes, no sólo iniciales, sino de todo el ciclo de vida del software. Para demostrar sus beneficios

se han realizado numerosos estudios, por ejemplo, Degraeve, Labro y Roodhooft (2000) utilizaron datos de TCO en una investigación con modelos de programación matemática para la selección de proveedores y de esta forma demostrar el potencial ahorro en costes que se podía observar en un entorno real. Por ello, el TCO se ha vuelto cada vez más importante a medida que las organizaciones buscan formas de mejorar, comprender y administrar sus costes en la selección de proveedores y el mantenimiento de sus relaciones con ellos (Bhutta y Huq, 2002).

Sin embargo como parte negativa, Anderson, Thomson y Wynstra (2000) demostraron que, en muchos casos, los gerentes de compras de las compañías parecen confiar más en información de precios que en la que pueden obtener del cálculo del TCO para la toma de decisiones. Por su parte, Ellram y Siferd (1998), identificaron algunos factores que actúan como barreras para la adopción del Coste Total de Propiedad, como son la propia resistencia del usuario y la complejidad que en ocasiones conlleva la información de costes.

2.4. Modelos de estimación de costes

En la encuesta enviada a las empresas desarrolladoras de software libre en una de las preguntas se les consultaba a las compañías qué modelo de estimación de costes utilizaban habitualmente, obteniendo, como se verá posteriormente, que más de la mitad de los participantes se decantaban por una de las opciones planteadas: “*La experiencia y madurez en proyectos similares*”, aun así, en este apartado se expondrán algunos de los más utilizados.

En primer lugar, se debe tener en cuenta que una **estimación tiene como objetivo medir la duración, los esfuerzos y los costes necesarios para realizar todas las actividades y obtener todos los productos asociados a un proyecto**. Cuando se estima, se indica el valor medio o más probable, el cual no siempre se va a cumplir, ya que cualquier estimación tiene una variabilidad.

Centrándonos en la estimación del coste software, ésta va a depender de muchos aspectos, el software es un producto abstracto, desarrollado en un proceso diferente al de la ingeniería convencional, porque cada software es único y tiene sus peculiaridades. Brooks (1987) afirma que, por definición, el software es complejo e irreducible, es decir, no se puede simplificar sin pérdida de información. Larry Putnam, autor pionero en el campo del modelado de procesos de software, apunta que la gestión del desarrollo del software considera la estimación como una actividad que tiene como objetivo responder a dos preguntas: ¿Cuánto costará? y ¿Cuánto tiempo llevará hacerlo? Se encontrarán multitud de aspectos que afectan a esta estimación, como son: la complejidad del propio proyecto, su estructura, el tamaño, qué recursos están involucrados, los riesgos asociados, etc. Por lo tanto, este proceso se vuelve difícil y **no es posible establecer un modelo universal al que recurrir en todos los casos**. Además, la rapidez de cambio que caracteriza a las Tecnologías de la Información y las metodologías de desarrollo de software son otros problemas para la “estabilización” de este proceso de estimación.

2.4.1. La experiencia y madurez en proyectos similares

Este modelo resulta de utilidad cuando no hay datos empíricos cuantitativos. Utiliza el conocimiento y la experiencia de los desarrolladores, obteniendo estimaciones basadas en la síntesis de los resultados obtenidos en proyectos anteriores en los que el desarrollador ha participado. El principal inconveniente de este método es que la calidad de la estimación

dependerá de la opinión del “desarrollador experto” y no hay forma de probar dicha opinión hasta que ya es demasiado tarde. Los años de experiencia en la materia no tienen por qué traducirse en todos los casos en habilidad para realizar la estimación.

2.4.2. Estimación por juicio de expertos

Este modelo, aunque no es el más recomendable, en muchas áreas es una parte importante de la información cuando las observaciones experimentales están limitadas (Utkin 2005). Consiste en que una persona que conozca y disponga de experiencia en la tarea estime sus costes medios, esperando que esa **ausencia de datos anteriores sea sustituida por la experiencia del experto**. Claramente esta estimación va a depender de la reputación del experto, la cual puede llegar a verse debilitada al tener que ser defendida ante un cronograma o presupuesto final, planteándose aquí uno de sus grandes inconvenientes. De todas formas, esto puede verse solventado a través del siguiente modelo, la técnica Delphi, donde en lugar de una única persona para la estimación, se hace uso de un grupo de expertos, utilizando el valor medio de los resultados obtenidos de sus opiniones.

2.4.3. Técnica Delphi

En este caso, como ya se ha mencionado, la estimación será llevada a cabo no por un solo experto sino por un grupo de ellos. Se interrogarán a estas personas a través de sucesivos cuestionarios, a fin de poner de manifiesto convergencias de opiniones para finalmente obtener una serie de consensos. Linston y Turoff (1975) definen la técnica Delphi como un método de estructuración de un proceso de comunicación grupal que es efectivo a la hora de permitir a un grupo de individuos, como un todo, tratar un problema complejo. La identificación de las personas que formarán parte del juicio de expertos es una parte crítica en este proceso, frente a lo cual Skjong y Wentworht (2000) proponen los siguientes criterios de selección:

- Experiencia en la realización de juicios y toma de decisiones basada en evidencia o experticia (grados, investigaciones, publicaciones, posición, experiencia y premios entre otras).
- Reputación en la comunidad.
- Disponibilidad y motivación para participar.
- Imparcialidad y cualidades inherentes como confianza en sí mismo y adaptabilidad.

Por otro lado, las características fundamentales de este método de estimación son:

- Conservar el anonimato de los expertos, no existiendo ningún contacto entre los participantes, solo el administrador o gestor podrá identificar a cada participante con sus respuestas.
- Iteración: se podrán realizar tantas rondas de preguntas/encuestas como sean necesarias.
- Retroalimentación controlada: solo se hará circular entre los participantes una parte controlada de la información, no habrá un flujo total de los datos obtenidos en la ronda anterior entre los diferentes expertos.
- Resultados estadísticos: la respuesta final obtenida del grupo puede ser presentada estadísticamente, mediante promedios y grado de dispersión.

Además, para que el método Delphi se realice correctamente deben seguirse las siguientes fases:

1. Formular el problema a través del cuestionario con preguntas precisas, cuantificables e independientes.
2. Seleccionar a los expertos en función de sus conocimientos sobre el tema, permaneciendo aislados y recogiendo sus respuestas de forma anónima.
3. Desarrollo práctico y explotación de los resultados: el cuestionario es enviado a cierto número de experto, teniendo en cuenta que se encontrarán cuestionarios no respondidos y algunos abandonos, por ello, la recomendación es que este grupo de colaboradores no sea inferior a 25 personas. En la segunda ronda de consulta, los colaboradores son informados sobre los resultados de la primera, debiendo dar una nueva respuesta y sobre todo justificarla si difiere mucho con el resto del grupo. Si es necesario un tercer turno de preguntas se pedirá a cada experto comentar los argumentos de los que disiente de la mayoría. Se repetirán los turnos de preguntas las veces que sea necesario hasta obtener las respuestas definitivas mediante una opinión consensuada.

Se debe destacar, que las estimaciones generadas son tan buenas como las obtenidas por modelos más costosos y que consumen más tiempo, aunque no se debe de olvidar que esta técnica se basa en gran medida de la intuición y experiencia de los expertos, quedando limitada por la disponibilidad de los mismos.

2.4.4. Estimación por tres puntos

Este método **aúna la técnica del juicio de expertos con el cálculo estadístico**, de tal forma, que a través de la experiencia y conocimientos de los expertos se puede determinar el valor medio y la variabilidad. Para ello, se solicita a los expertos que estimen tres valores para cada tarea a estudiar:

- Valor medio: coste que estimen más probable para la tarea.
- Valor optimista: coste estimado considerando que todo vaya bien.
- Valor pesimista: coste estimado valorando que surjan problemas, aquellas circunstancias que puedan ir mal y afecten a la tarea a estimar.

A partir de estos valores se calculará el valor medio y la variabilidad. El principal inconveniente de este modelo es que lleva implícito en él los riesgos, debido a que en el valor optimista y pesimista el experto está incluyendo el valor medio de los riesgos que afectan a la tarea. Por lo tanto, se debería, en la medida de lo posible, identificar esos riesgos y su efecto en la estimación, para de esta forma eliminarlos de ella y trasladarlos a la gestión de riesgos.

2.4.5. Estimación de punto de función

Este modelo propone para la estimación medir las funcionalidades del sistema (Pressman, 2009), evaluando no sólo los costes sino también aspectos como el número de componentes o las posibles líneas de código. Esta técnica necesita las siguientes entradas para llevar a cabo la estimación:

- Número de entradas externas por parte del usuario de otra aplicación.
- Número de salidas externas, como informes o pantallas.
- Número de consultas externas, serían las llamadas en línea con respuesta inmediata, como puede ser la realizada por otra aplicación.
- Número de archivos lógicos internos, como tablas o archivos de datos.
- Número de archivos de interfaz externos, datos externos a los que se accede a través de la propia aplicación.

De los datos obtenidos, cada elemento se cuenta, mide y clasifica en niveles de complejidad, de manera subjetiva, en simple, medio o complejo. Por último, se calculará una calificación total de puntos para finalizar la estimación de los costes del proyecto.

2.4.6. Modelo COCOMO

COCOMO (*Constructive Cost Model*) es un modelo para estimar el coste, el esfuerzo y el cronograma durante la planificación de proyectos de software. Fue publicado inicialmente por Barry Boehm en 1981, pero debido a los constantes avances en el mercado de desarrollo de software, posteriormente fue lanzada una versión actualizada denominada COCOMO II en 1995, convirtiéndose en el modelo de estimación de costes más ampliamente utilizado en el mundo.

Es un modelo abierto, que puede ser utilizado por cualquier organización, y hace un uso intensivo de los datos históricos. Se aplica a los desarrollos que siguen el **ciclo de vida en cascada** que, de forma resumida, se trata de un enfoque metodológico que ordena rigurosamente las etapas del proceso para el desarrollo de software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior. Dentro del modelo COCOMO se pueden encontrar tres modos de desarrollo: **orgánico, semilibre y rígido**, según sean las características de la aplicación y del entorno de desarrollo.

Modelo orgánico

El proyecto se desarrolla en **equipos relativamente pequeños**, en un entorno familiar, dentro de la propia empresa. Muchos de los integrantes del proyecto disponen de una **amplia experiencia en trabajos similares**, por lo tanto, gran parte de los miembros contribuirán al proyecto desde las etapas iniciales sin generar una sobrecarga de comunicación. Estas características provocarán un entorno relativamente relajado para que el software cumpla las especificaciones necesarias, es relativamente flexible en el cumplimiento de los requerimientos y especificaciones de cada fase y tiempos de entrega. Muy pocos proyectos de modo orgánico han desarrollado productos con más de 50 KLOC de nuevo software (donde 1 KLOC equivale a mil líneas de código fuente). En los casos de productos más extensos, han sido construidos frecuentemente a partir de software existente, recurriendo a la reutilización. Estas características permiten que los proyectos que se encuentran en este modo alcancen una **gran productividad**.

Modelo semilibre

Es un modelo para productos de software de **tamaño y complejidad media**. Las características de los proyectos se consideran intermedias a las de los modos orgánico y rígido. En este caso el equipo de desarrollo dispone de un **nivel de experiencia medio** en proyectos similares, con una **combinación de personal experto e inexperto**. Está constituido por personas con amplios conocimientos sólo en algunos aspectos. Además, los

sistemas presentan niveles variados de exigencia, algunas fases rigurosas y otras muy flexibles. Los desarrollos resultantes tienen un tamaño que llega a las 300 KLOG.

Modelo rígido

En esta clasificación están incluidos **proyectos de gran envergadura** que operan en un **ambiente complejo con altas restricciones** de hardware, software y procedimientos operacionales. Por lo tanto, el modelo rígido será utilizado en proyectos que deben desarrollarse bajo unas restricciones estrictas, su complejidad hace que los cambios o modificaciones sean poco posibles, así que el desarrollo del software debe ceñirse rigurosamente a las especificaciones y/o requerimientos. En este caso, la sobrecarga de comunicación inicial será importante debido al desconocimiento en el área de desarrollo. El equipo que interviene en el proyecto tiene un **conocimiento general** de los objetivos del mismo y una **moderada experiencia** en el tema. En general, el líder del proyecto destina en las primeras etapas a un grupo pequeño del equipo a las tareas de análisis, evitando así los problemas acarreados por la sobrecarga de comunicaciones. Una vez que se completa el diseño global del producto, se delegará a un equipo de programadores las tareas de diseño detallado, codificación y testeo.

A su vez, a cada uno de estos modelos se le puede aplicar tres métodos distintos de estimación: **básico, intermedio y detallado**. De los cuales se plantearán a continuación sus características fundamentales.

Modelo básico

Estima el esfuerzo y el tiempo empleado en el desarrollo de un proyecto de software usando dos variables predictivas denominadas factores de costo (*cost drivers*): el **tamaño del software y el modo de desarrollo**.

Modelo intermedio

Comparado con el modelo anterior, éste provee un **nivel de detalle y precisión superior**, por lo cual es más apropiado para la estimación de costes en etapas de mayor especificación. COCOMO intermedio incorpora un conjunto de **quince variables de predicción** que toman en cuenta las variaciones de costes no consideradas por COCOMO básico. Estos factores son agrupados en cuatro categorías:

- Atributos relativos al **software**:
 - Fiabilidad requerida del software, es decir, la probabilidad de que el software realice sus funciones satisfactoriamente en su próxima ejecución durante un periodo dado de tiempo.
 - Tamaño de la base de datos.
 - Complejidad del producto.
- Atributos relativos al **hardware**:
 - Limitaciones en el tiempo de ejecución.
 - Limitaciones de memoria principal.
 - Volatilidad de la máquina virtual, entendiéndose por máquina virtual el conjunto de hardware y software que el producto utiliza para realizar su tarea.
 - Tiempo de respuesta experimentado por el equipo que desarrolla el proyecto.
- Atributos del **personal**: capacidad y experiencia de los analistas, capacidad y experiencia en el tipo de software a desarrollar y en propio lenguaje de desarrollo por parte de los programadores.
- Atributos del **proyecto**: software usado para el desarrollo, lenguaje necesario para desarrollar el software, requisitos de planificación.

Modelo detallado

Este modelo puede procesar todas las características del proyecto para construir una estimación. Introduce dos características principales:

- Multiplicadores de esfuerzo sensitivos a la fase. Algunas fases se ven más afectadas que otras por los atributos. El modelo detallado proporciona un conjunto de multiplicadores de esfuerzo para cada atributo, lo cual ayudará a determinar la asignación del personal para cada fase del proyecto.
- Jerarquía del producto a tres niveles. Se definen tres niveles de producto. Estos son módulo, subsistema y sistema. La cuantificación se realiza al nivel apropiado, es decir, al nivel al que es más susceptible la variación.

A modo de resumen, señalar que el modelo COCOMO es uno de los sistemas de estimación de costes más utilizados en proyectos de desarrollo de software. La

estandarización de su uso y la facilidad de la aplicación del mismo junto con la aproximación al coste real, han convertido a este modelo en uno de los referentes en este tipo de proyectos. Sin embargo, a priori no se debe preferir ninguna técnica en concreto, sino aquella que resulte más adecuada para el proyecto y para los recursos del mismo, o incluso se pueden utilizar más de un modelo para a posteriori determinar qué puede motivar esa variación de estimaciones y llegar a obtener una mayor comprensión de los factores que determinan los diferentes costes. Las mejores estimaciones son las que conducen a decisiones de negocio correctas (McConnell, 2006). Las buenas estimaciones serán aquellas que proporcionan la aproximación necesaria para que el presupuesto y el esfuerzo se establezcan con un margen adecuado.

Capítulo 3. Metodología del estudio

La investigación mediante encuestas tiene por objetivo la obtención de información de un grupo de sujetos (muestra) que debe ser representativo de un colectivo más amplio (población).

Por lo tanto, se entiende como población el conjunto de todos los objetos que comparten una propiedad bien determinada en función de la cual se definen, es decir, que comparten una característica que los identifica.

En el caso de este trabajo tomaremos como **población** a empresas que dentro de sus actividades esté la del desarrollo del software libre y que dispongan de alguna de sus sedes en España.

Todas las encuestas se realizan en el contexto de la vida cotidiana y, mediante un conjunto de preguntas, consiguen medir cuantitativamente características objetivas y/o subjetivas de la población. A ese conjunto de preguntas se le conoce como **cuestionario**. Su construcción es clave en el proceso de investigación mediante encuestas al ser el instrumento de recogida de datos que operativiza las variables objeto de investigación.

3.1. Objetivo de la encuesta

Como paso previo a la construcción de un test es necesario tener muy claro cuál va a ser el objetivo, es decir, que es lo que se quiere medir, a quién y para qué se quiere medir.

Con los resultados obtenidos mediante esta encuesta se pretende analizar cómo las empresas dedicadas al desarrollo de software, y más concretamente al desarrollo de software libre, llevan a cabo sus objetivos a través de diferentes actividades, conociendo los aspectos estratégicos y operativos de estas compañías, utilizando inicialmente preguntas más genéricas, con la finalidad de clasificar a la empresa, para posteriormente centrar las cuestiones en el objetivo principal del estudio.

3.2. Diseño

Todas las preguntas del cuestionario utilizado en este trabajo son de elaboración propia, es decir, no se han utilizado cuestiones de encuestas realizadas con anterioridad por terceros.

Dado que las empresas participantes en esta investigación están repartidas por las diferentes comunidades españolas, se decidió llevar a cabo la recogida de datos mediante un cuestionario *online*. Dicho cuestionario fue enviado por **correo electrónico** a sus destinatarios junto con un texto explicando el objetivo del estudio y la importancia de su participación en el mismo.

El uso de Internet para la recogida de datos ofrece diferentes ventajas, como:

- La rapidez con la que el usuario puede rellenar el cuestionario.
- Es una solución más económica que otras alternativas como la encuesta por correo postal o la telefónica.
- Proporcionan cierta intimidad, al no tener al encuestador cara a cara, lo cual se traduce en una mayor honestidad a la hora de contestar las preguntas.
- El tipo de herramienta utilizada ofrece la posibilidad de emplear técnicas de corrección automática, como por ejemplo que los usuarios respondan a todas las preguntas consideradas como obligatorias, o que solo marquen una de las opciones en lugar de varias cuando la pregunta disponga de este tipo de limitación.

Uno de los principales inconvenientes de las encuestas vía correo electrónico es la restricción a su acceso, reduciendo la muestra solo a la población que emplee este tipo de herramienta. Sin embargo, en el caso del presente estudio esto no supone una desventaja ya que los destinatarios del cuestionario utilizan el correo electrónico a diario como herramienta de trabajo.

3.2.1. Software utilizado para la encuesta

Para la realización de la encuesta se decidió utilizar los *Formularios de Google*. Este software permite emplear diferentes tipos de respuestas en cada una de las preguntas:

- Selección múltiple.
- Casillas de verificación.
- Elegir de una lista mediante un desplegable.
- Escala.

- Cuadrícula.
- Tipo fecha y hora.

Entre sus ventajas se encuentran:

- Herramienta fácil de utilizar, sin necesidad de experiencia previa.
- Permite recopilar de forma sencilla y eficiente la información. La propia aplicación almacena, de forma organizada, las respuestas en un archivo de hoja de cálculo, que posteriormente puede ser modificado para el análisis de los datos obtenidos.
- Software gratuito, sólo es necesario disponer de un usuario de Google.

Y sus principales inconvenientes:

- Se debe disponer de conexión a Internet para utilizar esta herramienta.
- Presenta ciertas limitaciones relacionadas con la personalización y el diseño de la encuesta.

3.2.2. Formato de los ítems

Entre estos primeros pasos necesarios para la elaboración del test está seleccionar el tipo de ítems que se van a utilizar para construirlo teniendo en cuenta que, como señala Osterlind (1998), el ítem puede ser considerado como una unidad de medida dentro del test global y puede aparecer bajo diferentes formatos.

Haladyna (1994), considera que una primera aproximación es establecer dos grandes categorías de ítems en función del tipo de respuesta que exijan de los participantes: ítems de elección e ítems de construcción.

Los ítems de elección son ítems de respuesta cerrada, en ellos se obliga a los sujetos a que respondan eligiendo una o varias alternativas de entre las propuestas. En los ítems de construcción el sujeto deberá elaborar su propia respuesta.

3.2.3. Ítems de elección

Los formatos más comunes para los ítems de elección son:

- *Dos alternativas*: se debe elegir entre dos alternativas. Por ejemplo: Verdadero/Falso, Sí/No, Correcto/Incorrecto.

- *Elección múltiple*: este tipo de ítems consta, por un lado, del enunciado propiamente dicho, y por otro, de las alternativas u opciones de respuesta. Muñiz y García Mendoza (2002) muestran una clasificación a su vez de este tipo de ítems en función de la estructura que tenga el enunciado, el cual puede presentarse en forma interrogativa, enunciativa o como una frase truncada o incompleta. Cada una de estas formas darán lugar a un tipo de ítem.
- *Emparejamiento*: este formato implica que el sujeto empareje los elementos de dos columnas de acuerdo a las instrucciones dadas en el enunciado.
- *Formato Cloze o incompleto*: en este tipo de ítems se ofrece, por ejemplo, un párrafo o una frase en la que faltan algunas palabras y aparece un espacio en blanco en su lugar, a continuación, se ofrece una lista en la que se incluyen las palabras que faltan. La tarea de los usuarios consiste en seleccionar, de la lista de palabras que se ofrece, la que corresponda a cada espacio en blanco.
- *Escalas de clasificación*: se trata de un tipo de formato de ítems en el que se presenta un enunciado y distintas alternativas de respuesta que están ordenadas de forma gradual en una serie de categorías a lo largo de un continuo. El sujeto debe responder eligiendo, de entre las alternativas propuestas, aquella que mejor refleje su postura o actitud personal ante el enunciado.

3.2.4. Ítems de construcción

En este tipo de ítems es el propio sujeto el que elabora su respuesta, de ahí que se denominen de respuesta abierta. Dentro de esta categoría se puede distinguir entre los de respuesta corta y los de respuesta extensa o de ensayo.

- *Ítems de respuesta corta*: a veces no son más que modificaciones de los ítems de elección múltiple ya que se ha de elegir una única palabra, pero, en lugar de elegir la de entre una serie de alternativas que se ofrecen, la tiene que construir el usuario, otras veces el sujeto tiene que responder con una frase.
- *Ítems de respuesta extensa o ensayo*: se pide que desarrollen por ejemplo un tema.

3.2.5. Redacción de los ítems

Algunas recomendaciones que han de tenerse en cuenta a la hora de redactar los ítems que componen la encuesta:

- *Evitar la ambigüedad de los enunciados*
Redactándolos de forma clara. El significado de las palabras empleadas debe estar claro para todos los sujetos ya que difícilmente serían comparables sus respuestas si cada uno pudiera interpretar de manera distinta el significado del enunciado.
Los enunciados cortos y directos también contribuyen a evitar ambigüedad ya que la inclusión de palabras innecesarias complican la lectura y pueden provocar confusión en los sujetos (Payne, 1951).
- *Evitar enunciados que provoquen respuestas sesgadas*
Se deben evitar los enunciados que puedan provocar una respuesta sesgada, entendiendo por sesgada aquella que es más probable que elijan los sujetos independientemente de su opinión.
- *Expresar una única idea en el enunciado*
Es necesario evitar las dobles pregunta en un mismo enunciado ya que provocaría confusión en los sujetos y no sabrían qué respuesta emitir.

3.2.6. Tipo de preguntas

Para la elaboración del cuestionario se han utilizado preguntas objetivas dirigidas a recopilar datos para categorizar y cuantificar ciertos aspectos de las compañías, en ningún caso, se han utilizado preguntas de tipo subjetivo, ya que con ellas se miden percepciones, sentimientos o juicios, que no encajan con el objetivo del estudio.

En su mayoría se han empleado preguntas con **respuestas cerradas de elección única**, en las que el encuestado tiene que elegir una opción de entre las alternativas establecidas, en algunos casos dicotómicas, que se responden con un sí o un no, pero en general, las respuestas son politómicas, presentando varias alternativas al usuario para que elija la más conveniente para él. En la encuesta también se utiliza el modelo de **respuesta ranking**, otro tipo de respuestas cerradas en las que el encuestado jerarquiza sus respuestas

ordenándolas como considere oportuno, en este caso, se utilizaba en una escala en la que el 1 otorgaba la máxima importancia y el 5 la mínima.

Solo en el caso de las preguntas doce y dieciocho se usan preguntas con **respuestas mixtas**, las cuales se componen de una parte cerrada y otra abierta, esta última puede ser contestada por el usuario con total libertad utilizando sus propias palabras.

3.3. Recogida de los datos

Como ya se ha comentado anteriormente, la **población** a la que va dirigido el estudio la componen empresas desarrolladoras de software libre que dispongan de alguna sede en territorio español, por ello, para la recogida de datos el primer paso fue la recopilación de las diferentes empresas que encajarían dentro de los potenciales participantes. Se elaboró un listado con empresas que cumplían las siguientes características:

- Al menos una de sus sedes está ubicada en España.
- Parte de su actividad se dedica al desarrollo de software libre.
- La empresa ofrece en su página web un correo electrónico al cual enviar la encuesta. Muchas empresas no han podido ser incluidas en el estudio por carecer de datos de contacto.

Para la elaboración de este listado de empresas se consultaron diferentes páginas web, siendo de gran ayuda las de las siguientes asociaciones de software libre:

- Federación Nacional de Empresas de Software Libre (ASOLIF).
- Asociación de Empresas para el Software Libre (Cat PL).
- Asociación de Empresas Gallegas de Software Libre (AGASOL).
- Empresas de Software Libre de Andalucía (Anda Libre).
- Asociación de Empresas de Software Libre de Euskadi (ESLE).

Una vez investigada la empresa a través de su página web, si cumplía las características indicadas anteriormente se almacenaba de cada una de ellas la siguiente información:

- Nombre de la empresa.
- Página web.
- Correo electrónico.
- Nombre y cargo de la persona a la que se envía la encuesta. Aunque en muchos casos no se disponía de esta información y se envía a una dirección de correo electrónico genérica o de un departamento concreto de la compañía.
- Lugar o lugares en los que estaba ubicada la empresa.

3.4. Envío de la encuesta

Uno de los grandes inconvenientes de la utilización de la encuesta *online* es la baja tasa de respuesta que tiene este recurso, ya que es muy fácil eliminar este tipo de correos o simplemente que el receptor se olvide de su respuesta al dejar su resolución para “más tarde”, razón por la cual el envío de este cuestionario fue realizado en tres ocasiones, para de esta forma ejercer un efecto recordatorio al receptor del correo.

Para que la encuesta no fuese identificada como *spam* o correo basura el envío fue realizado de forma individual, lo cual además permitía personalizar el correo, dirigiéndose directamente al destinatario por su nombre en aquellos casos en los que se disponía de esta información.

Las fechas de envío de la encuesta no han sido elegidas al azar, ya que, aunque la elaboración de las diferentes preguntas que componen el formulario ya había sido realizado con anterioridad, se decidió realizar su envío una vez finalizada la época estival. Además, también se tuvo en cuenta el día de la semana y la hora a la que emitir el mensaje. Se decidió que los envíos se realizarán los miércoles a media mañana. Todas estas medidas lo único que perseguían era aumentar el índice de participación.

Para realizar este envío programado se ha utilizado la opción “Retrasar entrega” que proporciona *Outlook* para que, antes de enviar, se especifique el día y la hora de envío de cada uno de los correos, pudiendo realizar la personalización de cada email poco a poco para finalmente enviarlos todos el día fijado.

3.5. Tasas de respuesta

La realización de este trabajo se inició en el año 2015 pero, por motivos personales, tuvo que ser aplazado hasta el año 2019 donde se ha vuelto a retomar. Los datos obtenidos en el primer envío fueron considerados como obsoletos para el año 2019, por ello, se volvió a repetir en el envío de la encuesta. A continuación, se adjuntan los datos obtenidos en ambos envíos.

Estudio realizado en 2015

Para el estudio realizado en el año 2015 se utilizó una base de datos propia compuesta por **78 empresas** que cumplían las condiciones mencionadas anteriormente.

El primer envío del cuestionario fue realizado el 7 de octubre 2015 obteniendo la respuesta de 11 empresas.

Para el segundo intento se dejó un margen de tiempo de un mes aproximadamente para que los destinatarios no se sintieran presionados, llevando a cabo este envío el 11 de noviembre de 2015. En este caso, participaron en el estudio seis nuevas empresas.

El tercero y último envío fue realizado una semana después, es decir, el 18 de noviembre de 2015, añadiendo al estudio la respuesta de otras cinco compañías.

Por lo que, finalmente, han colaborado **22 empresas**, lo cual supone una tasa de respuesta del **28%**, más elevada de lo que se esperaba obtener antes de comenzar con el proceso de recogida de datos. Se puede concluir, que la estrategia de repetir el envío tres veces ha resultado satisfactoria, sirviendo para incrementar en gran medida el número de empresas participantes.

Estudio realizado en 2019

Para el estudio de 2019, y del cual se analizarán sus resultados en el siguiente capítulo, se decidió ampliar la base de datos, para de esta forma intentar incrementar el número de respuestas, llegando a recopilar los datos de **112 empresas**. Aprovechando la experiencia adquirida con el procedimiento realizado en 2015, se estableció que los envíos de correos se harían a lo largo del mes de noviembre, reforzando el número de respuestas con tres envíos como se había hecho entonces.

El primero de ellos fue realizado el 6 de noviembre de 2019, obteniendo la respuesta de 15 empresas.

Para el segundo envío se dejó un margen de tiempo de dos semanas, efectuando el envío el 20 de noviembre de 2019 y sumando las respuestas de 8 nuevas empresas.

Para finalizar este proceso de recolección de datos, se llevó a cabo el último envío el 27 de noviembre de 2019 añadiendo al estudio 3 nuevas compañías.

Por lo tanto, finalmente se han recolectado datos de **26 empresas**. Por lo tanto, se ha conseguido aproximadamente un **23 % de tasa de respuesta**.

Si se compara la participación, en cuanto a dicha tasa, se puede observar que ha disminuido para el envío realizado en 2019, quizás pueda deberse a que algunas de las empresas que ya habían colaborado en 2015 no han obtenido la suficiente motivación para volver a cubrir la encuesta en 2019 o no han considerado oportuno repetirla porque sus datos no han variado.

Aun así, se debe indicar que, tanto en el año 2015 como en 2019, algunas de las compañías partícipes del estudio no solo han respondido a la encuesta sino que han manifestado su deseo de conocer los futuros resultados una vez finalizado este trabajo y han animado a que continuemos con él, lo cual implica el haber conseguido un buen nivel de implicación e interés de algunos de los participantes.

Además, resaltar que el hecho de realizar el recordatorio de su participación, enviando los correos tres veces, ha sido valorado positivamente por algunos de los receptores, agradeciendo mediante otro *email* el recordatorio y disculpándose por no haber podido dedicarle tiempo desde un inicio.

A todas aquellas empresas que han respondido la encuesta se les ha enviado un correo individual agradeciéndoles su participación y recordándoles la importancia de sus respuestas para lograr el objetivo del estudio, dado que los participantes no han obtenido nada a cambio de su tiempo, al menos hacerles llegar lo importante que es su aportación para este trabajo fin de máster.

3.6. Problemas encontrados

Durante los procesos de envío de la encuesta a las diferentes empresas se han encontrado una serie de problemas o pequeños inconvenientes:

- Algunas páginas web ofrecían direcciones de correo electrónico erróneas o antiguas, lo cual alargó el proceso debido a la necesidad de visitar nuevamente la web en busca de otro *email*.
- Muchas empresas no tenían disponible ninguna dirección de correo electrónico y no se recibió respuesta a la comunicación realizada mediante los formularios de contacto ubicados en la web o incluso directamente desde redes sociales como *Whatsapp*, por lo que no se pudieron incluir dichas empresas en el estudio.
- Algunas de las compañías han respondido indicando que no disponían de tiempo para cubrir la encuesta.
- Las empresas reciben habitualmente demasiados correos electrónicos, en muchos casos de publicidad, por lo que la encuesta pudo pasar desapercibida o incluida dentro de ese tipo de correos.
- Aunque en el cuerpo del email se informaba de la importancia de la participación de las empresas en el estudio, puede que algunos destinatarios se hayan encontrado poco motivados o incentivados para realizar la encuesta, más al tratarse de un correo que recibirán en su dirección corporativa y durante su jornada laboral, por lo que podrían haber dado prioridad a otras actividades propias de su trabajo.

Capítulo 4. Análisis de los datos

En este apartado se analizarán los datos obtenidos de las veintiséis empresas participantes en la encuesta realizada en el año 2019. Para que los resultados sean más legibles y claros se utilizan varios gráficos donde representar mejor dichos resultados y obtener así las conclusiones finales del presente trabajo.

4.1. Clasificación genérica de las empresas

En primer lugar, se detallan los resultados pertenecientes a las preguntas más generales que permiten clasificar las empresas en función de características como el número de empleados o del tipo de software que comercializan.

Como refleja el *Gráfico 1*, las respuestas obtenidas han sido de **empresas pequeñas**, ya que **más del 60% de ellas no superan los 10 empleados** en su plantilla. Sólo el 15,4 % de los encuestados son compañías con 50 o más trabajadores.

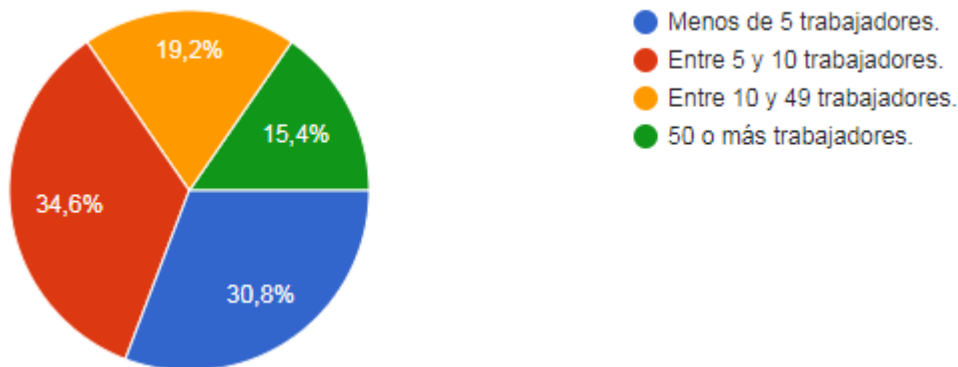


Gráfico 1. Clasificación de las empresas según el número de empleados

En cuanto, al tipo de software comercializado, no se han obtenido sorpresas en los resultados, ya que, mediante una selección previa, se habían enviado los *emails* a compañías que cumplieran el perfil adecuado, por ello, ninguna de ellas se dedica únicamente al desarrollo de software privativo. El **65,4% de los encuestados desarrollan exclusivamente software libre** frente al 34,6% que hacen un desarrollo mixto, a través de software libre y privativo.

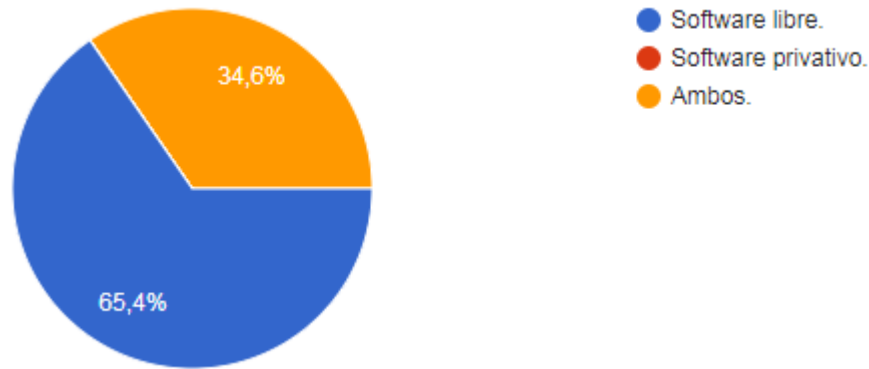


Gráfico 2. Clasificación de las empresas según el software comercializado

Por otro lado, fijándose en **los clientes prioritarios** de estas empresas los resultados reflejan que **la mayor parte de ellas ofrecen sus servicios a la mediana empresa**, aunque también se debe destacar que un 23,1% tienen como principal cliente la administración pública, lo cual indica que poco a poco se intenta integrar este tipo de herramientas para las tareas diarias de la administración, un nicho de mercado importante para estas compañías. A raíz de esto último, ha nacido el movimiento “*Public money, public code*”³, con el propósito de conseguir una legislación que exija que el software financiado con fondos públicos desarrollado para el sector público se ponga a disposición de cualquier usuario bajo una licencia de software libre, afirmando que si es dinero público también debería ser código público. Más de 28.000 personas y 193 organizaciones ya han apoyado este llamamiento mediante la firma de su “carta abierta”, con la finalidad de hacerlas llegar a los representantes europeos correspondientes y conseguir su objetivo.

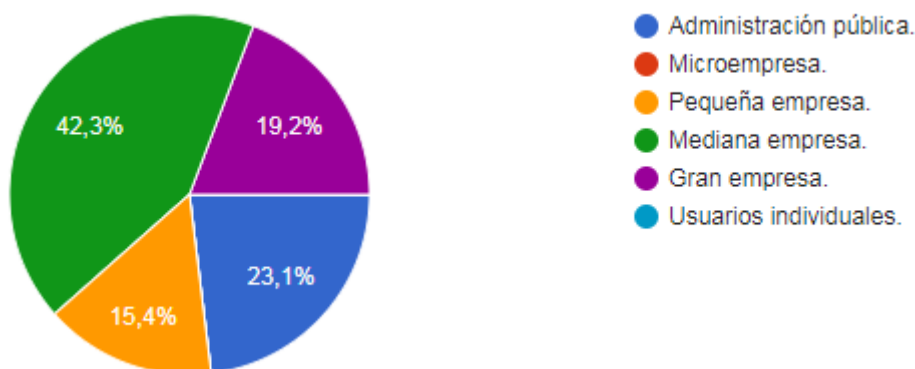


Gráfico 3. Clientes prioritarios de los servicios ofrecidos por la empresa

³ Se puede conocer más en profundidad esta iniciativa en: <https://publiccode.eu/>

4.2. Financiación e ingresos

Dos aspectos importantes planteados a continuación en la encuesta son las fuentes de financiación empleadas por las empresas y cuál es su mayor fuente de ingresos. En cuanto a la financiación, la alternativa más utilizada por las empresas partícipes del estudio ha sido **los recursos generados por la propia empresa**, marcando esta opción catorce de las veintiséis compañías. Esta financiación propia aporta independencia financiera no dependiendo en este campo de agentes externos, y por lo tanto, el coste financiero no existe, aunque sí el coste de oportunidad. Dicha autofinanciación permite que la compañía disfrute de una mayor autonomía y libertad de acción, aunque como inconveniente, las inversiones en estas empresas pueden ser de una cuantía más limitada.

Destacar que el **30,8%** de los encuestados han elegido **las aportaciones de socios** como su fuente de financiación pasando en este caso a fuente de financiación externa. Esta inversión es una clara muestra de compromiso por parte de los propios socios con el logro de los objetivos de la empresa.

A la vista de los resultados obtenidos, se puede afirmar que **este tipo de empresas optan principalmente por un tipo de financiación propia**, bien a través de los recursos generados por la empresa o por aportaciones realizadas por los socios, y descartan, en su mayoría, el tipo de financiación ajena, tanto a corto como a largo plazo.

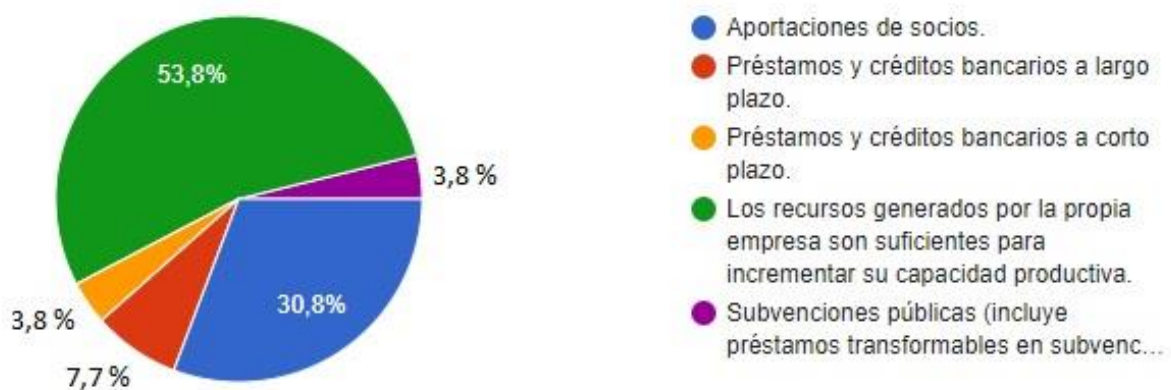


Gráfico 4. Fuentes de financiación más habituales

Un punto importante para toda empresa es identificar cuál va a ser **su mayor fuente de ingresos** y, según los datos obtenidos, dieciocho de las compañías encuestadas se dedican principalmente a **servicios de consultoría y desarrollos personalizados sobre software libre**. Este resultado es el esperado, ya que los desarrollos personalizados son uno de los puntos fuertes de este tipo de software, al ofrecer gran flexibilidad y personalización. En el lado opuesto, se encuentran los servicios de formación o de soporte que son elegidos por una pequeña minoría, dos y uno de los encuestados respectivamente. También se puede observar que el sistema de *partner* aún no está muy desarrollado para este tipo de software, fórmula que sí podemos ver habitualmente en el software privativo.

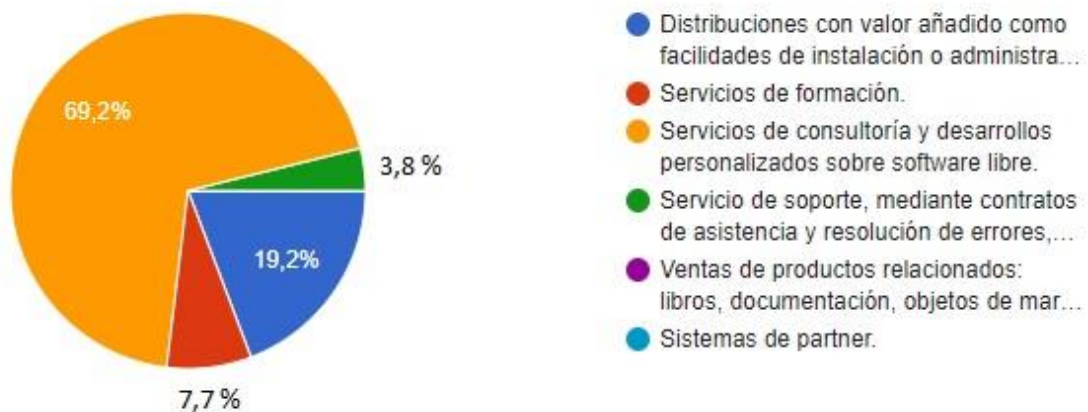


Gráfico 5. Principal método utilizado para generar ingresos

4.3. La estrategia de posicionamiento

Dado que el desarrollo de software realizado por estas empresas suele ser personalizado, cumpliendo los requisitos fijados por el cliente, no es de extrañar que la estrategia de posicionamiento elegida por un **61,5%** de los encuestados, sea la de **especialización**. A ese porcentaje le sigue la estrategia de calidad en el servicio, pero con un índice mucho más bajo, solamente del 23,1% de los encuestados.

4.4. Las funciones dentro de la empresa

En la pregunta siete del cuestionario se planteaban a los participantes una serie de funciones dentro de la empresa para que fuesen ordenadas en función de su importancia, dando la puntuación 1 a la función más importante de la compañía y con un 5 a la menos relevante para ellos. Analizando los datos obtenidos se observa que no hay unanimidad entre los usuarios colaboradores en el estudio, ya que las puntuaciones han sido muy similares para las diferentes funciones como se muestra en el *Gráfico 6*.

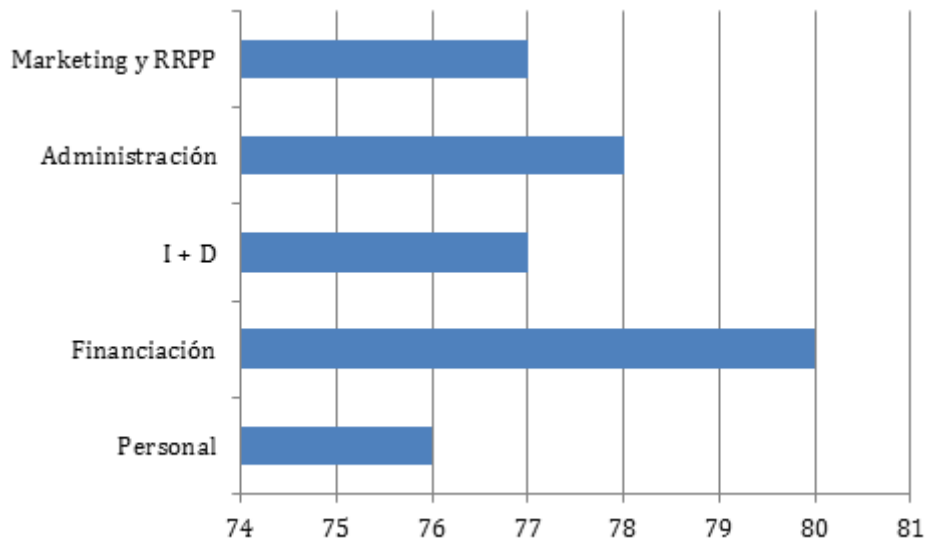


Gráfico 6. Funciones de la empresa

Aun así, los encuestados otorgan **una mayor importancia al personal de la empresa**, ya que, como se ha comentado anteriormente, en las empresas de base tecnológica se necesita personal especializado con gran formación, de ahí que se le dé una mayor consideración. A esta opción le siguen, las funciones de I+D y marketing y relaciones públicas, con la misma puntuación entre ellas; a continuación les seguiría la administración de la compañía; y por último, el aspecto menos valorado ha sido la financiación, quizás porque en un inicio estas empresas no necesitan un gran desembolso para constituirse ni tampoco para sus labores cotidianas, y por ello, las compañías eligen a los empleados como su mayor valor.

4.5. Planificación estratégica

Como se puede observar en el *Gráfico 7*, en el siguiente ítem de la encuesta se ha encontrado un mayor acuerdo entre las empresas. El **76,9%** de los encuestados han indicado que para ellos **la información fluye de abajo hacia arriba**, es decir, las bases de la información descansan en las unidades elementales que constituyen la empresa y a través de un proceso de integración se llega a ofrecer una información general para la totalidad del negocio.

Este modelo de gestión estratégica “abajo hacia arriba” implica involucrar más activamente en el trabajo a los empleados de la empresa, logrando que se esfuercen en mayor medida para alcanzar los objetivos comunes. El equipo directivo o el pequeño empresario, seguirá determinando los objetivos generales de la empresa, pero sus empleados de los diferentes niveles le asistirán en el desarrollo de los mecanismos para alcanzar las metas de la compañía.



Gráfico 7. Cómo fluye la información en la planificación estratégica de la empresa

4.6. Sistema de información

Como se puede apreciar en el *Gráfico 8* hay más disparidad de opiniones en cuanto a cuál es el principal sistema de información utilizado, el 32% de las compañías se decantan por los comités y reuniones, seguidos de quienes emplean en su lugar el presupuesto, con un 28% de los resultados, el resto de opciones les siguen con porcentajes menos significativos



Gráfico 8. Principal sistema de información utilizado por la empresa

4.7. La información contable

En cuanto a la utilidad que las empresas conceden a la información contable, **un 32% manejan esta información para orientar sus decisiones y como ayuda en su verificación.** La segunda opción más elegida, con un 24 % de las respuestas, corresponde a la elección de emplear la información contable como un mecanismo fundamental de la gestión. Solo tres de las compañías encuestadas afirman que no utilizan esta información para la toma de decisiones.

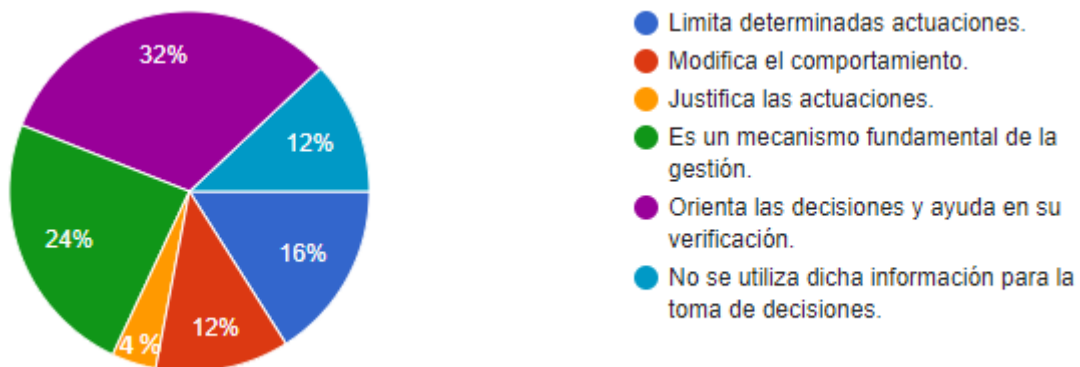


Gráfico 9. La información contable

4.8. Subcontratación de servicios

Las empresas de pequeño tamaño suelen subcontratar ciertos servicios ya que, precisamente debido a su tamaño y volumen de negocio, no precisan disponer de personal dedicado en exclusiva a ciertas tareas, razón por la cual la encuesta planteaba dos cuestiones al respecto: si se subcontrataba la contabilidad, algo que suele ser habitual en este tipo de compañías, y si se subcontrataban más servicios y cuáles eran.

Analizando los resultados, se obtuvo que **el 60% de las empresas encuestadas sí disponían de una contabilidad externalizada**, y además, casi el mismo porcentaje, **el 58,3 %, subcontratan otros servicios**. Varios han coincidido en la subcontratación de las nóminas, recursos humanos y prevención de riesgos laborales. Otros han incluido servicios como: comunicación, diseño gráfico, calidad o asesoramiento legal, fiscal y jurídico.

4.9. Contabilidad de costes

La contabilidad de costes consiste en realizar una imputación razonable de costes directos e indirectos que permita obtener información analítica en la que apoyarse en la toma de decisiones de la dirección de la empresa. Las empresas a través de esta contabilidad de costes realizan un minucioso análisis sobre los costes de producción, distribución, financiación y administración para posteriormente procesar toda esa información.

En este caso, para la contabilidad de costes, los resultados obtenidos son que **veintiuna de las veintiséis empresas participantes, lo que representa el 80,8%, han manifestado que su empresa se apoya en sistemas de información contable interna plasmados en documentos e informes, para la toma de decisiones**. Sólo cinco de las empresas basan las acciones futuras en la experiencia de su gerente.

Si se fija la atención en cuál es el **modelo de estimación de costes** utilizado, se encuentra un menor acuerdo en los resultados, ya que, aunque **el 64% de las empresas utilizan la experiencia y la madurez en proyectos similares**, para el resto de respuestas no se muestra una alternativa clara, sino que se dividen entre el resto de opciones propuestas.

4.10. Reutilización de código

La encuesta dispone de tres ítems dedicados a la reutilización de código, una de las grandes fortalezas del software libre, que permite una mayor productividad y un ahorro en costes. En el estudio se pretende consultar los porcentajes de código libre y código propio (desarrollado por la empresa) reutilizado en sus proyectos, con cuatro posibles respuestas:

- a) Inferior al 20%
- b) Entre el 20% y el 50%
- c) Superior al 50%
- d) No sabe / no contesta

Los resultados obtenidos se pueden observar en los siguientes diagramas de sectores:

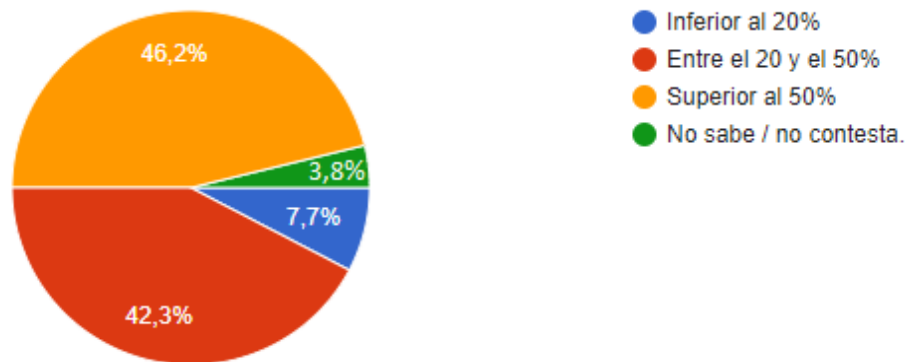


Gráfico 10. Porcentajes de código libre reutilizado por la empresa

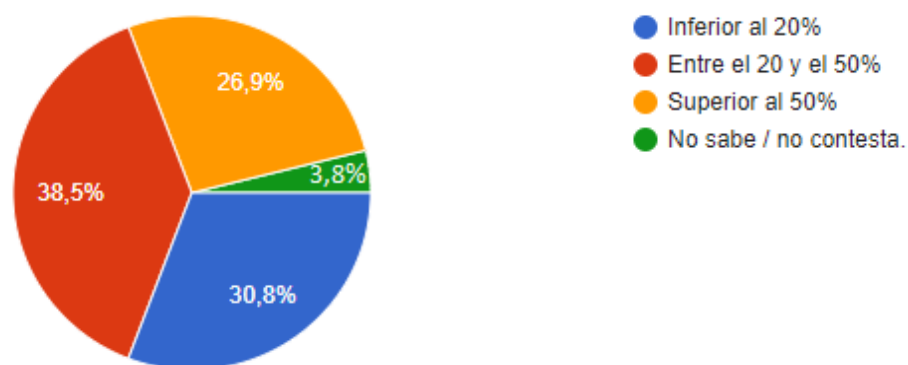


Gráfico 11. Porcentajes de código propio reutilizado por la empresa

Si se observan los resultados, se puede concluir que, en el caso del software libre las empresas en su mayoría se encuentran divididas entre dos opciones, o bien reutilizan entre el 20% y el 50% o más del 50% del código.

Sin embargo, en el caso del software desarrollado por la propia empresa hay una mayor división de opiniones entre las tres respuestas ofrecidas a los usuarios, por lo que no se puede precisar un rango determinado.

Relacionado a la reutilización de código la encuesta plantea un último ítem en el cuál se plantean algunas afirmaciones sobre esta reutilización del código, para que los participantes elijan las que consideren más adecuadas sin limitación de número. Entre todas ellas, en la que se ha encontrado un mayor acuerdo es en que la reutilización de código *“Incrementa la productividad”*, seguida de que *“Aumenta la calidad del software”*, *“Reduce la tasa de fracaso en los proyectos”*, *“Mejora los costes de mantenimiento”* y proporciona *“Código más seguro”*.

4.11. Donación al software libre

Por último, para finalizar la encuesta se consultaba si las empresas donan parte de sus beneficios anuales a alguna fundación, asociación, comunidad o proyecto para contribuir al mantenimiento de la comunidad de software libre, obteniendo que, poco más de la mitad, **14 de los 26 (el 53,8%) encuestados han respondido afirmativamente.**

En cuanto al porcentaje destinado a tal fin, se han recibido diversidad de opiniones: algunos no destinan un porcentaje de sus beneficios sino un valor fijo en donaciones puntuales, otra compañía ha respondido que no dona dinero sino horas de trabajo como colaboración y en algunos casos no han querido ofrecer este dato. Realizando la media de los porcentajes conseguidos se ha obtenido que **las compañías dedican de media un 5,66% de sus beneficios anuales al mantenimiento y crecimiento del software libre.**

Capítulo 5. Conclusiones

Este apartado final dedicado a las conclusiones obtenidas tras la realización de este Trabajo Fin de Máster, se dividirá a su vez en dos pequeñas partes, por un lado, las conclusiones extraídas tras evaluar los resultados obtenidos de la encuesta *online* y por otro, aquellas de carácter más general que se han alcanzado tras la lectura y análisis de la información obtenida de las diferentes fuentes.

En primer lugar, realizando, un análisis final de los datos obtenidos a través de la encuesta enviada por correo electrónico se pueden sacar varias conclusiones que se expondrán a continuación, sin olvidar que el perfil mayoritario de empresas, que de manera desinteresada han decidido participar en el estudio, son empresas pequeñas de menos de 10 empleados.

Las empresas con este perfil se han decantado por ofrecer servicios de consultoría y desarrollos personalizados para sus clientes como modelo de negocio, trabajando en gran parte con la mediana empresa y la administración. Para conseguir una mayor competitividad, en tiempo y en costes, hacen uso de la reutilización de código, tanto en el desarrollado por la propia compañía como el ofrecido por la comunidad de software libre. Además, gracias a esta reutilización los participantes han puesto de manifiesto que consiguen incrementar la productividad, aumentar la calidad del software y reducir la tasa de fracaso en sus proyectos. Aunque con esta última medida de reutilización explotan una de las bases del movimiento de software libre, con su filosofía de compartir y reutilizar contenidos, se puede observar que estas compañías desaprovechan otras oportunidades, como es la de ofrecer servicios de formación o soporte, y así además contribuir a disminuir una de las debilidades de este tipo de software, la carencia de estos servicios, y que provoca gran parte del rechazo que genera por algunos usuarios. Tampoco han sabido desarrollar en exceso la figura del *partner*, como sí ocurre en el software privativo.

Debido a su naturaleza, estas empresas recurren a la financiación propia, bien a través de los recursos generados por la propia empresa o la aportación de sus socios, lo cual puede frenar la expansión y crecimiento de la compañía. El gran activo de estas empresas, como ya se ha mencionado a lo largo de este documento, es su personal al que a la vista de los resultados otorgan gran importancia, tanta que delegan en ellos su modelo de estimación de costes, basándolos en la experiencia y madurez en proyectos similares, desperdiciando la

oportunidad de utilizar mejores modelos de estimación según las características del proyecto y dejando demasiado peso en dicha estimación a sus desarrolladores o expertos.

Para finalizar con esta conclusión sobre la recogida de datos, señalar que, aunque estas empresas realicen donaciones de código al establecer las licencias libres, han declarado una donación monetaria bastante pequeña para el mantenimiento y crecimiento del movimiento del software libre, aunque por otro lado, ha puesto de manifiesto algo que ya se ha mencionado con anterioridad, y es que, por sorprendente que parezca, las mayores aportaciones proceden de las grandes empresas, algo muy a destacar dentro de este fenómeno. Como aspecto positivo se debe resaltar la buena acogida que ha tenido el estudio por parte de las empresas encuestadas, ya que, muchos de los que han decidido participar, no sólo se han limitado a responder al cuestionario que se les planteaban, sino que también enviaban ánimos para continuar con iniciativas como esta o se interesaban por conocer los resultados finales una vez concluido el trabajo. Por ello, aunque al inicio tenía alguna duda sobre si la encuesta sería la mejor opción para la parte más práctica de este trabajo, finalmente su uso ha sido muy satisfactorio, obteniendo una visión de cuál es la situación actual en este tipo de compañías y observando cómo se cumplían muchos de los aspectos analizados en la parte teórica.

Cerrando el documento con las conclusiones o reflexiones más generales, quiero destacar que el software libre se ha convertido en una parte relevante de la industria del software, convirtiéndose en una alternativa al software comercial en diversas áreas, como: la educación, la sanidad o la economía. La relevancia de este software ha ido en aumento progresivamente, incrementando su uso entre diferentes perfiles de usuarios y de compañías, desde la empresa privada hasta la administración pública han apostado por la implantación de este tipo de software en sus sistemas.

El gran paso para el inicio de este fenómeno fue el momento en el que un grupo de personas fue capaz de canalizar toda su capacidad productiva y movilizar a un número elevado de individuos con un fin único, **liberar todo su *know-how* a la comunidad**, estableciéndose lo que podría concebirse como la semilla de este movimiento. Esta filosofía de trabajo no solo perdura con el paso del tiempo, sino que además sigue creciendo y evolucionando, creando potentes lazos de colaboración entre sus miembros, y convirtiendo dicha **colaboración social en un impulsor clave de la productividad**.

El éxito del sistema operativo Linux ha demostrado la viabilidad del software de código abierto, una forma alternativa de desarrollo de software que desafía los supuestos tradicionales sobre los mercados de software. Estas comunidades emergen rápidamente y tienen éxito, sin los mecanismos de informes y organigramas visibles dentro de las empresas (Fleming y Waguespack, 2007).

Pero no sólo se debe pensar en Linux como ejemplo de éxito del software libre, a lo largo del presente documento se han citado diferentes casos que reflejan el éxito de este movimiento, ejemplos de ello son nombres tan conocidos como: Firefox, uno de los navegadores web más populares; Android, sistema operativo líder para dispositivos móviles; Open Office, paquete que ofrece diversas herramientas, como procesador de texto u hojas de cálculo; Moodle, plataforma orientada a la enseñanza para educadores y estudiantes o WordPress, herramienta muy utilizada para la gestión de webs. Sin olvidar que, las empresas que ofertan servicios en la red, como son Google, Facebook o Amazon depende en gran medida del software libre, sin él sus fundadores no habrían podido establecerlas ni escalarlas a sus tamaños actuales, consiguiéndolo gracias al acceso al código fuente y su modificación, principios básicos de este tipo de software.

Además, comprender por qué los desarrolladores participan en proyectos de código abierto es crucial para evaluar el impacto de este software. Sus motivaciones principales quedan divididas en dos grandes categorías: factores internos, como pueden ser el disfrute, la creatividad o el deseo de resolver un problema y recompensas externas, tales como los rendimientos futuros esperados, transformados por ejemplo en una buena oportunidad de trabajo. También la propia pertenencia a una comunidad impulsa a gran parte de los colaboradores del movimiento libre, personas distribuidas en todo el mundo son capaces de trabajar con otras con las que tienen afinidades e intereses compartidos. El software libre permite a sus colaboradores intercambiar esfuerzos en lugar de productos, bajo un régimen en el que los proveedores de código se autoidentifican como proveedores de productos en un mercado, en lugar de empleados en una empresa (Langlois y Garzarelli, 2008).

Aunque cualquier individuo anónimo puede donar, tanto su propio código fuente, como realizar una aportación monetaria al movimiento iniciado por Stallman, ha quedado de manifiesto que son las grandes empresas las que financian la evolución y crecimiento de este software, un aspecto que hasta realizar el proceso de investigación para este trabajo desconocía y que ha resultado todo un descubrimiento, comprobando como empresas de

las dimensiones de Microsoft o Google colaboran tan activamente en este proyecto. Una de las últimas iniciativas en las que se puede observar el fuerte compromiso de estas empresas con el funcionamiento del software libre es el anuncio de la Linux Foundation en el que informaba que, debido a las circunstancias vividas en el año 2020 por la pandemia del Covid-19, Intel realizaría una inversión adicional en el *Mentorship Program* de la fundación, con una contribución de 250.000 dólares para fomentar el crecimiento de este programa. Según palabras de la vicepresidenta de Intel, Melissa Evers-Hood, “*durante estos tiempos difíciles, nuestra capacidad de unirnos para ayudar a cultivar la próxima generación de desarrolladores de software es más importante que nunca*”. Este programa está diseñado para permitir que los desarrolladores experimenten, aprendan y contribuyan a las comunidades de código abierto al tiempo que fortalecen los proyectos de código abierto y crean un grupo de desarrolladores cada vez más capacitados y heterogéneos. Su compromiso de protección a la comunidad, adaptándose a las necesidades cambiantes hace que el futuro del software libre sea cada vez más prometedor.

Ya como última reflexión, indicar que tras el análisis realizado en este Trabajo Fin de Máster se ha podido comprobar como la colaboración y el software libre pueden ser beneficiosos tanto para las comunidades como para empresas. Las grandes compañías han sabido apoyarse en los puntos fuertes que este software les ofrece, como su robustez y seguridad, aportando a esta simbiosis otros aspectos como su imagen de marca y una serie de servicios relacionados cosechando así las claves del éxito sin perder la esencia del movimiento del software libre.

Capítulo 6. Bibliografía

- ALONSO MAGDALENO M.I. y GARCÍA GARCÍA J. 2015. *Sustainability and social responsibility reporting in open source software*. International Journal of the Commons, Nº 9, pp. 369-397. <http://doi.org/10.18352/ijc.503>
- ASTIGARRAGA E. 2011. *El método Delphi*. Universidad de Deusto. https://www.academia.edu/1778724/El_M%C3%A9todo_Delphi._Universidad_de_Deusto
- BUENO E. y MERINO C. 2007. *El capital intelectual y la creación de empresas en la sociedad del conocimiento*. Encuentros Multidisciplinares, Vol. 9, Nº 26, pp. 37-46. ISSN 1139 9325. https://www.researchgate.net/publication/28237773_El_capital_intelectual_y_la_creacion_de_empresas_en_la_sociedad_del_conocimiento
- CENATIC. 2011. *El software libre en los organismos públicos de ámbito estatal*. https://administracionelectronica.gob.es/pae_Home/dam/jcr:d04fda8f-8457-413e-9c6e-002307c29b26/Software_Libre_en_organismos_publicos_ambito_estatal_2011.pdf
- CHIAVENATO A. 2012. *Introducción a la teoría general de la administración*. McGraw-Hill. ISBN 9789701055007.
- DAVENPORT T. y KLAHR P. 1998. *Managing customer support knowledge*. California Management Review, Vol. 40, Nº 3, pp. 195-208. <https://doi.org/10.2307/41165950>
- DRUCKER P. y ORBEGOZO G. 2003. *Harvard Business Review, Gestión del Conocimiento*. Ediciones Deusto. ISBN 9788423420230.
- ESCOBAR J. y CUERVO A. 2008. *Validez de contenido y juicio de expertos. Una aproximación a su utilización*. Avances en Medición, Vol. 6, Nº 1, pp. 27-39. ISSN 1692-0023. https://www.researchgate.net/publication/302438451_Validez_de_contenido_y_juicio_de_expertos_Una_aproximacion_a_su_utilizacion
- EUROPEAN COMMISSION. 2017. *The Economic and Social Impact of Software and Services on Competitiveness and Innovation*. <https://ec.europa.eu/digital-single-market/en/news/economic-and-social-impact-software-and-services-competitiveness-and-innovation>

- FARIÑAS J. C. y LÓPEZ A. 2007. *Las empresas pequeñas de base tecnológica en España: delimitación, evolución y características*. Economía Industrial, Nº 363, pp. 149-160. ISSN 0422-2784. <http://www.ipyme.org/publicaciones/informeebt.pdf>
- GARCÍA FUENTES R. 2014. *Software libre: coste, valor y estrategia*. Trabajo Fin de Máster, Universidad de Oviedo. http://digibuo.uniovi.es/dspace/bitstream/10651/34626/3/TFM_RebecaGarciaFuentes.pdf
- GARCÍA GARCÍA J. y ALONSO MAGDALENO M.I. 2012. *La contribución empresarial al software libre como actividad socialmente responsable*. Teknokultura. Revista de Cultura Digital y Movimientos Sociales, Nº 10, pp. 221-230. <https://revistas.ucm.es/index.php/TEKN/article/download/48062/44939>
- GARRIGA A. 2018. *Guía práctica en gestión de proyectos*. Ediciones Tagus. ISBN 9788409057849. <https://es.scribd.com/book/392553309/Guia-practica-en-gestion-de-proyectos-plantillas-editables>
- HEEMSTRA F.J. 2003. *Software Cost Estimation*. Information and Software Technology, Vol. 34, Nº 10, pp. 627-639. [https://doi.org/10.1016/0950-5849\(92\)90068-Z](https://doi.org/10.1016/0950-5849(92)90068-Z)
- INSTITUTO NACIONAL DE ESTADÍSTICA. 2019. *Encuesta sobre el uso de TIC y comercio electrónico en las empresas*. https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=estadistica_C&cid=1254736176743&menu=ultiDatos&idp=1254735576799
- INSTITUTO NACIONAL DE ESTADÍSTICA. 2018. *Encuesta sobre innovación en las empresas*. https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=Estadistica_C&cid=1254736176755&menu=ultiDatos&idp=1254735576669
- INSTITUTO NACIONAL DE ESTADÍSTICA. 2018. *Encuesta de uso de TIC y Comercio Electrónico en las empresas*. <https://www.ine.es/jaxi/Datos.htm?path=/t09/e02/a2017-2018/10/&file=04001.px#!tabs-tabla>
- LANDETA J. 1999. *El método Delphi una técnica para de predicción para la incertidumbre*. Ariel. ISBN 9788434428362.

- LEE D. y KIM B.C. 2013. *Motivations for open source project participation and decisions of software developers*. Computational Economics, N° 41, pp. 31-57.
<https://doi.org/10.1007/s10614-011-9311-x>
- LERNER J. y TIROLE J. 2003. *Some simple economics of open Source*. The Journal of Industrial Economics, Vol. 50, N° 2, pp. 197-234.
<http://dx.doi.org/10.1111/1467-6451.00174>
- LESSIG L. 2005. *Por una cultura libre. Cómo los grandes grupos de comunicación utilizan la tecnología y la ley para clausurar la cultura y controlar la creatividad*. Traficantes de sueños. ISBN 9788496453029.
- LEUNG H. y FAN Z. 2001. *Software Cost Estimation*.
https://doi.org/10.1142/9789812389701_0014
- MENDONZA F.J.E. 2009. *Metodología para el cálculo de costo total de propiedad para sistemas informáticos*. Trabajo de Graduación, Universidad de San Carlos de Guatemala. http://biblioteca.usac.edu.gt/tesis/08/08_0444_CS.pdf
- MERINO C. 2010. *Dirección estratégica del conocimiento en el contexto de las nuevas empresas de base tecnológica*. Encuentros Multidisciplinares, Vol. 12, N° 36. ISSN: 1139 9325. <https://dialnet.unirioja.es/servlet/articulo?codigo=3331085>
- PATEL J. 2013. *Software Engineering & Object Oriented Modeling*. EBookIt.
- PEREIRA H. 2011. *Implementación de la gestión del conocimiento en la empresa*. Publicaciones CEGESTI.
https://www.cegesti.org/exitoempresarial/publicaciones/publicacion_135_310111_es.pdf
- RAMOS D. 2014. *Desarrollo de software: requisitos, estimaciones y análisis*. Createspace Independent Pub. ISBN 9781503259485.
- RAMOS D., NORIEGA R., LAÍNEZ J. R. y DURANGO A. 2015. *Curso de ingeniería de software*. IT Campus Academy. ISBN 9781544132532.
- RONKEN H. O. y LAWRENCE P. R. 1952. *Administering changes: a case study of human relations in a factory*. Boston, Division of Research, Harvard Business School, Vol. 288, N° 1, pp.180-181.
<https://doi.org/10.1177/000271625328800150>

- RUSSO B., BRAGHIN C., GASPERI P., SILLITTI A. y SUCCI G. 2005. *Defining the Total Cost of Ownership for the Transition to Open Source Systems*. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.205.5679>
- SAFFIRIO M. 2016. *Costo Total de la Propiedad (TCO) y la Administración del Ciclo de Vida (LCM)*. <https://msaffirio.wordpress.com/2006/04/08/costo-total-de-propiedad-tco-y-administracion-del-ciclo-de-vida-lcm>. Consultado el 20/06/2020.
- SARVARY M. 1999. *Knowledge management and competition in the consulting industry*. California Management Review, Vol. 41, N° 2, pp. 95-107. <https://doi.org/10.2307/41165988>
- SEOANE J., GONZÁLEZ J.M. y ROBLES G. 2007. *Introducción al software libre*. Universidad de Zaragoza. <http://softlibre.unizar.es/manuales/softwarelibre/sobre.pdf>
- SHAIKH M. y CORNFORD T. 2011. *Framing the Conundrum of Total Cost of Ownership of Open Source Software*. 9th Open Source Systems: Grounding Research, Octubre 2011, Salvador, Brazil, pp. 208-219. ISSN 1868-4238. https://link.springer.com/chapter/10.1007/978-3-642-24418-6_14
- SOMALO N. 2017. *Marketing digital que funciona: Planifica tu estrategia e invierte con cabeza*. LID Editorial. ISBN 9788416624751.
- SOMMERVILLE I. 2011. *Ingeniería del software*. Addison-Wesley. ISBN 9786073206037.
- STALLMAN R. 2004. *Software libre para una sociedad libre*. Traficantes de sueños. ISBN 9788493355517.
- VEN K., VERELST J. y MANNAERT H. 2008. *Should You Adopt Open Source Software?* IEEE Software, Vol. 25, N° 3, pp. 54-59. <https://ieeexplore.ieee.org/document/4497765>
- VENTURA VICTORIA J. 2008. *Análisis estratégico de la empresa*. Paraninfo. ISBN 9788497323024.
- WOUTERS M., ANDERSON J.C. y WYNSTRA F. 2005. *The adoption of total cost of ownership for sourcing decisions, a structural equations analysis*. Accounting, Organizations and Society, Vol. 30, N° 2, pp. 167-191. <https://doi.org/10.1016/j.aos.2004.03.002>

Páginas web:

- Asociación Asturiana de Startups. <http://www.asturvalley.com>. Consultada el 26/09/2019.
- Asociación Española de Business Angels. <https://www.aeban.es>. Consultada el 30/10/2019.
- Centro de apoyo a la PYME en materia de gestión de propiedad industrial, intelectual e innovación. <http://www.cevipyme.es>. Consultada el 26/09/2019.
- Centro Europeo de Empresas e Innovación. <http://www.ceei.es>. Consultada el 26/09/2019.
- Creative Commons. <https://creativecommons.org>. Consultada el 19/02/2020.
- Debian. <https://www.debian.org>. Consultada el 27/02/2020.
- Emprende Pyme. <https://www.emprendepyme.net>. Consultada el 30/10/2019.
- Escuela de Negocios de la Innovación y los Emprendedores. <https://www.iebschool.com>. Consultada el 20/02/2020.
- Escuela de Organización Industrial. <https://www.eoi.es>. Consultada el 18/02/2020.
- Free Software Foundation. <https://www.fsf.org>. Consultada el 27/09/2020.
- Fullstep. <https://www.fullstep.com>. Consultada el 27/05/2020.
- Gartner. <https://www.gartner.com>. Consultada el 27/05/2020.
- Instituto Nacional de Estadística. www.ine.es. Consultada el 30/11/2019.
- International Data Corporation (IDC). <https://idcspain.com>. Consultada el 30/05/2020.
- Open SUSE. www.opensuse.org. Consultada el 30/05/2020.
- Proyecto GNU. <https://gnu.org>. Consultada el 26/09/2019.
- Pymes y autónomos. <https://www.pymesyautonomos.com>. Consultada el 19/01/2020.

- Red Hat. <http://www.redhat.com>. Consultada el 28/05/2020.
- Repositorio de la Universidad Complutense de Madrid. <https://repositorio.uam.es>. Consultada el 31/11/2019.
- SUSE. www.suse.com. Consultada el 29/05/2020.
- The Apache Software Foundation. <http://www.apache.org>. Consultada el 24/02/2020.
- The Linux Foundation. www.linuxfoundation.org. Consultada el 17/11/2019.
- TIC Portal. <https://www.ticportal.es>. Consultada el 27/05/2020.
- Universo Crowdfunding. <https://www.universocrowdfunding.com>. Consultada el 30/10/2019.

Anexo I: Encuesta

A continuación, se muestra la encuesta que las empresas han recibido por correo electrónico:

1. *Clasificación de la empresa según el número de empleados:*
 - a. Menos de 5 trabajadores.
 - b. Entre 5 y 10 trabajadores.
 - c. Entre 10 y 49 trabajadores.
 - d. 50 o más trabajadores.

2. *Tipo de software comercializado:*
 - a. Software libre.
 - b. Software privativo.
 - c. Ambos.

3. *Clientes prioritarios de los servicios ofrecidos:*
 - a. Administración pública.
 - b. Microempresa.
 - c. Pequeña empresa.
 - d. Mediana empresa.
 - e. Gran empresa.
 - f. Usuarios individuales.

4. *Las fuentes de financiación utilizadas habitualmente son:*
 - a. Aportaciones de socios.
 - b. Préstamos y créditos bancarios a largo plazo.
 - c. Préstamos y créditos bancarios a corto plazo.
 - d. Los recursos generados por la propia empresa son suficientes para incrementar su capacidad productiva.
 - e. Subvenciones públicas (incluye préstamos transformables en subvenciones).

5. *El principal método utilizado por la empresa para generar ingresos es ofrecer al cliente:*
- Distribuciones con valor añadido como facilidades de instalación o administración.
 - Servicios de formación.
 - Servicios de consultoría y desarrollos personalizados sobre software libre.
 - Servicio de soporte, mediante contratos de asistencia y resolución de errores, o soportes telefónicos y/u *online*.
 - Ventas de productos relacionados: libros, documentación, objetos de marca,...
 - Sistemas de *partner*.
6. *La estrategia de posicionamiento elegida es:*
- Liderazgo en costes.
 - Especialización.
 - Ventaja relacionada con el trato del personal de la empresa al cliente.
 - Ventaja asociada a la cualificación del personal.
 - Calidad del servicio.
7. *Ordene las siguientes funciones, teniendo en cuenta su importancia dentro de la empresa (escala del 1(más importante) al 5 (menos importante)):*
- Personal.
 - Financiación.
 - Investigación y desarrollo (I+D).
 - Administración.
 - Marketing y relaciones públicas (con clientes y comunidades de desarrolladores).
8. *La información dentro de su planificación estratégica fluye de:*
- “Arriba hacia abajo”, es decir, la elaboración parte de la máxima jerarquía de la empresa y llega hasta las unidades básicas de la misma.
 - “Abajo hacia arriba”, las bases de la información descansan sobre las unidades elementales que constituyen la empresa y a través de un proceso

de integración se llega a ofrecer una información general para la totalidad del negocio.

9. *El principal sistema de información utilizado es:*

- a. El presupuesto.
- b. Los comités y reuniones.
- c. El sistema contable.
- d. Las conversaciones telefónicas, informales, correos electrónicos, comidas de negocios.
- e. Los análisis de desviaciones, comparando los tiempos y costes previstos con el resultado final.
- f. Las medidas de logros y objetivos conseguimos finalmente.
- g. Los informes elaborados dirigidos a la dirección de la empresa.

10. *En su empresa la información contable...*

- a. Limita determinadas actuaciones.
- b. Modifica el comportamiento.
- c. Justifica las actuaciones.
- d. Es un mecanismo fundamental de la gestión.
- e. Orienta las decisiones y ayuda en su verificación.
- f. No se utiliza dicha información para la toma de decisiones.

11. *La empresa subcontrata la contabilidad: (en caso afirmativo responder la pregunta 12, sino saltar a la 13).*

- a. Sí.
- b. No.

12. *¿Subcontrata algún servicio más que la contabilidad? En caso afirmativo indicar cuál o cuáles.*

- a. Sí.
- b. No.

Servicios subcontratados:

13. En cuanto a la contabilidad de costes...

- a. La empresa se apoya en sistemas de información contable interna, plasmados en documentos e informes, para la toma de decisiones.
- b. Las acciones futuras se basan en la intuición y experiencia del gerente que es quien marca las pautas a seguir.

14. El modelo de estimación de costes utilizado es:

- a. Técnica Delphi.
- b. Modelo con base estadística.
- c. La experiencia y madurez en proyectos similares.
- d. Modelo de estimación orientado a las líneas de código.
- e. Modelos compuestos (por ejemplo: COCOMO, puntos función).
- f. Otros. Indicar cuál o cuáles:

15. Estimación del porcentaje de código libre reutilizado en los proyectos:

- a. Inferior al 20%
- b. Entre el 20 y el 50%
- c. Superior al 50%
- d. No sabe / no contesta.

16. Estimación del porcentaje de código propio (desarrollado por la empresa) reutilizado en los proyectos:

- a. Inferior al 20%
- b. Entre el 20 y el 50%
- c. Superior al 50%
- d. No sabe / no contesta.

17. A continuación, se plantean una serie de afirmaciones sobre la reutilización de código, señale las considere acertadas, puede marcar más de una opción:

- a. Reduce la tasa de fracaso en los proyectos.
- b. Mejora los costes de mantenimiento.
- c. Incrementa la productividad.
- d. Aumento de la calidad del software.

- e. Se obtienen mejoras en las actividades de control y planificación por la reducción de desviaciones en los desarrollos.
- f. Código más seguro.
- g. La reutilización de código supone un esfuerzo extra respecto al desarrollo desde cero.
- h. Se necesita personal más cualificado.
- i. Ninguna de las anteriores.

18. Para contribuir a la construcción y el mantenimiento de la comunidad de software libre la empresa dona parte de sus beneficios anuales a alguna fundación, asociación, comunidad o proyecto. En caso afirmativo indicar el porcentaje destinado a tal fin.

- a. Sí.
- b. No.

Porcentaje destinado.

Anexo II: Listado de empresas participantes en el estudio

A continuación, se enumeran las empresas que han participado en el estudio respondiendo perfectamente a la totalidad de la encuesta:

En el estudio realizado en 2015:

- Abertal Networks. *www.abertal.com*
- Aner Sistemas Informáticos SL. *www.aner.com*
- Anubía soluciones en la nube, S.L. *www.anubia.es*
- Clarcat Consulting. *www.clarcat.com*
- CodeSyntax. *www.codesyntax.com*
- Dabne Tecnologías de la Información. *www.dabne.net*
- Dimensiona. *www.dimensiona.com*
- Earcon SL. *www.earcon.com*
- Edooca. *www.edooca.com*
- Enreda S. Coop. And. *www.enreda.coop*
- Entelgy-Ibai, S.A.U. *www.entelgy.com*
- Gestiwweb Integración de Soluciones Web S.L. *www.gestiwweb.com*
- Icarto S.L. *www.icarto.es*
- Igalia. *www.igalia.com*
- Innova4B. *www.innova4b.com*
- Irontec. *www.irontec.com*
- Kernet S.L. *www.kernet.es*
- Librebit. *www.librebit.com*
- Open Systems Consulting. *www.opscons.com*
- Open Trends Solucion Sistemas S.L. *www.opentrends.net*
- Queres Tecnologías. *www.quires.es*
- Tagen Ata Lingua e Comunicación S. Coop. Galega. *www.tagenata.com*

En el estudio realizado en 2019:

- Anubía soluciones en la nube, S.L. www.anubia.es
- Arcadia Consulting. www.arcadiaconsulting.es
- Ártica ST. www.artica.es
- Asociación de Empresas Gallegas de Software Libre (Agasol). www.agasol.gal
- Avanzosc, S.L. www.avanzosc.es
- BarnetikKoop. www.barnetik.com
- Bitergia. www.bitergia.com
- Btactic, SCCL. www.btactic.com
- Comusys. www.comusys.com
- Corunet. www.coru.net
- Deindo Ideas. www.deindo.es
- Gnuo Consultores. www.gnuo-consultores.com
- Grupo Oesia. www.grupooesia.com
- Icarto. www.icarto.es
- Igalia. www.igalia.com
- Innova4B. www.innova4b.com
- Irontec. www.irontec.com
- Queres Tecnologías. www.ques.es
- Librebit. www.librebit.com
- Libreinnova Digital, SLU. www.libreinnova.com
- Openinnova. www.openinnova.es
- Tagen Ata. www.tagenata.com
- TICgal. www.tic.gal/es
- 2Byte Soluciones informáticas S.L. www.2byte.es
- 3&Punt Solucions Informàtiques S.L. www.tresipunt.com
- 3.14 Financial Contents. www.3.14financialcontents.com

Anexo III: Glosario

BI: corresponden a las siglas de *Business Intelligence* definiéndose en español como Inteligencia de Negocio o Inteligencia Corporativa. Es un término genérico que incluye las aplicaciones, la infraestructura y las herramientas, y las mejores prácticas que permiten el acceso y el análisis de la información para mejorar y optimizar las decisiones y rendimiento.

Código binario: sistema de numeración en el que los números se representan utilizando solamente dos cifras: cero y uno. Este sistema de codificación se utiliza, entre otras muchas cosas, para representar textos o procesar instrucciones.

Código fuente: texto que escribe un programador utilizando sentencias válidas del lenguaje con que esté trabajando.

Correo basura o spam: correo electrónico no solicitado, habitualmente de tipo publicitario, enviados en grandes cantidades (incluso masivas) que perjudican de alguna o varias maneras al receptor.

Cost drivers: es el factor que crea el coste de una actividad, la causa del mismo.

CustomerRelationship Management (CRM): aplicación que permite centralizar en una única base de datos todas las interacciones entre una empresa y sus clientes.

Distribución de software: conjunto de software específico ya compilado y configurado.

Downtime: período de tiempo en el que una máquina no está disponible para su utilización debido a un fallo o incidencia.

Enterprise ResourcePlanning (ERP): programas que se hacen cargo de distintas operaciones internas de una empresa, desde producción a distribución o incluso recursos humanos.

Escalabilidad: es la propiedad deseable de un sistema, una red o un proceso, que indica su habilidad para reaccionar y adaptarse sin perder calidad, o bien manejar el crecimiento continuo de trabajo de manera fluida, o bien para estar preparado para hacerse más grande sin perder calidad en los servicios ofrecidos.

Hacker: persona con grandes habilidades en el manejo de computadoras que investiga un sistema informático para avisar de los fallos y desarrollar técnicas de mejora. También, habitualmente en medios de comunicación, persona que accede ilegalmente a sistemas informáticos ajenos para apropiárselos u obtener información secreta, aunque su denominación correcta en este caso ha de ser *cracker*.

Hardware: equipo, conjunto de aparatos de una computadora.

Internet: red informática mundial, descentralizada, formada por la conexión directa entre computadoras mediante un protocolo especial de comunicación.

KLOG: es el equivalente a mil líneas de código efectivo, es decir, líneas que no contienen comentarios o líneas en blanco.

MacOs X: es una serie de sistemas operativos gráficos desarrollados y comercializados por Apple Inc. desde 2001. Es el sistema operativo principal para la familia de computadoras Mac de Apple.

Motor de búsqueda o buscador: sistema informático que distribuye información a los usuarios web a partir de palabras clave, ejemplos de ello son Google o Yahoo.

Multiplataforma: que puede utilizarse en diversos entornos o sistemas operativos.

Navegador web: es un software, aplicación o programa que permite el acceso a la web, interpretando la información de distintos tipos de archivos y sitios web para que estos puedan ser vistos.

Nube: espacio de almacenamiento y procesamiento de datos y archivos ubicado en internet, al que puede acceder el usuario desde cualquier dispositivo.

Núcleo o kernel: es un software que constituye una parte fundamental del sistema operativo, siendo el principal responsable de facilitar a los distintos programas acceso seguro al hardware del ordenador.

Parche: cambios que se aplican a un programa, para corregir errores, agregarle funcionalidad, actualizarlo, etc.

P2P: modelo de comunicación descentralizado, es decir, no se necesita un servidor central, sino que cada parte o usuario actúan por igual y pueden tener la función de servidor o de cliente. Permite el intercambio directo de información, en cualquier formato, entre los ordenadores interconectados.

Servidor de correo: aplicación que nos permite enviar mensajes (correos) de unos usuarios a otros, con independencia de la red que dichos usuarios estén utilizando.

Sistema operativo: consiste en varios programas fundamentales que necesita el ordenador para poder comunicar y recibir instrucciones de los usuarios; tales como leer y escribir datos en el disco duro, cintas, e impresoras; controlar el uso de la memoria; y ejecutar otros programas.

Software: conjunto de programas o aplicaciones, instrucciones y reglas informáticas que hacen posible el funcionamiento del equipo.

Software propietario: el software propietario (también llamado privativo, de código cerrado o software no libre) es cualquier programa informático en el que el usuario tiene limitaciones para usarlo, modificarlo o redistribuirlo (esto último con o sin modificaciones).

Supercomputadora: dispositivo informático con capacidades de cálculo superiores a las computadoras comunes y de escritorio y que son usadas con fines específicos.

Ubuntu: sistema operativo de código abierto.

Usabilidad: proviene del inglés *usability* y hace referencia a la facilidad con que un usuario puede utilizar una herramienta fabricada por otras personas con el fin de alcanzar un cierto objetivo.