

# Proyectos fin de carrera y software libre

Antonio Araúzo Azofra, Antonio J. Cubero Atienza, Lorenzo Salas Morera

**Resumen** – El software libre, cada vez más presente en nuestra sociedad, está adquiriendo una importancia creciente en la Universidad por numerosas razones. Por otra parte, el proyecto fin de carrera es una materia muy importante en las carreras de ingeniería. Este trabajo analiza las relaciones entre ambos, y las posibles ventajas de su combinación, desde el simple uso de software libre en el proyecto fin de carrera, pasando por la liberación del proyecto, hasta la integración del proyecto en la comunidad del software libre y la elección de temas directamente relacionados.

## I. INTRODUCCIÓN

El Proyecto Fin de Carrera (PFC) es probablemente una de las materias a las que más importancia se da al cursar una carrera académica de ingeniería. Se trata de un ejercicio integrador o de síntesis de los conocimientos adquiridos durante toda la carrera, donde se profundiza en algún aspecto concreto o se desarrolla una aplicación definida. Por tanto, es quizá la materia más cercana a la posterior labor profesional.

Los proyectos fin de carrera pueden ser trabajos de ingeniería: de ejecución, de desarrollo o de investigación aplicada; requieren un trabajo considerable; y son realizados por alumnos que son casi ingenieros, pues les queda poco para terminar su carrera. Además, normalmente se trata de trabajos dirigidos o tutelados por profesores de universidad, y a veces en contacto con profesionales. Son, por tanto, unos trabajos de una envergadura y nivel de calidad considerables.

En general, todas las ingenierías incluyen en sus planes de estudios las asignaturas de Proyectos y Proyecto Fin de Carrera. Aunque los planes de estudios de las ingenierías en informática no las incluyen como materias troncales [1], las diferentes Escuelas sí suelen incluir el Proyecto Fin de Carrera (sobre todo en el caso de la ingeniería superior) y algunas la asignatura de Proyectos [2, 3].

Una situación habitual, por desgracia, es que el PFC después de presentado se quede en el olvido. Aunque en algunos casos los proyectos son utilizados para alguna investigación puntual, o en las clases de alguna asignatura, no cabe duda de que podrían ser todavía más útiles para más gente.

Por otra parte, el Software Libre (SL) es aquel en el que

1 Aunque la materia Proyectos de ingeniería no está incluida como tal, sí están parte de sus contenidos dentro de la materia Sistemas Informáticos, que incluye como descriptor Proyectos de Sistemas Informáticos.

Los autores pertenecen al Área de Proyectos de Ingeniería del Departamento de Ingeniería Rural de la Universidad de Córdoba. (Email: arauzo (arroba) uco.es)

su/s autor/es otorgan licencia para ejercer las siguientes cuatro libertades:

- (0) libertad de usarlo, sin ninguna restricción o discriminación, ni económica ni de cualquier otro tipo.
- (1) libertad de estudiar cómo funciona, aprender de él, y adaptarlo (el acceso al código fuente es un requisito previo).
- (2) libertad de distribuir copias, y así ayudar al prójimo.
- (3) libertad de mejorar el programa y hacer públicas las mejoras a los demás, de forma que toda la comunidad se beneficie.

Existe una gran cantidad de personas, empresas y organizaciones, que usan, colaboran y/o desarrollan SL. Entre ellos existe un clima de cooperación infundido por la naturaleza del SL. Nos solemos referir al conjunto de todos ellos como la comunidad del SL.

Las Universidades están apoyando cada vez más el SL, y no son pocas las razones [4] para hacerlo. Hay un anillo web universitario pro SL, con el propósito de ser punto de encuentro de todas las asociaciones que comparten este fin. Además de que cada vez es mayor la implicación de profesores y alumnos, varias universidades han creado Oficinas de SL. Concretamente la Oficina de Software Libre de la Universidad de Cádiz (OSLUCA) ha expresado su interés en promover PFC relacionados con el software libre.

La simbiosis entre los proyectos fin de carrera y el software libre puede generar sinergias positivas muy interesantes para ambos. Un PFC, en vez de permanecer olvidado en una biblioteca, puede ser muy útil para el SL y la Sociedad en general, y a su vez un PFC puede tener grandes ventajas al sumergirse en el mundo del SL. Podría pensarse que esto es así sólo para informáticos, pero en realidad otras titulaciones también pueden aprovechar las ventajas de realizar un PFC relacionado con SL.

En este trabajo trataremos de mostrar las posibilidades que existen, y las ventajas que se pueden obtener, en la combinación de SL y los PFC. Partiremos desde el mero uso de SL en el desarrollo del proyecto, pasando por los detalles de cómo hacer libre un PFC, para terminar comentando cómo se puede lograr la integración completa de un PFC en el mundo y la comunidad del software libre.

## II. USO DE SOFTWARE LIBRE EN EL PROYECTO FIN DE CARRERA

Existe actualmente tal cantidad de software libre que casi cualquier tarea que imaginemos posible realizar con un ordenador puede hacerse con algún programa libre, desde escribir una simple carta, pasando, por ejemplo, por el dibujo de moléculas químicas, hasta la realización de efectos especiales, como los de la película *Titanic*. Es cierto que

existen algunas lagunas, como el reconocimiento automático de caracteres, pero casi siempre hay alguien que está trabajando por eliminarlas. Así, es muy probable que dentro de poco el software libre esté a la cabeza de la innovación en todos los ámbitos.

El uso de SL para el desarrollo de un PFC presenta diversas ventajas. La primera es poder usarlo sin tener que adquirir costosas licencias, “pactar con el diablo” (a veces regalan licencias con fines educativos, pero antes de explotar profesionalmente el proyecto habrá que pasar por caja), ni hacer nada ilegal. También hay programas libres que permiten realizar tareas que no se pueden hacer con programas no libres, o que funcionan mejor que otro no libre. Sin olvidar que incluso podemos adaptar el programa a nuestros intereses. A veces un pequeño cambio puede resultarnos muy útil. Además, colaborar con el SL es bueno para la Sociedad, y simplemente con usarlo ya se está colaborando con él. Porque aquel que lo usa probablemente hablará de él ayudando a difundirlo, enseñará a otros a usarlo, y si encuentra algún fallo o tiene alguna sugerencia puede comunicárselo a los desarrolladores para que lo mejoren.

Como indican en su presentación [5] los miembros del grupo de usuarios de GNU/Linux de la Universidad Carlos III de Madrid, hay muchos programas de SL que pueden ser de interés a la hora de desarrollar un PFC.

La redacción de la memoria puede hacerse con el Writer de OpenOffice, o algún otro procesador de textos libre como Abiword o Kword. LateX es algo más difícil de usar pero se trata de una herramienta profesional de tipografía, que incorpora un soporte excelente de expresiones matemáticas y una gestión de citas bibliográficas muy versátil. Docbook es una alternativa más moderna a LateX para documentación técnica, pero quizá le falta un editor visual y mejorar la gestión de bibliografías.

Para preparar la presentación del proyecto disponemos, entre otros programas, del Impress de OpenOffice, muy completo y fácil de manejar, MagicPoint, que representa una forma distinta de preparar presentaciones, y también LateX puede ser usado para preparar presentaciones.

A la hora de programar disponemos de compiladores e intérpretes para diversos lenguajes (C, C++, Java, Fortran...). Queremos mencionar aquí especialmente a Python, un lenguaje de programación que es completamente libre (al contrario que otros como C++ donde la especificación de sus bibliotecas estándares no es pública, o Java donde éstas son públicas con restricciones), y además goza de un buen diseño, facilidad de aprendizaje y de uso, y una gran mantenibilidad del código. También hay editores con características que facilitan mucho la programación como X/Emacs, y diversos entornos completos de programación.

Podemos encontrar herramientas para trabajo en grupo que pueden ser útiles si el proyecto se desarrolla entre varios alumnos o en colaboración con otros. Por ejemplo, Subversion (SVN), una mejora del conocido CVS, permite trabajar a varias personas sobre los mismos ficheros simultáneamente y llevar un registro de los cambios.

El lenguaje R, y el programa GNUPlot nos permiten generar gráficos y estadísticas de datos que necesitemos analizar. También hay programas para simulación, gestores de contenidos web, y mucho software específico que puede estar relacionado con el tema concreto del proyecto. Por supuesto, debemos incluir los que permiten realizar otras tareas habituales, que pueden ser útiles o incluso necesarias para el desarrollo del PFC: navegar por la web (Mozilla, Galeon, Konqueror, w3m...), email (Mozilla Mail, Evolution, Mutt...), mensajería instantánea (Xchat, Gabber, Gaim, Amsn...), retoque de imágenes (Gimp)...

### III. LIBERAR UN PROYECTO FIN DE CARRERA

Los derechos sobre el PFC, así como los de cualquier trabajo que realice un alumno en el desarrollo de su estudio, pertenecen en principio al alumno como autor de los mismos, según señala la Ley de Propiedad Intelectual [6] en su artículo 1 (el ámbito de esta ley incluye tanto la documentación, como el software, mencionados explícitamente en el artículo 10). Cuando para desarrollar el proyecto se cuente con financiación ajena, por ejemplo de un departamento de investigación universitario, tales como ayudas, becas o contratos, habrá que mirar las condiciones de los mismos para ver si éstas exigen la cesión, ya sea parcial o total, de los derechos. Si el proyecto es dirigido, es posible que parte de los derechos de autor correspondan a los directores, y por tanto quizá pertenezcan a la entidad donde éstos trabajen.

Las bibliotecas de las Escuelas de Ingeniería normalmente guardan copias de los PFC, y permiten su consulta bajo determinadas condiciones. Se considera que esta consulta puede ser útil para la comunidad universitaria, además de ser una garantía de transparencia. En estos casos la normativa de la universidad suele exigir al alumno varias copias del PFC y su consentimiento a que estén en la biblioteca como requisito previo a la presentación del proyecto. Se trata de una licencia o permiso del autor para que la biblioteca muestre su PFC, no de una cesión de los derechos de autor, que seguirán perteneciendo por completo al alumno.

Si el alumno posee los derechos de autor de su proyecto, puede otorgar licencias de uso como él crea conveniente. De esta forma, puede asignar a su proyecto una licencia, otorgada a todo el mundo, que garantice las cuatro libertades antes mencionadas, y desde ese momento el proyecto se podrá considerar libre. En el caso de que, por cualquier razón, no se posean todos los derechos sobre el mismo, habrá que recabar el consentimiento de todos los propietarios.

Dado que el mundo del Derecho es muy complicado, y que puede haber muchos matices, existen una gran cantidad de licencias redactadas. Las hay genéricas y específicas para código fuente, documentación, así como para otros tipos de contenidos como expresiones gráficas o sonidos. Los conceptos de lo que podemos hacer con un contenido libre son fáciles de entender, pero averiguar si una licencia se puede considerar libre o no, puede llegar a ser bastante engorroso. Por esta razón, el proyecto Debian [7] redactó una guía básica de las condiciones que debe cumplir una licencia

para considerarse libre, las directrices de software libre de Debian (Debian Free Software Guidelines, DFSG). Los abogados especializados de este proyecto analizan las diferentes licencias para ver si cumplen las DFSG. Sólo aquellos programas o contenidos que cumplen las directrices son incorporados a la parte principal del sistema operativo Debian GNU/Linux, y la comunidad del software libre adopta este mismo criterio para determinar qué licencias son libres.

Para el software en sí mismo, por lo general se suelen usar las del proyecto GNU [8], no en vano son los primeros promotores del software libre, y todo el mundo conoce sus licencias. Existen dos licencias para código, la GPL y la LGPL.

La licencia GPL (General Public License) es la original y la que más protege la obra. Otorga las cuatro libertades del software libre, pero con una condición importante: las modificaciones o mejoras que se hagan al software podrán ser distribuidas sólo en los mismos términos en los que se ha recibido el software original, esto es, licenciando las modificaciones a su vez con la licencia GPL.

La licencia GPL está muy bien para programas, pero cuando esta licencia se usaba en bibliotecas de programación, la obligación de liberar todo aquello que se enlace con código GPL podía representar un problema. Por ejemplo, podría llegar a ser ilegal crear cualquier tipo de software no libre para GNU/Linux. Por esta razón se creó la licencia LGPL (Lesser General Public License), que es igual a la GPL pero permite que los trabajos derivados de un código bajo licencia LGPL sean licenciados con otras licencias.

Para la documentación se ha estado usando durante bastante tiempo la misma licencia que para el código, pero en GNU pensaron que podía ser conveniente crear una licencia especial para documentación. Como resultado aparece la FDL (Free Documentation License), y aunque resulte paradójico viniendo de GNU, esta licencia tiene características que pueden violar las directrices de Debian (DFSG). Por lo tanto en Debian siguen recomendando [9] utilizar la misma licencia para el código y la documentación, apuntando además que esto tiene la ventaja de permitir copiar partes de código fuente a la documentación y viceversa.

Las licencias de GNU están traducidas en varios idiomas con objeto de que se comprendan. Sin embargo, en las traducciones, no está garantizado que su valor legal sea exactamente el mismo que el de la original en inglés, por lo que se recomienda usar la original.

Para aplicar las licencias al código, documentación o cualquier otro contenido del proyecto, basta con seguir las instrucciones que figuran en el apéndice de las licencias y con mayor detalle en las páginas web de las mismas.

Es importante señalar, que el hecho de liberar un PFC, no implica perder la posibilidad de explotarlo comercialmente. Aunque se está dando la oportunidad de usarlo a cualquiera, los autores serán con toda seguridad los más indicados para realizar la implantación, el mantenimiento, la ampliación, o dar soporte del sistema desarrollado [10,11].

#### IV. INTEGRACIÓN DEL PROYECTO EN LA COMUNIDAD

Un PFC libre alcanza una mayor utilidad para la Sociedad, y también probablemente para su autor, cuando es difundido. La forma más simple de publicarlo es simplemente ponerlo en una página web. Hoy en día existen estupendas herramientas de recuperación de información (buscadores) que probablemente permitirán localizar el proyecto a la gente interesada en él.

Si se quiere, se puede dar un nivel mayor de integración del PFC en el mundo del software libre abriendo el desarrollo a la comunidad[12]. Aunque requiere un esfuerzo mayor, esto puede dar continuidad al proyecto y por tanto más importancia y utilidad. No olvidemos que un programa informático nunca se termina, siempre es mejorable, o requiere mantenimiento para adaptarse a los continuos cambios del entorno. Además, puede ser muy gratificante ver cómo otros lo usan, e incluso colaboran en su desarrollo, y cómo lo que uno ha comenzado puede llegar a ser utilizado en todo el mundo.

En el modelo habitual de desarrollo de software libre suele haber colaboraciones de varios desarrolladores, publicación de versiones frecuentemente, informes de fallos o sugerencias de usuarios, y decisiones abiertas sobre la evolución del proyecto.

Para facilitar el desarrollo se utilizan diversas herramientas, y es casi imprescindible tener una página web del proyecto. En ésta se presentará el proyecto, indicando en qué consiste, el estado en que está el desarrollo y las ideas o planes de desarrollo futuro. Por supuesto, se incluirá en la página web un apartado para descargar tanto el programa como su código fuente, así como un apartado para la documentación, ya sea en línea o descargándola, para consultarla localmente o imprimirla. También se suele incluir un listado de tareas por hacer, instrucciones sobre cómo colaborar, agradecimientos de las colaboraciones recibidas, e información sobre quienes son los desarrolladores principales. Una herramienta interesante, a la que normalmente se podrá acceder desde la web, es el sistema de seguimiento de errores ("bug tracking system"). Éste permite: a los usuarios, informar de los problemas que encuentren en el software; y a los desarrolladores, llevar el control de los problemas que se van arreglando. Finalmente, se suele incluir un apartado para desarrolladores con todos los documentos del desarrollo (diseño, ideas, normas o estilos a seguir...), así como indicaciones sobre la forma de acceder a todos los recursos de desarrolladores. Normalmente, entre estos recursos se cuenta con un almacén de código fuente con sistema de control de versiones (CVS, Subversion). Se suele disponer también de diversas herramientas de comunicación, como las listas de correo, tanto para desarrollo como para dar soporte a los usuarios; o como canales de IRC que también pueden ser usados para ambos fines.

La tarea de poner los servicios comentados en marcha podría ser compleja, pero se simplifica mucho, usando alguno de los portales web que integran todas estas herramientas, y las ponen a disposición de los proyectos de software libre gratuitamente. Hay bastantes, entre ellos los

más conocidos son: SourceForge[13], Savannah[14] del proyecto GNU, y Software-libre.org[15] en español.

Hasta ahora hemos considerado la liberación de un proyecto nuevo, pero hay otras dos opciones, que incluso pueden ser más enriquecedoras. La primera es que el PFC se ligue de alguna forma a un SL ya existente. Un ejemplo de esto podría ser el desarrollo de un interfaz gráfico para un reproductor de música con interfaz de línea de órdenes. La segunda es que el PFC se integre completamente en un proyecto libre existente. Se puede considerar que el PFC sea la incorporación de una funcionalidad concreta, o el desarrollo de un cierto módulo necesario en el proyecto libre. En este caso, será muy importante concretar claramente los objetivos del PFC, diferenciándolos de los del proyecto libre en que se integra. También habrá que tener en cuenta que, a veces, para contribuir aportando código a un proyecto libre se nos pide que cedamos los derechos a una determinada organización, ésta garantizará a cambio que se mantiene como software libre. Será conveniente comprobar que se poseen todos los derechos sobre el código a integrar y las condiciones de la cesión.

La realización de un PFC de este tipo puede proporcionar al alumno que lo desarrolla una experiencia de trabajo real muy valiosa, compartiendo información y código con otros, y utilizando las diversas herramientas de trabajo en grupo que antes hemos comentado. Además, le proporcionará un conocimiento de primera línea sobre la comunidad del software libre y todo este fenómeno que todavía es una novedad en el ámbito empresarial. Lo que puede resultar beneficioso de cara a su carrera profesional porque cada vez hay más empresas trabajando en SL.

Otra ventaja es la reutilización. Al igual que nuestro código podrá ser reutilizado al ser libre, nosotros podremos utilizar otros códigos libres, consiguiendo realizar mejores proyectos con el mismo esfuerzo.

Es necesario que quede muy claro, haciéndolo constar en la memoria y en la presentación, qué es lo que aporta el PFC que se está realizando y qué partes son externas. A pesar de ello, la reutilización sigue siendo ventajosa, no sólo porque el PFC pueda aparentar tener mayor envergadura, sino principalmente porque se estará demostrando capacidad para trabajar en un entorno real en el que ya existe trabajo hecho por otras personas, y se demostrará capacidad para realizar una tarea más importante al conseguir hacer más cosas en el PFC.

La elección del tema del PFC la realiza el alumno, bien a propuesta propia, o bien eligiendo entre los ofertados por los profesores. En ambas opciones ya han aparecido iniciativas con software libre. Hay demanda por parte de los alumnos, que podemos comprobar leyendo algunos foros en internet, y hay profesores, como los del Grupo de Sistemas y Comunicaciones del Departamento de Informática de la Universidad Rey Juan Carlos, que ofertan temas de software libre.

Una última consideración sobre la elección de tema es que en el mundo del software libre se suele evitar “reinventar la

rueda”. Es decir, no se suele iniciar un proyecto de software libre repetido, salvo que se crea que lo que ya existe no puede ser reconducido y el sistema nuevo puede aportar ventajas considerables.

## V. CONCLUSIONES

El proyecto fin de carrera es una magnífica ocasión para el alumno de afrontar un trabajo real muy cercano a su entorno profesional. La experiencia del SL en este sentido puede ser aprovechada como vía que propicia un trabajo completo en varias líneas:

- Análisis de algo ya existente
- Definición clara de líneas de mejora
- Diseño de esta mejora dentro de un marco real.
- Verificación del resultado en un medio en el que incluso podrían participar agentes exteriores.
- Enriquecerse con una experiencia solidaria en un campo aparentemente “sin alma” por lo sofisticado y tecnificado.

## REFERENCIAS

[1] Real Decreto 1459/1990, de 26 de octubre, por el que se establece el título universitario oficial de Ingeniero en Informática y las directrices generales propias de los planes de estudios conducentes a la obtención de aquel.

[2] Torralba Martínez, J.M.(1992) Proyectos de Ingeniería Informática. En *Actas del VIII Congreso Nacional de Ingeniería de Proyectos.*, páginas 17-30.

[3] Mantellán Olivera V. ¿Qué tiene que estudiar un informático? *TodoLinux*, nº 23, páginas 12-13. (También en [16])

[4] René Mérou. Argumentos en favor del software libre en las aulas, <http://bulma.net/body.phtml?nIdNoticia=1831>

[5] Grupo de Usuarios de Linux de la Universidad Carlos III de Madrid. Cómo hacer el Proyecto Fin de Carrera con software libre, <http://gul.uc3m.es/gul/cursos/doc/PFCconSoftwareLibre.pdf>

[6] LEY 22/1987, de 11 de noviembre, de Propiedad Intelectual. BOE 275/1987 de 17 de noviembre.

[7] El proyecto Debian, <http://www.debian.org>

[8] El proyecto GNU, <http://www.gnu.org>

[9] DFSG and Software License FAQ, <http://people.debian.org/~bap/dfsg-faq.html>

[10] Jesús M. González Barahona. El software como servicio. O cómo producir programas libres y no morir en el intento. *TodoLinux*, nº 25, páginas 12-13. (También en [16])

[11] Juan Tomás García, Alfredo Romeo, Cristóbal Prieto. *La pastilla roja*. EditLin, 2003 ISBN:84-932888-5-3

[12] Jesús M. González Barahona. ¿Y cómo hago para que mi código sea libre? *TodoLinux*, nº 30, páginas 12-13. (También en [16])

[13] SourceForge, <https://sourceforge.net>

[14] Savannah, <http://savannah.gnu.org>

[15] #software-libre, <http://software-libre.org/>

[16] Vicente Mantellán Olivera et al. eds. *Sobre software libre: Compilación de ensayos sobre software libre*. Dijusa. 2004 ISBN: 84-9772-402-X