

Experiencia del primer año de dictado de una asignatura electiva sobre Software Libre en la carrera de Ingeniería en Sistemas

Ricardo Medel

Departamento de Ingeniería en Sistemas de Información,
Universidad Tecnológica Nacional, Facultad Regional Córdoba
Maestro M. López esq. Cruz Roja Argentina
Ciudad Universitaria - Córdoba, Argentina
ricardo.h.medel@gmail.com

Resumen Si bien el software libre es claramente beneficioso, útil y hasta necesario para la educación universitaria en informática, algunas carreras no lo incluyen en sus asignaturas, ni en su práctica ni en su análisis teórico. En este artículo se describe una asignatura de remediación de este problema, cuya primera edición se desarrolló en el año 2015 en la Facultad Regional Córdoba de la Universidad Tecnológica Nacional (UTN FRC). Lo novedoso de esta asignatura es que los alumnos, como parte de su cursado, deben participar de una comunidad asociada a un proyecto de software libre previamente existente, analizar su dinámica y realizar una presentación sobre su experiencia.

1. Introducción

El software libre y/o de fuentes abiertas es tan antiguo como las computadoras. En los comienzos de la era de la computadora lo valioso para los negocios era el hardware, mientras que el software era compartido libremente entre los pocos operadores de computadoras [2]. Sin embargo, fue recién en 1984 que Richard Stallman caracterizó al software libre en su Manifiesto GNU [16]. Más tarde, la creación de la Open Source Initiative (Iniciativa por las Fuentes Abiertas) definió al software de fuentes abiertas [12]. Si bien este último incluye al software libre, lo contrario no es cierto. No obstante esta diferencia, consideramos que ambos comparten la misma filosofía de la construcción colectiva y la libertad para utilizar y modificar el software, por lo que en este trabajo utilizaremos el término "software libre" para referirnos indistintamente a ambos tipos de software.

En la actualidad el software libre está presente en gran parte de los sistemas informáticos [4]. Ha sido clave en la construcción de la infraestructura de la Internet [9] y se lo encuentra, puro o en sus muchas derivaciones no libres, hasta en los teléfonos celulares que usamos habitualmente (por ejemplo, en forma del sistema operativo Android) [6].

Si bien técnicamente puede considerarse similar al software no libre, también llamado software privativo, su diferencia no sólo se basa en los

términos de las licencias de uso sino en la estructura de desarrollo y mantenimiento que cada proyecto tiene. Aunque en ciertos proyectos que son llevados adelante exclusivamente por empresas, en la mayoría de los casos los proyectos son creados y mantenidos por una comunidad de desarrolladores y usuarios.

Es por todo esto que es deseable que los practicantes de la disciplina tengan un acabado conocimiento de las características, ventajas y limitaciones del software libre, a fin de poder tomar decisiones informadas respecto al desarrollo y/o uso de este tipo de software durante su práctica profesional.

Debido a su bajo o nulo costo inicial y la posibilidad de acceder al código fuente, el software libre es de uso común en universidades [8,19]. Sin embargo, en nuestra universidad, en particular en la carrera de Ingeniería en Sistemas de Información, el software libre es muy poco utilizado durante las clases, y en los pocos casos en que es utilizado, no se analiza en su carácter de software libre. Por esta razón, en 2014 se propuso al Consejo Departamental el dictado de una asignatura electiva dedicada al tema y que permitiera a los alumnos de quinto año conocer este tipo de software, utilizarlo, analizarlo y así poder tomar decisiones informadas durante su vida profesional [1].

La propuesta fue aprobada por el Consejo Departamental y la asignatura se dictó en el primer cuatrimestre del año 2015. En este artículo presentamos los resultados de la experiencia, las lecciones aprendidas y proponemos cambios para las futuras ediciones.

Debe tenerse en cuenta que esta experiencia no ha sido planteada como un experimento, lo que incluiría un grupo de control y evaluaciones pre- y post-dictado. Simplemente presentamos aquí la experiencia de dictar una asignatura electiva de remediación sobre software libre a un grupo reducido de alumnos de quinto año de la carrera de Ingeniería en Sistemas de Información, con la esperanza de que sea útil para quienes se encuentran en situaciones similares.

En el resto del artículo presentamos la experiencia, incluyendo los detalles del curso, las actividades incluidas y sus resultados. En la tercera sección realizamos un análisis de los resultados y proponemos algunas mejoras al dictado de la asignatura. La última sección presenta las conclusiones de nuestro trabajo.

2. Experiencia

En esta sección se describe la experiencia de dictado de la asignatura, desde su planificación a su ejecución, haciendo especial hincapié en las diferencias entre el plan y lo que finalmente se pudo implementar.

2.1. Plan

En el año 2014 se propuso al Consejo Departamental del Departamento de Ingeniería en Sistemas de Información el dictado de una asignatura electiva para alumnos de quinto año de la Ingeniería en Sistemas de Información [1]. El programa presentado y aprobado por el Consejo proponía

90 horas de clases teórico-prácticas divididas en seis unidades temáticas y evaluaciones, tal como se muestra en la Tabla 1.

Tema	Horas
Conceptos básicos e historia	12
Aspectos legales y sociales	12
Análisis y gestión de proyectos	12
Diseño e implementación de software libre	12
Sistemas de software libre	12
Mantenimiento y calidad de proyectos	6
Evaluaciones	24

Cuadro 1. Temas y carga horaria propuestos

Las evaluaciones propuestas incluían trabajos prácticos, un examen parcial y la participación en una comunidad real de software libre. Esta última evaluación requería que el alumno participe durante varias semanas en un proyecto de software libre realizado por una comunidad pre-existente, efectúe aportes al proyecto, evalúe la comunidad utilizando las herramientas de análisis provistas durante el cursado de la asignatura y realice una presentación escrita y oral de los resultados.

Esta última modalidad tiene antecedentes en prácticas propuestas en la Universidad de Buenos Aires [18] y la Universidad Nacional de Quilmes [17], aunque sus resultados no han sido publicados a la fecha.

Se planteó que los requisitos de aprobación de la asignatura serían tener el 80% de asistencia a las clases (lo estándar en la UTN FRC) y un promedio ponderado de 8 (ocho) en las calificaciones de los trabajos prácticos, el examen parcial y el informe de la práctica en una comunidad de software libre. La modalidad sería de promoción únicamente, sin la posibilidad de regularizar la asignatura y rendir un examen final en algún turno de exámenes posterior.

2.2. Implementación

La asignatura fue aprobada para ser dictada en el primer semestre de 2015, a principios de año se promovió (junto con las otras asignaturas electivas) entre los alumnos en situación de cursar este tipo de asignaturas. Una semana después se abrieron las inscripciones y se recibieron 31 solicitudes.

Durante una reunión departamental se decidió que el día y horario de cursado fuera los sábados de 8 a 10:30 hs. Esto redujo la carga horaria de las 90 horas originales a un máximo posible de 37 horas. Sin embargo, debido a una serie de feriados largos, que impidieron el dictado de clases ciertos sábados, la carga horaria final fue de 25 horas. Esta carga horaria presencial se dividió en 15 horas de clases teórico-prácticas y 10 horas de evaluación (incluyendo 2 hs. 30 min. de examen parcial recuperatorio). El resto de las horas debió ser cubierto por trabajo de los alumnos en su casa.

La Tabla 2 detalla el tema abordado o la evaluación realizada en cada clase de la asignatura.

Clase	Día de clase	Tema
1	28/03	Presentación - Conceptos básicos e historia
2	11/04	Aspectos legales, sociales y económicos
3	18/04	Comunidades y gestión de proyectos
4	25/04	Alumnos presentan la comunidad donde realizar la práctica
5	09/05	Diseño e implementación de software libre
6	16/05	Sistemas de software libre
7	30/05	Examen parcial
8	06/06	Calidad en proyectos
9	13/06	Examen parcial recuperatorio
10	27/06	Presentación del informe de participación en un comunidad

Cuadro 2. Clases y temas

De los 31 alumnos inscriptos, sólo 19 asistieron a clases. De éstos, 3 abandonaron sin haber llegado a realizar ni un trabajo práctico, mientras que otros 2 abandonaron luego de realizar un trabajo práctico. Tres de los cinco alumnos que abandonaron la asignatura lo justificaron por demasiada carga horaria (un alumno) y por viajes relacionados a su trabajo (dos alumnos). Dos alumnos no justificaron su abandono.

Se realizaron tres trabajos prácticos y la participación en una comunidad de software libre a elección del alumno. El trabajo práctico 1 requirió escribir un ensayo comparativo entre dos licencias de software (a elegir entre un conjunto de 6 licencias posibles). El trabajo práctico 2 requería analizar y proponer tres proyectos de software libre donde hacer la práctica, y realizar una presentación oral. A fin de simplificar la búsqueda, la cátedra proveyó un listado sugerido de proyectos. Cada alumno expuso públicamente los resultados de su análisis de al menos tres proyectos y sus razones para elegir uno. El trabajo práctico 3, sin nota, suponía la refactorización de un código escrito en Python.

Las participaciones en comunidades de software libre se realizaron en los proyectos pilas-engine [13] (5 alumnos), OpenStreetMap [11] (3 alumnos), Ninja IDE [10], rst2html5 [15], Ícaro [14], Eoptes [3], Koha [7] y FullCalendar [5]. La Tabla 3 muestra las actividades realizadas por cada alumno en el proyecto seleccionado.

El examen parcial, a diferencia de otras asignaturas, no se "aprobaba" o "desaprobaba", simplemente aportaba con una calificación al promedio general. La opción de examen recuperatorio se brindó a alumnos que hubieran faltado, por cualquier motivo, al examen parcial o quisieran obtener una nota más alta. Finalmente, 14 alumnos completaron el cursado, los trabajos prácticos y el examen parcial. Un alumno tomó el examen parcial recuperatorio por haber faltado al examen parcial por razones de salud.

Alumno	Proyecto	Actividades
1	pilas-engine	Refactoring de tests unitarios. Propuestas de mejoras en la gestión de fallos.
2	pilas-engine	Detección de errores en la documentación.
3	pilas-engine	Detección de errores en soporte a SO Windows. Mejora de la documentación de actores.
4	pilas-engine	Desarrollo de un juego ejemplo. Diseño de fondos de pantalla. Participación en el foro virtual.
5	pilas-engine	Documentación de actores.
6	OpenStreetMap	Agregado y edición de información sobre localidad Santa María (Catamarca).
7	OpenStreetMap	Agregado y edición de información sobre la ciudad de Córdoba.
8	OpenStreetMap	Traducción al castellano de etiquetado de mapas.
9	Ninja-IDE	Intento de corregir un bug. Propuso mejoras al sistema de gestión de fallos.
10	rst2html5	Corrección de 2 bugs.
11	Epopotes	Desarrollo de dos manuales de instalación, configuración y uso.
12	ICARO	Revisión y corrección del tutorial paso-a-paso. Traducción del menú al inglés.
13	LibLime Koha	Colaboración en la traducción al castellano del manual de uso.
14	FullCalendar	Reporte de bugs y solicitud de nueva funcionalidad. Desarrollo de nueva funcionalidad.

Cuadro 3. Proyectos y actividades realizadas

De los 14 alumnos que completaron el cursado, todos aprobaron la asignatura. La nota final N_{se} calculó como el promedio ponderado de las notas de los dos trabajos prácticos (T_1 y T_2), el examen parcial o su recuperatorio, en un único caso, (E) y de la presentación del informe final (F), utilizando la fórmula $N = T_1 * 0,2 + T_2 * 0,2 + E * 0,2 + F * 0,4$. Como resultado, cinco alumnos aprobaron con una nota final de 8 (ocho), cuatro con una nota de 9 (nueve) y cinco con la nota máxima de 10 (diez).

2.3. Encuesta

Al finalizar el cursado, a cada alumno se le envió un correo electrónico solicitando feedback, de la siguiente manera:

Me gustaría que me escribas un email contando qué te pareció la asignatura, si satisfizo tus expectativas, qué estuvo bien y qué se puede mejorar. Desde ya, muchas gracias por tu colaboración.

Diez alumnos respondieron a este requerimiento. Las respuestas pueden resumirse en los siguientes conceptos.

Aspectos positivos:

- Las expectativas se cumplieron o fueron superadas.
- Es importante tener una asignatura que desarrolle el tema del software libre.
- Que se analicen los aspectos sociales, legales y económicos, no solo el tema técnico.
- Que el docente aportara con experiencias propias.
- El ambiente de discusión y de apertura a los aportes de los alumnos.
- La participación en un proyecto de software libre real.
- El uso de email y medios virtuales para una comunicación alumno-docente más activa.

Mejoras posibles:

- Realizar más prácticas utilizando el software estudiado.
- Mostrar más casos reales, preferentemente locales, de uso de software libre.
- Tener las filmas antes de las clases.
- El horario de sábados a las 8 de la mañana.
- Explicar previamente cómo realizar los ensayos solicitados para los trabajos prácticos, por no ser una práctica común en esta carrera.
- Hacer mayor uso de la bibliografía complementaria.
- Debería dictarse también como electiva de tercer año (último año de los alumnos que estudian la carrera de Analista de Sistemas de Información).

3. Discusión de resultados

El plan original proponía 66 horas de clases teórico-prácticas y 24 horas de prácticas con evaluación (trabajos prácticos, parciales y participación en un proyecto de software libre existente). Sin embargo, debido a restricciones en el cursado, la carga horaria tuvo que reducirse considerablemente, a un tercio de lo planificado. A pesar de esta importante restricción y a fuerza de reducir la parte práctica casi exclusivamente a la participación en un proyecto, se lograron elaborar todos los temas planificados en las clases teóricas.

La parte práctica, por otra parte, fue la que sufrió el mayor recorte. No se utilizó ninguna de las herramientas libres que se vieron en las clases teóricas. Además de la participación en un proyecto de software libre ya establecido, las prácticas se redujeron al análisis de distintas licencias y la aplicación de técnicas de *refactoring* sobre código Python provisto por la cátedra.

A pesar de esto, pudo notarse una gran disparidad en los conocimientos de los alumnos respecto tanto a herramientas de software como a las técnicas que éstas implementan. Por ejemplo, al presentar herramientas libres que permiten realizar la integración continua, una práctica relativamente común en la ingeniería de software, la discusión en clase permitió apreciar una importante diferencia entre aquellos alumnos que ya conocían la práctica y aquellos que no. Cabe suponer que realizar

prácticas en esas condiciones hubiera sido un desafío para el docente y los alumnos, ya que algunos simplemente hubieran ejercitado sus conocimientos previos utilizando las herramientas libres, mientras que otros habrían tenido que esforzarse mucho más al aprender primero el concepto y luego ejercitarlo por primera vez con las herramientas propuestas.

La selección de proyectos de software libre donde participar fue todo un desafío, tanto por lo novedoso como por la exigencia de tener resultados mensurables en tan poco tiempo. La selección y análisis de hasta tres proyectos por alumno resultó provechosa, aunque hubo una fuerte tendencia a elegir uno de los proyectos sugeridos por el profesor. Sólo tres alumnos propusieron y participaron de proyectos no incluidos en la lista de proyectos sugeridos. Los proyectos propuestos por los alumnos fueron Eptotes, LibLime Koha y FullCalendar.

La participación de los alumnos en proyectos de software libre, a pesar del poco tiempo disponible (la selección del proyecto se hizo el 25 de abril y el informe final se presentó el 27 de junio), fue satisfactoria tanto para los alumnos como para el docente. Si bien el objetivo de esta práctica fue analizar la dinámica de las comunidades asociadas a cada proyecto libre, las participaciones que produjeron análisis más ricos fueron las que incluyeron la corrección de bugs, desarrollo de una nueva funcionalidad (alumnos 10 y 14, ver Tabla 3) y el reporte de bugs. La mayoría de los alumnos tuvo participación en la creación, traducción y corrección de documentación y tutoriales, lo que requirió cierta interacción con la comunidad y, por lo tanto, produjo buenos reportes. En algunos casos incluso se propuso a las comunidades mejoras en los procesos o herramientas de gestión. Las participaciones más pobres, en cuanto a interacción con la comunidad del proyecto y, por ende, su capacidad de analizar su dinámica, fueron las que proveyeron información cartográfica a OpenStreetMap (alumnos 6, 7 y 8). Cabe destacar que en dos casos los alumnos debieron cambiar de proyecto debido a la falta de respuesta de los responsables.

Las presentaciones escritas y orales de los alumnos fueron creciendo en complejidad y calidad, siendo las primeras (un comparativo de licencias) muy pobres y poco estructuradas, mientras que todas las presentaciones finales fueron muy buenas, analizando la estructura de las comunidades, los patrones de participación de sus miembros, las técnicas y herramientas de gestión, y la actitud de los integrantes.

Respecto a la asistencia y aprobación, no hay una explicación clara de por qué se inscribieron 31 alumnos y sólo comenzaron 19. Por otra parte, la gran mayoría de los que asistieron aprobaron la materia, ya que de los 19 alumnos que asistieron a alguna clase, 14 que cursaron la asignatura completa (con una asistencia promedio de 81 %) y la promocionaron con una nota promedio de 9. Tres de los 5 alumnos que abandonaron la asignatura lo justificaron por problemas de horario.

Las encuestas muestran una aprobación general del desarrollo de la asignatura entre los alumnos, con algunos puntos a mejorar.

4. Conclusiones

En este artículo se presenta una experiencia de desarrollo de una asignatura electiva sobre software libre, para alumnos de quinto (último) año de la Ingeniería en Sistemas de Información. Esta asignatura viene a cubrir la falencia en el uso y análisis de software libre durante la mencionada carrera.

Se reporta la primera edición de la asignatura en 2015, que como características de relevancia tuvo un importante recorte de las horas de clase pautadas en el proyecto original (se desarrolló un 30 % de la carga horaria estipulada inicialmente) y la participación de los alumnos en diferentes proyectos de software libre ya establecidos.

El 75 % de los alumnos promocionó la asignatura con una nota promedio de 9. El 25 % restante abandonó no más allá de la tercera clase. Una encuesta muestra un alto grado de satisfacción de los alumnos con los temas tratados y la forma de encararlos.

Cabe acotar que uno de los alumnos refirió en la encuesta que “hasta hace unos meses no tenía noción de lo que era el software libre” que “ni siquiera se me ocurre qué definición habré tenido en su momento”. En base a este comentario, se propone para la segunda edición, a desarrollarse en 2016, realizar una encuesta inicial a los alumnos, preguntando exclusivamente qué consideran ellos que es el software libre. La misma pregunta entonces se hará al final del cursado y se compararán los resultados, a fin de evaluar la adquisición de conocimientos sobre el tema.

Entre las principales mejoras que se plantean para el próximo año es realizar algunas prácticas utilizando herramientas de software libre, definir previamente la estructura y contenido de las entregas que deben realizar los alumnos, ya que no están acostumbrados a escribir ensayos como parte de la aprobación de las asignaturas, y asegurar que la participación en una comunidad de software libre debe permitir el interactuar con dicha comunidad, a fin de realizar un análisis cabal de la dinámica de los miembros del proyecto.

Referencias

1. Cabral, J., R. Medel, N. Navarro y M. Reingart: *Propuesta de incorporación de la Ingeniería de Software Libre y de Código Abierto al currículo de Ingeniería en Sistemas de Información y carreras afines*. En *Anales de las XLIII Jornadas Argentinas de Informática e Investigación Operativa (43 JAIIO) - I Simposio Argentino de Tecnología y Sociedad (STS)*, páginas 286–300. Sociedad Argentina de Informática e Investigación Operativa (SADIO), Septiembre 2014.
2. DiBona, Chris, Sam Ockman y Mark Stone: *Open Sources: Voices from the Open Source Revolution*. O'Reilly Media, enero 1999.
3. *Epothes*. <http://www.epothes.org/>.
4. Feller, Joseph, Brian Fitzgerald, Scott A. Hissam y Karim R. Lakhami: *Perspectives on Free and Open Source Software*. The MIT Press, 2005.
5. *FullCalendar*. <http://fullcalendar.io/>.

6. Hoffman, Chris: *Android is Based on Linux, But What Does That Mean?* Hot-to Geek (<http://www.howtogeek.com>), diciembre 2014.
7. *LibLime Koha*. <http://www.koha.org/>.
8. Lakhan, Shaheen E. y Kavita Jhunjhunwala: *Open Source Software in Education*. EDUCAUSE Quarterly, 31, abril-junio 2008.
9. Newman, Nathan: *The Origins and Future of Open Source Software: A NetAction White Paper*. <http://www.netaction.org/opensrc/future/oss-whole.html>, 1999.
10. *Ninja-IDE*. <http://ninja-ide.org/>.
11. *OpenStreetMap*. <http://www.openstreetmap.org/>.
12. *Open Source Initiative*. <https://opensource.org/>.
13. *pilas-engine*. <http://pilas-engine.com.ar/>.
14. *Proyecto ICARO*. <http://roboticaro.org/>.
15. *rst2html5*. <http://marianoguerra.github.io/rst2html5/>.
16. Stallman, Richard: *El Manifiesto de GNU*. <http://www.gnu.org/gnu/manifiesto.es.html>.
17. Universidad Nacional de Quilmes: *Plan de Estudios Licenciatura en Desarrollo de Software*. www.unq.edu.ar/advf/documentos/50a6a6ca41fd3.pdf. (visitado el 30/06/2014).
18. Wachenchauzer, Rosita: *Trabajos de Carreras de Informática en Comunidades de Código Abierto*. En *Anales de las Jornadas Argentinas de Software Libre 2013*, páginas 130–140, 2013.
19. Wilson, Scott: *Open source in higher education: how far have we come?* The Guardian, marzo 2013.