

Propuesta de incorporación de la Ingeniería de Software Libre y de Código Abierto al currículo de Ingeniería en Sistemas de Información y carreras afines

Juan B. Cabral¹; Ricardo Medel¹; Nestor Navarro¹; Mariano Reingart²

¹ Universidad Tecnológica Nacional, Facultad Regional Córdoba, Argentina

² Instituto Superior Tecnológico Blaise Pascal, Buenos Aires, Argentina

{jbc.develop,ricardo.h.medel.nestornav,reingart}@gmail.com

Abstract

Desde sus inicios el software libre/de código abierto ha generado intensos debates sobre la propiedad intelectual, las libertades del usuario y las formas de creación y distribución de software. Sin embargo, en estos últimos años el debate ha disminuido notablemente ya que ha demostrado ser viable técnica y económicamente, como lo demuestra el hecho de que algunas de las empresas y sistemas más utilizados en la actualidad son o están basados en software libre. El sistema universitario argentino no ha estado libre ni del debate ni de una temprana adopción tanto del software libre como de los principios que dicho software encarna. Sin embargo, existe una carencia en cuanto a asignaturas que cubran tanto los aspectos metodológicos del desarrollo de software libre como los complejos aspectos legales y los siempre cambiantes modelos de negocios. En este trabajo justificamos la importancia de incluir la enseñanza de metodologías de desarrollo de software libre, proponiendo el dictado de un curso de “Ingeniería de software libre” integrado al currículo de la carrera de Ingeniería en Sistemas de Información o afines con el propósito de que los graduados comprendan las particularidades filosóficas, económicas, legales y técnicas de este cambio de paradigma en la ingeniería de software.

Palabras Clave

Software libre, ingeniería de software, código abierto, educación, currículo.

Introducción

El software libre/de fuentes abiertas (FLOSS, por sus siglas en inglés y español: Free/Libre/Open-Source Software) ha, sin dudas, revolucionado la forma en que se crea y distribuye el software. Los orígenes del movimiento que produjo tal cambio en la industria del software pueden trazarse en la creación en 1983 del Proyecto GNU por

Richard Stallman [1]. Fue Stallman quien en su GNU Manifiesto [2] estableció las cuatro libertades que permiten definir a un software como libre:

1. La libertad de usar el programa, con cualquier propósito.
2. La libertad de estudiar cómo funciona el programa y modificarlo, adaptándolo a sus necesidades.
3. La libertad de distribuir copias del programa, con lo cual se puede ayudar al prójimo.
4. La libertad de mejorar el programa y hacer públicas esas mejoras a los demás, de modo que toda la comunidad se beneficie.

Esta lista define implícitamente un marco legal y técnico para el FLOSS. En particular las libertades 2 y 4, ya que requieren el acceso al código fuente para ser ejercidas.

Otros dos hitos remarcables en la historia del FLOSS son el desarrollo de Linux, comenzado en agosto de 1991 por Linus Torvalds y que en conjunto con las herramientas GNU ha llegado a ser el sistema operativo libre (GNU/Linux) más utilizado en la actualidad [4], y la creación en 1998 de la Iniciativa de Código Abierto (OSI, por sus siglas en inglés: Open Source Initiative), la cual intenta diferenciarse del percibido extremismo ideológico del Proyecto GNU y ofrece una visión estrictamente técnica sobre las ventajas de brindar el acceso al código de los programas [5]. Vistos en perspectiva, estos dos desarrollos marcan un punto de quiebre en la historia del FLOSS y establecieron las bases para su difusión actual, tanto en el ámbito de los desarrolladores independientes como de las grandes empresas.

En la actualidad un importante número de sistemas son software libre o están basados en él, y gran número de empresas han desarrollado negocios millonarios utilizándolos [6]. En particular los sistemas basados en LAMP (Linux, Apache, MySQL y PHP) han tenido un éxito notable [7], mientras que el sistema operativo Android, basado en Linux, está presente en el 75% de los teléfonos inteligentes (*smartphones*) y *tablets* en el mercado [8]. Empresas transnacionales producen millones de dólares en ganancias con modelos de negocio basados en software libre, tales como Google, Red Hat y Facebook. Incluso empresas con importantes intereses en el software privativo se cuentan entre las mayores contribuyentes a la Fundación Linux, tales como Microsoft, Intel e IBM [9].

Es por esto que podemos afirmar que hoy toda actividad de desarrollo de software incluye de una u otra manera algo de FLOSS y los egresados de la carrera de Ingeniería en Sistemas de Información, y otras similares, deben tener conocimientos de los temas técnicos, legales y, sobre todo, metodológicos involucrados en el uso y desarrollo de software libre y de código abierto.

El más famoso análisis sobre cómo el FLOSS afecta al desarrollo de software, fue realizado tempranamente, en 1999, por Eric Raymond [22]. En su serie de artículos, que luego fueron publicados como libro, compara al desarrollo del software cerrado (privativo) con la construcción planificada de una catedral, mientras que el software libre y de código abierto, que incluye una administración distribuida, *forks* (trabajos derivados de un original) y apertura a recibir aportes de distintas fuentes, se compara con la creación y crecimiento de un bazar, un mercado persa, en apariencia caótico pero con relaciones funcionales y organizacionales complejas y que cumple con su cometido a la perfección.

Nuestra propuesta es definir e implementar una asignatura electiva para los últimos años de la carrera de Ingeniería en Sistemas de Información en la que se estudien metodologías de desarrollo para proyectos de FLOSS, sus implicancias legales y los factores de negocios. En las siguientes secciones explicamos los antecedentes y justificación, estudios realizados, los resultados obtenidos y discutimos las ventajas y limitaciones de nuestra propuesta.

Antecedentes

En Argentina el concepto de FLOSS se introdujo tempranamente en el ámbito académico universitario, en particular debido a sus ventajas económicas, ya que buena parte del software libre es en la práctica gratuito, y desde allí se extendió paulatinamente a la industria. En un principio el software libre fue utilizado como herramienta de las prácticas realizadas en clase y en proyectos de investigación, debido a su costo nulo y a la posibilidad de la libre distribución [10, 11, 12, 13]. Al poco tiempo algunos grupos comenzaron a desarrollar “distribuciones” GNU/Linux, adaptadas a las necesidades, reales o percibidas, del entorno local, tales como Ututo [14], Tuquito [15], Lihuen [16] y Lisex [17]. Más adelante, proyectos de software libre comenzaron a desarrollarse en las universidades, principalmente para suplir sistemas de gestión [18, 19, 20, 21].

Más recientemente, se plantea la posibilidad de incorporar a las prácticas de las asignaturas la participación de los alumnos en proyectos reales de software libre, ya sea en Tesis de Grado o Trabajos Profesionales, como se ejemplifica en un análisis de las modalidades a las que pueden optar de los estudiantes de Informática de la Facultad de Ingeniería de la UBA [46].

En este sentido, aproximadamente desde 2009 se inició “Proyecto Institucional de la carrera de Análisis de Sistemas orientado al software libre”, que incluía un Espacio de Definición Institucional (materia EDI 3 según el plan de estudios), con temáticas de apoyo para el trabajo final de los alumnos, relacionadas con conceptos de esta propuesta. A partir de 2012, dicho espacio se reorganizó en la materia “Práctica Profesional”, incorporando y

ampliando los contenidos, triplicando en 2014 la carga horaria, con un claro enfoque al software libre para la producción del trabajo final integrador [44].

A partir del año 2011, en la reciente carrera “Licenciatura en Desarrollo de Software” de la Universidad Nacional de Quilmes se dicta la materia “Participación y Gestión en Proyectos de Software Libre”, que según el plan de estudios [43] busca principalmente una “Experiencia concreta de participación en al menos un proyecto existente” (luego de analizar conceptos como “cibercultura y cultura hacker”, “movimiento de software libre”, y un estudio de herramientas de desarrollo y el funcionamiento de proyectos).

Por la información disponible, un análisis preliminar indicaría que en ambos casos se trataría de contenidos introductorios y no totalmente estructurados desde las disciplinas de Ingeniería de Software, más orientados a las actividades prácticas y experimentales.

En el ámbito internacional, podemos citar la materia “Ingeniería del software en entornos del software libre” de la Universidad Abierta de Cataluña [41], que si bien presenta un enfoque más cercano al de esta propuesta, contiene unidades demasiado genéricas orientadas a la Ingeniería de Software más tradicional. Por otro lado, otros acercamientos como la materia “Desenvolvemento de Software Livre” [42] de la Universidad de San Pablo, están mucho más enfocados al desarrollo de Software Libre pero por los tópicos publicados no contemplan temas de Ingeniería de Software en sí que podrían ampliar la utilidad de este tipo de cursos.

Por lo expuesto, no tenemos conocimiento a la fecha de asignaturas dedicadas al estudio de metodologías para desarrollo de software libre enfocadas desde ingeniería de software en Argentina.

Fundamentación

Según la Resolución 786/09 del Ministerio de Educación [42], que busca definir los criterios básicos y estándares para las carreras relacionadas (informática, sistemas y ciencias de la computación), el "Software Libre" debería ser un contenido curricular básico del área "Aspectos Profesionales y Sociales".

El FLOSS no solo es un contenido fundamental, también podría fácilmente cubrir otros temas de dicha área dada su diversidad, extensión y características únicas:

- Historia de la Computación
- Responsabilidad y Ética Profesional.
- Computación y Sociedad.
- Propiedad intelectual, licenciamiento de software y contratos informáticos.
- Aspectos legales.

La carga horaria obligatoria recomendada para este área "Aspectos Profesionales y Sociales" es 50hs, lo que sería adecuado mínimamente para una materia teórica. Esto puede complementarse desde las 1000 horas adicionales de la que disponen las instituciones para establecer las orientaciones y contenidos específicos que consideren más adecuados. A su vez, otra área "Ingeniería de Software, Bases de Datos y Sistemas de Información" tiene asignadas entre 400hs y 650hs, desde donde también podrían adecuarse contenidos de software libre que se indican en esta propuesta.

Las ventajas y oportunidades de presentar e involucrar a los alumnos en proyectos de software libre han sido reconocidas por varias investigaciones (incluso a nivel profesional para su currículum [40]), pudiendo citar como resumen un párrafo de Rosita Wachenchauer, una destacada docente de la UBA y otras instituciones [46]:

La participación de estudiantes de carreras de Informática en proyectos de código abierto es entonces una buena idea porque les permite ser parte un grupo de innovación abierta y, si sus contribuciones son aceptadas, atraer la atención de sus pares en la comunidad, lo cual podría redundar en estatus, oportunidades de trabajo o dinero. Esta participación además les provee la posibilidad de ser parte de la construcción de un conocimiento para uso colectivo, a partir de diversos sistemas de conocimiento. Pero hay más: ... esta participación les permitir a entrenarse en una modalidad de trabajo que refleja cada vez más a la manera como las empresas de alta tecnología enfocan sus desarrollos.

El uso de software libre / código abierto en contextos pedagógicos de Ingeniería de Software ha sido analizado en varias investigaciones con conclusiones satisfactorias [38, 40]. Se resaltan los elementos comunes entre la gestión de proyectos FLOSS y su contrapartida en las áreas de la "Guía del cuerpo de conocimiento de la Ingeniería del Software" (SWEBOK¹) y el "Conocimiento Educativo de la Ingeniería de Software" (SEEK²), que se resumen a continuación.

Temas esenciales como la gestión de configuración y calidad pueden ser abordados más profundamente con software libre, y la comparación con el software propietario puede ser un punto de partida para analizar aspectos sociales (ética, licenciamiento, etc.).

-
- 1 P. Bourque and R.E. Fairley, eds., Guide to the Software Engineering Body of Knowledge, Version 3.0, IEEE Computer Society, 2014; www.swebok.org
<http://www.computer.org/portal/web/swebok>
 - 2 Sobel, Ann EK. "Computing Curricula--Software Engineering Volume." First Draft of the Software Engineering Education Knowledge (SEEK) (2002).
<http://sites.computer.org/ccse/know/FinalDraft.pdf> (ACM / IEEE)

También se notan algunas falencias, como la falta de documentación completa o inexistencia del soporte técnico en ciertas ocasiones, lo que podría presentar dificultades para un aprendizaje independiente (sin acompañamiento docente), pero por otro lado, podría ayudar a vigorizar el currículum de los profesores.

Otro aspecto importante es la disponibilidad de herramientas libres/abiertas CASE (Ingeniería de Software Asistida por Computadora). Aunque si bien no todas son amigables y están tan avanzadas como las comerciales, podrían permitir a los alumnos desarrollar incrementalmente su propia colección de herramientas a un mínimo costo de manera más especializada.

Por último, el análisis de código fuente escrito por otros posibilitaría un enfoque más constructivista y habilitaría varias direcciones para contribuir a los diversos proyectos FLOSS (especialmente sobre mantenimiento, de gran incidencia en la práctica, pero con poca cobertura en los cursos de Ingeniería de Software más tradicionales):

La disponibilidad del código fuente del OSS provee una oportunidad única de experimentación para los educadores. Comparado con los ejemplos teóricos de “juguete” de los libros de texto, los contextos OSS del “mundo real” pueden frecuentemente proveer mejores oportunidades para enseñar conceptos intrincados.

Elementos del trabajo y metodología

Con el objetivo de definir una propuesta para la incorporación del estudio de las metodologías de desarrollo de software libre en una carrera de sistemas, computación o informática, y a fin de concretizar la propuesta basada en una realidad que conocemos, primeramente seleccionamos la carrera de Ingeniería en Sistemas de Información, dado que nuestro ámbito de estudio y desarrollo es la Facultad Regional Córdoba de la Universidad Tecnológica Nacional.

El primer paso fue realizar un análisis de la bibliografía sobre el tema, comenzando con el Manifiesto GNU [2], estudios comparativos de licencias de FLOSS [23] y el libro “La catedral y el bazar”, del ya mencionado Eric Raymond [22], entre otros. En esta etapa, resultaron de gran ayuda los contenidos de la Maestría en Software Libre de la Universidad Abierta de Cataluña, que está cursando uno de los autores. Esta recopilación bibliográfica se completó con una serie de anuncios de prensa y decisiones legales (leyes, resoluciones, ordenanzas y decretos) nacionales, regionales e internacionales con respecto al software libre [24, 25, 26].

En base a dichos estudios se realizó una primera propuesta, que fue socializada a través de una serie de presentaciones, charlas y discusiones en el ámbito de las comunidades de

software libre, tales como las comunidades Python Argentina - PyAr [27], SciPyAr [28], ArPug [29] y en particular las comunidades de desarrollo de web2py [30] y Pylas [31]. Más específicamente, el tema se trató en los eventos locales de la comunidad Python Argentina: PyDay González Catán 2011, Pyday Córdoba 2011 y Django Day Córdoba 2011, y dos eventos internacionales: PyCon Argentina 2012 y SciPyCon Argentina 2013, así como una charla en el Ciclo de Charlas 2012 del grupo UNI-CODE (en la UTN Facultad Regional Córdoba) y una presentación en el Congreso Nacional de Estudiantes de Ingeniería de Sistemas de Información - CNEISI 2011 [32], con una asistencia de aproximadamente 500 personas.

Los comentarios recibidos durante todas estas actividades permitieron realizar mejoras sobre la primera propuesta, en particular sobre los temas de manejo de requerimientos, administración de la comunidad, reporte y corrección de errores y control de versiones.

Resultados

En base al estudio y las consultas realizadas, se definieron los objetivos de la asignatura electiva propuesta. Se pretende que el alumno al finalizar la asignatura alcance los siguientes objetivos generales:

- Comprender las ventajas técnicas, económicas y filosóficas del FLOSS.
- Aprender sobre las herramientas más populares actuales dentro de la comunidad FLOSS.
- Comprender la problemática actual sobre los bienes intangibles.
- Saber identificar las licencias correctas para distintos proyectos de software.
- Ser capaz de diseñar soluciones FLOSS económica y técnicamente viables.

Para lograr estos objetivos se propone dividir la asignatura cuatrimestral en cinco unidades temáticas bien diferenciadas y cada una atacando un aspecto del desarrollo de FLOSS, tal como se muestra en la Tabla 1. A fin de sentar las bases para su estudio en profundidad, es necesario dedicar la primera unidad a temas de la historia y filosofía del movimiento del FLOSS. La segunda unidad tiene una continuidad conceptual con la primera, ya que se dedica a los distintos tipos de licenciamiento del FLOSS, que no es otra cosa que la concretización legal de los principios filosóficos del movimiento. Luego se deben comenzar a tratar los temas más técnicos, por lo que la tercera unidad introduce temas de diseño y desarrollo de software libre. Un tema de mucha relevancia es la forma de gestión de un grupo de desarrollo, que en FLOSS puede transformarse, y es deseable que así sea, en una comunidad de desarrollo, con reglas e interrelaciones de variada complejidad, por lo que la cuarta unidad es dedicada a estudiar la gestión de proyectos FLOSS. Como último tema, pero no menos importante, en la quinta unidad se analizan los distintos modelos de negocios de y alrededor del software libre.

Asignatura	Ingeniería de Software Libre
Carga horaria	6 horas semanales
Régimen	Cuatrimestral
Objetivos	<p>Se pretende que el alumno al finalizar la asignatura alcance los siguientes objetivos generales:</p> <ul style="list-style-type: none"> • Comprender las ventajas técnicas, económicas y filosóficas del Software Libre (SL). • Aprender sobre las herramientas más populares actuales dentro de la comunidad SL. • Comprender la problemática actual sobre los bienes intangibles. • Saber identificar las licencias correctas para distintos proyectos de software. • Diseñar soluciones SL económica y técnicamente viables.
Programa Analítico	
Unidad 1 - Historia y filosofía del movimiento de Software Libre	
<p>Objetivos específicos:</p> <ul style="list-style-type: none"> • Conocer la historia del movimiento de software libre y comprender cómo ésta afecta a las comunidades de desarrollo actuales. • Comprender las diferencias filosóficas de las distintas vertientes del movimiento de software libre. <p>Contenidos:</p> <ul style="list-style-type: none"> • Orígenes: Proyecto GNU, GNU Manifesto, Free Software Foundation y las 4 libertades. • Éxitos técnicos: Emacs, GCC y Linux. • Cisma: la filosofía del código abierto y la creación de la Open Source Initiative. • La comunidad como generadora de contenido y de productos. 	
Unidad 2 - Temas legales y licencias de Software Libre	
<p>Objetivos específicos:</p> <ul style="list-style-type: none"> • Comprender las licencias de software libre, sus diferencias filosóficas y legales, su interacción y su importancia en los proyectos de desarrollo de software. • Entender la problemática de las patentes de software y los sistemas de DRM (Gestión Digital de Derechos). <p>Contenidos:</p> <ul style="list-style-type: none"> • Licencias privativas y libres. Dominio Público. Familia licencias GPL, BSD, Apache, MIT y Mozilla Public License. • Creative Commons. • Patentes de software, DRM y otros mecanismos de control de copias. 	
Unidad 3 - Diseño y Desarrollo de Software Libre	
<p>Objetivos específicos:</p> <ul style="list-style-type: none"> • Estudiar distintas metodologías de desarrollo de software a la luz de los proyectos de desarrollo de software libre. <p>Contenidos:</p> <ul style="list-style-type: none"> • La Catedral y el Bazar. • Filosofía Unix de Diseño. • Métricas de Software libre y planificación. • Patrones: mitos y realidades. • Herramientas comunes en el diseño de proyectos open source. 	
Unidad 4 - Gestión de proyectos de Software Libre	
<p>Objetivos específicos:</p> <ul style="list-style-type: none"> • Comprender la estructura de los distintos tipos de comunidades de software libre. • Comprender la filosofía organizativa de la cultura hacker y las motivaciones de sus miembros. • Aprender a administrar las contribuciones de una comunidad basado en las prioridades del proyecto. <p>Contenidos:</p> <ul style="list-style-type: none"> • Administración de comunidades. La figura del BDFL (Benevolent Dictator for life). • Motivaciones de los miembros de una comunidad de software libre. • Manejo de contribuciones y documentación. • Ventajas y desventajas de los forks. 	
Unidad 5 - Modelos de Negocios para Software Libre	
<p>Objetivos específicos:</p> <ul style="list-style-type: none"> • Aprender cómo seleccionar el modelo de negocios más adecuado para cada proyecto de desarrollo de software libre. <p>Contenidos:</p> <ul style="list-style-type: none"> • Modelos de negocios con y para software libre. • Estudios de casos: éxitos y fracasos. 	
Asignaturas correlativas	<p>Se requieren los conocimientos desarrollados en las siguientes asignaturas:</p> <ul style="list-style-type: none"> • Algoritmos y Estructuras de Datos • Diseño de Sistemas • Ingeniería de Software (deseable) • Legislación y Economía (deseable)

Tabla 1. Unidades temáticas de la propuesta de asignatura electiva.

La unidad 1 estudia específicamente la historia del movimiento del FLOSS, incluyendo la creación del Proyecto GNU y la Fundación Software Libre (FSF, por sus siglas en inglés: *Free Software Foundation*), el cisma producido con la creación de la Iniciativa de Código Abierto (OSI), sus diferencias filosóficas y cómo esto afectó y afecta a la difusión del FLOSS. Estas etapas históricas permitirán, además, comprender el origen de las formas de organización de las diferentes comunidades FLOSS.

La unidad 2 trata sobre la concretización de las diferentes filosofías del FLOSS en sistemas de licenciamiento del software y su relación con la creación y distribución de otros productos intelectuales. El estudio y comparación de la gran variedad de licencias de software libre/de código abierto es un proceso tedioso y complejo, pero necesario para poder comprender las dificultades que muchas empresas de desarrollo de software encuentran cuando quieren desarrollar sistemas basados en FLOSS.

La unidad 3 se enfoca en estudiar cómo el FLOSS ha modificado las metodologías de desarrollo de software. Para esta unidad se requiere un conocimiento previo de los ciclos de vida del software y diversas metodologías de desarrollo. Esta unidad es el punto central de la asignatura, ya que, desde nuestra experiencia, las metodologías tradicionales de desarrollo de software tienen que ser, en el mejor de los casos, adaptadas para el desarrollo de FLOSS. En casos extremos se han desarrollado metodologías completamente nuevas para adaptarse a la nueva estructura de los grupos de desarrollo.

La unidad 4 se relaciona fuertemente con la anterior, abocándose en particular a la administración de comunidades de desarrollo de FLOSS. Estas comunidades usualmente están extendidas por todo el mundo y, a veces, sus miembros tienen intereses diversos. Su administración requiere conocimientos tanto técnicos, de informática y de negocios, como sociológicos y psicológicos, donde la motivación de los participantes es un tema clave que debe trabajarse [33, 34].

Por último, la unidad 5 trata sobre el importante tema de los modelos de negocios asociados al FLOSS. Este es, a nuestro entender, el mayor aporte del FLOSS a la comunidad: una nueva forma de ver los negocios, con la responsabilidad social no solo como un agregado de marketing a la empresa sino como un objetivo de negocios por sí mismo. Las fuentes de ingresos y formas de interacción con los usuarios y otras empresas son tan diversas como sistemas FLOSS existen. En particular, el estudio de casos de éxito y fracasos proveen una buena noción de los pasos que hay que seguir para lograr un proyecto basado en FLOSS económicamente viable.

En base a esta propuesta se nominó una asignatura electiva cuatrimestral en el Departamento de Ingeniería en Sistemas de Información de la Universidad Tecnológica Nacional - Facultad Regional Córdoba, la cual no fue aprobada para 2013. Además, este

documento también sirvió como referencia en la re-formulación de la materia final de la carrera terciaria orientada al software libre en el Instituto Blaise Pascal de la ciudad de González Catán en la provincia de Buenos Aires (actualmente en curso, ver antecedentes).

Discusión

Durante nuestra práctica profesional hemos detectado un indudable aumento de la demanda de conocimientos en el uso y desarrollo de herramientas FLOSS, tanto en la academia como en la naciente industria regional de software. Un indicador concreto de dicho aumento es la creciente formación de cámaras de empresas dedicadas al software libre [35, 36] y grupos de usuarios, resoluciones políticas de alto impacto mediático, como la expresada explícitamente en los artículos 45 y 46 en la reunión en Montevideo del Mercosur [37] y la inclusión de *tracks* o simposios sobre software libre en toda conferencia o jornadas de la disciplina.

Los comentarios recibidos durante nuestras presentaciones, tanto formales como informales, también apuntan en ese sentido. En particular durante la mencionada presentación en el CNEISI 2011 [32], donde se recibió un *feedback* muy positivo de parte de los 500 alumnos asistentes, pertenecientes a todas las facultades regionales de la Universidad Tecnológica Nacional, quienes en su mayoría consideraron necesario la incorporación en la currícula de los temas en cuestión.

Sin embargo, debido al fracaso de nuestra propuesta en el ámbito del Departamento de Ingeniería en Sistemas de Información de la UTN-FRC, consideramos que debemos cambiar de estrategia de construcción de los saberes relevantes para el uso y desarrollo de FLOSS.

En cuanto a la estructura de la asignatura propuesta, un punto que ha sido discutido en varias oportunidades es cuánta importancia asignarle, y por lo tanto cuánto tiempo dedicarle, a las metodologías de desarrollo de FLOSS. En nuestra propuesta estos temas son tratados principalmente en la unidad 3 y en menor medida en la unidad 4, lo que le asigna entre uno y dos quintos del tiempo total de la asignatura. Se ha propuesto expandir dichas unidades a cuatro o cinco unidades, cada una dedicada a distintas etapas del ciclo de vida del software.

En estos momentos nos encontramos abocados a las tareas conducentes a la creación de un grupo de investigación de software libre en el ámbito del citado Departamento. El principal objetivo que perseguirá este grupo será estudiar en profundidad el fenómeno del FLOSS y sus implicaciones en los aspectos filosóficos, políticos, legales y técnicos. A partir de este trabajo se propone interactuar con otras unidades académicas, grupos de investigación, grupos de usuarios y empresas, publicar estudios y artículos sobre el tema,

desarrollar y publicar una guía de autoestudio y, finalmente, volver a presentar nuestra propuesta de la asignatura electiva, fomentando su adopción tanto en esta Facultad Regional como en otras.

Por otra parte, la creación del citado grupo de investigación permitirá, a mediano plazo, generar entrenamiento para empresas y cursos de postgrado que podrán ser dictados en diplomaturas, maestrías y doctorados que requieran conocimientos en este campo de estudio.

Conclusión

El gobierno, la industria y la academia de la región están demandando mayores conocimientos en el ámbito del software libre y open source para sus profesionales e idóneos informáticos. Frente a esta situación creemos que es necesario que la formación de grado y la investigación en las universidades tome con interés esta disciplina de la ingeniería de software que se está volviendo angular y transversal a toda la industria.

Se propone la inclusión de una asignatura electiva en la carrera de Ingeniería en Sistemas de Información dedicada al estudio de los temas técnicos, legales y de negocios que son afectados por el software libre/de fuentes abiertas (FLOSS). Asimismo, se propone la creación de un grupo de investigación y desarrollo en el tema, de modo de crear un corpus de conocimiento que permita la generación de guías de autoestudio, entrenamiento y cursos sobre FLOSS.

Agradecimientos

A los alumnos y docentes consultados por darnos sus opiniones en una manera libre y franca. A la comunidad de software libre/de código abierto en general, por aportar herramientas y conocimiento para el desarrollo de nuestra industria y recordarnos que un gran poder conlleva una gran responsabilidad.

Referencias

- [1] Williams, Sam, *“Free as in Freedom: Richard Stallman's Crusade for Free Software”*, O'Reilly, 2011.
- [2] Stallman, Richard, *“The GNU Manifesto”*, Dr. Dobbs' Journal, 10 (3): 30, March 1985.
- [3] Dijkstra, Edsger W., *“Why numbering should start at zero (EWD 831)”*, E. W. Dijkstra Archive, University of Texas at Austin, August 1982.

- [4] Moody, Glyn, *“Rebel code: Linux and the open source revolution”*, Basic Books, 2009.
- [5] DiBona, Chris, Ockman, Sam, *“Open sources: Voices from the open source revolution”*, O'Reilly, 2008.
- [6] Weber, Steven, *“The Success of Open Source”*, Harvard University Press, 2004.
- [7] Fogel, Karl, *“Producing Open Source Software: How to Run a Successful Free Software Project”*, O'Reilly, 2005.
- [8] King, Peter, *“Global Tablet OS Market Share: Q2 2013”*, Strategy Analytics, July 2013.
- [9] Corbet, Jonathan, Kroah-Hartman, Greg, McPherson, Amanda, *“Linux Kernel Development: How Fast It is Going, Who is Doing It, What They are Doing and Who is Sponsoring It”*, The Linux Foundation, April 2012.
- [10] Pardini, Adrián, *“Fundamentación del uso de software libre en la universidad pública. Enseñando matemática con herramientas alternativas”*, I Jornadas de Enseñanza e Investigación Educativa en el campo de las Ciencias Exactas y Naturales, 2007.
- [11] Marzocchi, V. A., Cagnola, E. A., D'Amato, M. A., Vanzetti, N. A., Leonarduzzi, R., *“Las TICs en la Enseñanza de la Química: Una Experiencia con Software Libre de Visualización y Modelado Molecular”*, FABICIB, Volumen 14, Suplemento 1, 2010.
- [12] Aguilera, Sergio, Tommassone, Juan, *“Desarrollo cooperativo de software libre aplicado a la enseñanza de sistemas operativos sobre plataforma de cloud computing”*, XIII Workshop de Investigadores en Ciencias de la Computación, 2011.
- [13] Depetris, Beatriz, Feierherd, Guillermo Eugenio, *“Formación de usuarios inteligentes: uso combinado de software libre y propietario en una cátedra universitaria de Ciencias Económicas”*, II Congreso de Tecnología en Educación y Educación en Tecnología, 2012.
- [14] <http://www.ututo.org/>
- [15] <http://www.tuquito.org.ar/>
- [16] <http://lihuen.info.unlp.edu.ar/>
- [17] Cristiá, Maximiliano, Giusti, Gisela, Manzano, Felipe, *“The implementation of lisex, a mls linux prototype”*, Proceedings of ASSE (Argentine Symposium of Software Engineering), 2005.

[18] Babini, Dominique., Vergara-Rossi, Florencia, Medici, Flavia, González, Jessica, “*Biblioteca virtual cooperativa descentralizada con software libre Greenstone*”, El profesional de la información, 17(1), 64-68, 2008.

[19] Mariño, Sonia Itatí, Godoy, María V., Busso, Lorena E., “*Sistema de gestión de evaluaciones basado en software libre*”, XI Congreso Argentino de Ciencias de la Computación, 2005.

[20] Díaz, Francisco Javier, Banchoff Tzancoff, Claudia M., Amadeo, Ana Paola, Lanfranco, Einar Felipe, “*Utilizando herramientas de software libre para la gestión de cursos de grado*”, IV Congreso de Tecnología en Educación y Educación en Tecnología, 2012.

[21] Chiarani, Marcela Cristina, Pianucci, Irma Guadalupe, García, Berta Elena, Allendes Olave, Paola A., Torres, Vanesa, Ponce, Viviana, Leguizamón, Mario Guillermo, “*Recursos Educativos abiertos, gestión de contenidos digitales y Software de código libre*”, XIII Workshop de Investigadores en Ciencias de la Computación, 2011.

[22] Raymond, Eric, “*The Cathedral and the Bazaar: Musings on Linux and Open Source from an Accidental Revolutionary*”, O’Reilly, 1999.

[23] <http://www.gnu.org/licenses/license-list.html> (visitado 01/08/2013)

[24] Fundación Software Libre América Latina, “*Legislación argentina sobre software libre*”, <http://www.fsfla.org/ikiwiki/legis/argentina/index.es> (visitado 30/07/2013)

[25] Subsecretaría de Tecnologías de la Información, “*Software Libre: casos en el Gobierno*”, http://www.informatica.catamarca.gov.ar/software_libre.php (visitado por última vez el 30/07/2013)

[26] Gobierno Local, “*Banco de Recursos: Ordenanzas, Decretos y Leyes relacionados con el uso y la promoción del Software Libre a nivel Municipal y Provincial*”, http://www.gobiernolocal.gob.ar/index.php?option=com_flexicontent&view=category&Itemid=14&cid=34&idColeccion=161079 (visitado por última vez el 30/07/2013)

[27] PyAr - Python Argentina, www.python.org.ar/ (visitado 30/07/2013)

[28] SciPyAr, www.python.org.ar/SciPyar (visitado por última vez el 30/07/2013)

[29] ArPug, <http://www.arpug.com.ar/trac> (visitado por última vez el 30/07/2013)

[30] web2py, www.web2py.com (visitado por última vez el 30/07/2013)

[31] Pilas engine, <http://pilas-engine.com.ar/> (visitado por última vez el 30/07/2013)

[32] Cabral, Juan, “*Software libre para ingenieros de sistemas*”, Congreso Nacional de Estudiantes de Sistemas de Información - CNEISI 2011, Embalse (Córdoba), 2011.

[33] Zanotti, Agustín, “*Comunidades de software libre en Argentina: Algunas exploraciones y vectores de análisis*”, Global Movements, National Grievances, 2012.

[34] Robert, Verónica, “*Límites y efectos de la difusión de software libre en un país en desarrollo. El caso de la Argentina*”, Yoguel et al., *La informática en la Argentina. Un desafío a los problemas de especialización y competitividad*, Buenos Aires, UNGS-Prometeo, 2006.

[35] Cámara Argentina de Empresas de Software Libre, www.cadesol.org.ar/ (visitado por última vez el 30/07/2013)

[36] Capítulo de Software Libre de CAMTIC - Cámara de Tecnologías de Información y Comunicación (Costa Rica), <http://www.camtic.org/sobre-camtic/capitulos/software-libre/> (visitado por última vez el 30/07/2013)

[37] Presidencia de la República Oriental del Uruguay, “*Comunicado Conjunto de los Presidentes de los Estados partes del MERCOSUR*”, <http://www.telam.com.ar/advf/documentos/2013/07/51e42f6acb4c9.pdf> (visitado 30/07/2013)

[38] Chavez, Christina, et al. "Free/Libre/Open Source Software Development in Software Engineering Education: Opportunities and Experiences." http://fees.inf.puc-rio.br/FEESArtigos/artigos/artigos_FEES11/fees11_02.pdf

[39] Kamthan, Pankaj. "On the prospects and concerns of integrating Open Source Software environment in software engineering education." *Journal of Information Technology Education: Research* 6.1 (2007): 45-64. <http://www.jite.org/documents/Vol8/JITEv8p229-242Long694.pdf>

[40] Long, Ju. "Open Source Software Development Experiences on the Students' Resumes: Do They Count?-Insights from the Employers' Perspectives." *Journal of Information Technology Education: Research* 8.1 (2009): 229-242. <http://www.jite.org/documents/Vol8/JITEv8p229-242Long694.pdf>

[41] Universidad Abierta de Cataluña. Materia “*Ingeniería del software en entornos del software libre*”. Maestría en Software Libre. Barcelona, España. Febrero 2007. <http://ocw.uoc.edu/informatica-tecnologia-y-multimedia/ingenieria-del-software-en-entornos-del-software-libre>

[42] Fabio Kon. “Desenvolvimento de Software Livre”. Centro de Competencias en Software Libre. Instituto de Matemática y Estadística. Universidad de San Pablo (2009-2013) <http://ccsl.ime.usp.br/wiki/MAC5856> (visitado por última vez el 30/06/2014)

[43] Ministerio de Educación. Resolución Ministerial 786/09: “Aprobar los contenidos curriculares básicos, la carga horaria mínima, los criterios de intensidad de la formación práctica y los estándares para las carreras de Licenciatura en Sistemas -Sistemas de Información-, Análisis de Sistemas, Licenciatura en Informática”. Buenos Aires, 26 de Mayo de 2009 <http://repositorio.educacion.gov.ar:8080/dspace/handle/123456789/7870>

[44] Reingart, Mariano. "Materia Práctica Profesional" (programa). Instituto Superior Tecnológico Blaise Pascal. Provincia de Buenos Aires. DIPREGEP N° 6131. <http://reingart.blogspot.com.ar/p/materia-practica-profesional.html>

[45] 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)

[46] Wachenchauzer, Rosita. “Trabajos de Carreras de Informática en Comunidades de Código Abierto”. Anales de las Jornadas Argentinas de Software Libre 2013. ISSN: 1850-2857 (130-140) <http://42jaiio.sadio.org.ar/proceedings/simposios/Trabajos/JSL/12.pdf>

Datos de Contacto:

Juan Bautista Cabral. Laboratorio de Investigación de Software – Universidad Tecnológica Nacional, Facultad Regional Córdoba. Ayacucho 437 ID, Córdoba, Córdoba, Argentina - jbc.develop@gmail.com