

La Tecnicatura Universitaria en Software Libre de la Universidad Nacional del Litoral: fundamentos, lineamientos curriculares y desafíos de su implementación.

Martín Bayo¹, Oscar Vallejos²

¹ **Cátedra informática, Universidad Nacional del Litoral**

{martin.bayo}@gmail.com, ² **Cátedra Ciencia, Tecnología y Sociedad, Universidad Nacional del Litoral** {oscarrvallejos}@gmail.com

Resumen. Este trabajo presenta un análisis de las condiciones y fundamentos que dieron lugar a la Tecnicatura Universitaria en Software Libre de la Universidad Nacional del Litoral. Se presentan las tres dimensiones que se tuvieron en cuenta para el diseño curricular. Se analizan y exponen los principales lineamientos curriculares del Plan de Estudios. Por último se plantean los principales desafíos que implica su implementación: una didáctica del software libre y el grado de coherencia y adecuación de la modalidad de trabajo que propone la carrera en relación con los valores y las prácticas propias del Software y la Cultura Libre.

Palabras Clave: Tecnicatura Universitaria de Software Libre, Fundamentos del Plan de Estudios, Diseño curricular, Didáctica del Software Libre

1 Introducción

El software libre (SL) y la Cultura Libre (CL) se convirtieron en dos grandes movimientos sociales nucleados en torno a los principales problemas emergentes en el contexto de un capitalismo cognitivo (Boutang, 2008). El avance de la controversia en torno de la naturaleza del software en particular, y de los productos culturales, en general, abrió un amplio debate acerca del papel de las universidades y de las agencias estatales de ciencia y tecnología en materia de la apropiación y circulación del conocimiento socialmente valorado. Las universidades, específicamente, tienen un papel ambivalente en este proceso. Los estudios históricos del surgimiento del movimiento del software libre plantean que las universidades atravesaron tres momentos en relación con el software y la cultura libre. Un primer momento en el que emerge una cultura del software libre en el contexto de prácticas académicas

organizadas en torno de un ethos de producción de conocimiento científico-tecnológico vinculado con la circulación de las mejoras de los softwares y de modos de producir ajustes en los artefactos - algo próximo a la idea de comunalidad del ethos científico identificado por Merton - y el de una frontera difusa entre la vida académica y la vida sin más. Un segundo momento que se corresponde con el giro de las universidades hacia el mercado y el surgimiento de fuertes vínculos con las empresas, los contratos de confidencialidad, los procesos de patentamiento y la emergencia de nuevas subjetividades académicas en un proceso creciente de mercantilización de las relaciones sociales [1]. Este segundo momento produce la convergencia de este "fermento" universitario con los movimientos sociales libertarios que ponen en discusión los valores que organizan el mundo capitalista [2]. Un tercer momento de reingreso y de reemergencia de la cultura libre y abierta a partir de una nueva era en la percepción del potencial del software libre, de las demandas estudiantiles y de ciertos grupos sociales por una educación diferente. Este reclamos de educación diferente tienen dos sentidos; uno que está en consonancia con una de las libertades del software libre: libertad para estudiar y modificar - para lo que se necesita acceso al código. El otro en el sentido de una educación menos comprometida con los valores del mundo capitalista. Este tercer momento está fuertemente vinculado con los procesos de tensión que emergen en el capitalismo cognitivo en el cual las disputas sobre los derechos de autor y el acceso a los productos intelectuales son centrales.

Estos tres momentos tienen un despliegue temporal y espacial no lineal; un indicador de esta situación es la coexistencia en un espacio de rivalidad o de enfrentamiento entre lo que podemos llamar estilos socio-cognitivos del software. Un estilo socio-cognitivo asociado al software privativo, con un modelo pedagógico implícito (el modelo de caja negra) y una orientación hacia el mercado. Un estilo socio-cognitivo asociado al software libre con un modelo pedagógico implícito - el modelo de mejoras colaborativas constantes a partir del acceso al código y de los procesos de documentación - y con orientaciones plurales que no necesariamente rivalizan con el mercado.

Los Laboratorios de Informática de las universidades suelen ser el espacio donde las disputas entre estos dos estilos se manifiestan. La idea de introducir lo que se llama Live CD fue una respuesta creativa de docentes y estudiantes enmarcados en el software libre a esta situación y poder o bien enseñar o analizar las características y funcionalidades del software o de trabajar en un ambiente de software libre sin usar el software privativo - una cuestión ética. En general, en los Laboratorios de Informática hay pocas máquinas con software libre instalado puesto que la mayoría tiene software privativo. Además, como se sabe, en los últimos años las grandes empresas productoras de software privativo han modificado su estrategia hacia las

universidades produciendo modos de licenciamiento para que docentes y estudiantes puedan “acceder” al software sin pagar licencias. El personal de planta que lleva adelante el mantenimiento del parque de máquinas no suele contar con conocimientos requeridos para que funcionen con software libre puesto que fueron entrenados en el modelo pedagógico del software privativo.

Este momento de reingreso y reemergencia del software libre y de la cultura libre en las universidades se impulsa desde campañas globales como la que lleva a cabo el Ranking de universidades en Software Libre (RuSL) [3]. La visibilización del papel de las universidades en España y América Latina en materia de Software Libre es parte de una estrategia global de apoyo. Este ranking es importante puesto que produce informes sobre los aspectos que hacen a la investigación, la colaboración y la transferencia de conocimientos asociados al Software Libre. La Universidad Nacional de La Plata ocupa un lugar importante dentro del ranking. En cuanto a la Universidad Nacional del Litoral falta realizar un estudio profundo acerca de su contribución al Software Libre y sobre todo, a aquello que hace a la cultura en software libre.

La Tecnicatura Universitaria en Software Libre surge en este contexto general con el reconocimiento de que se requiere una formación específica tanto de las condiciones en las que se desarrolla el software y cómo se organiza socialmente su utilidad. En este sentido, el Plan de Estudio está diseñado para que sus estudiantes puedan entender y participar en los espacios del Software y la Cultura Libre.

2 Propuesta Curricular

Los lineamientos curriculares que organizan la TUSL de la Universidad Nacional del Litoral son los siguientes:

1. Una mirada socio-tecnica
2. Una mirada vinculada a los desempeños comunitarios
3. Una mirada vinculada a las oportunidades laborales de los egresados

2.1. Fundamentos

1. El enfoque socio-técnico es una concepción de la tecnología y del software que reconoce las condiciones en las que los artefactos técnicos adquieren existencia, significados y se difunden.

La cuestión de los significados está asociada a lo que se reconoce en las comunidades de Software Libre como "filosofía del Software Libre". Esta filosofía pone en evidencia el tipo de finalidades con las que están vinculados quienes desarrollan o trabajan con un sistema informático. La sistematización general de las cuatro libertades expresa un ethos identificativo y una orientación programática en términos de libertades asociadas a ámbitos específicos: para ejecutar el programa; para estudiar y modificar su código; para redistribuir copias exactas; y para distribuir versiones modificadas. Este ethos orientó la creación de software libre para producir y compartir el conocimiento que era necesario para su propia existencia y circulación. Es decir, las motivaciones internas de las comunidades de SL impulsaron la producción y la circulación de software para desarrollarse y ampliarse. Las tecnologías wikis en el caso de wikipedia o wikileaks o los repositorios públicos como github fueron diseñados y desarrollados para permitir formas de trabajo e interacción de colaboración distales, asincrónicas, distribuidas y no jerárquicas. Las condiciones y las tecnologías colaborativas desarrolladas por las comunidades de Software Libre permiten que los dispositivos funcionen asociados a demandas e intereses de los usuarios. Sin embargo, las propias comunidades reconocen que el grado de avance en el uso del software libre parece depender de una doble estructuración: una distribución social amplia y una configuración especializada. La codificación de los conocimientos que producen las universidades juegan en este sentido un papel muy importante; sin embargo, como plantea Margarita Padilla [4], la posibilidad de que el conocimiento esté disponible para los usuarios requiere de pensar formas de sostener económicamente a quienes pueden de hacer de hacer de transmisores de ese saber. Como plantea McGaw [5], las tecnologías funcionan en base a un amplio proceso de adaptaciones o ajustes; en las tecnologías informáticas esas adaptaciones requieren no sólo acceso al código fuente de los softwares sino del montaje de un amplio sistema de conversación, de consultas y de compartir las experiencias de éxitos y de fracasos.

2. La mirada vinculada a los desempeños tecno-sociales propios del software libre es la que permite articular la propuesta pedagógica a lo que los miembros de la comunidad reconocen como "saberes propios de sus miembros". Las comunidades de Software libre trabajan en proyectos de manera colaborativa, comparten sus conocimientos, aprenden a gestionar la información y organizan comunidades de

práctica [6]. Estos diferentes desempeños, a su vez, tienen distintas estructuras, desde más verticales como la estructura social del desarrollo del kernel Linux, pasando por estructuras democráticas como el proyecto Debian; funcionamiento más horizontales como los diferentes grupos de usuarios de GNU/Linux entre los que situamos al LugLi, un grupo de usuario referente de la región.

Estos desempeños se ven también reflejados en los emprendimientos económicos que se articulan en formas cooperativas como las nucleadas en la Federación Argentina de Cooperativas de Trabajo de Tecnología, Innovación y Conocimiento (FACTTIC) y en empresas de servicios y producción de software como las que hemos estado observando de manera informal en la región, estas son Netmanagers y Linux [7].

Uno de los desempeños más importantes es el espíritu colaborativo que es uno de los valores fuertes de las comunidades de SL. En cierto modo, este desempeño fue incorporándose a las empresas como Google. Los modelos de economía social y solidaria también incorporan fuertemente este valor.

3. La mirada vinculada a las oportunidades laborales de los egresados está pensada a partir de la ampliación del uso del SL en el Estado, el surgimiento de empresas de informáticas vinculadas con el SL y el Programa Conectar Igualdad que plantea la incorporación de las TIC con un fuerte anclaje en el SL en las instituciones escolares de todo el país. Al pensar en estas oportunidades laborales que se presentan hoy pero que son muy amplias se dio lugar a las diferentes orientaciones que ofrece la TUSL.

Como se analiza en Bayo, Morales y Vallejos (2015), la incorporación del software libre en el Estado es un proceso creciente que ofrecerá oportunidades laborales fundamentales para los futuros egresados en la medida en que las burocracias estatales no sólo solicitan conocimientos funcionales sino también certificaciones. Específicamente la Ley de Software Libre de la Provincia de Santa Fe plantea la obligatoriedad del uso del software libre en todos los ámbitos del Estado y eso sólo es posible sobre la disponibilidad de personal cualificado que se incorpore a la esfera estatal. Los funcionarios de la Secretaría de Tecnologías para la Gestión dependiente del Ministerio de Gobierno y Reforma del Estado manifestaron en una reunión mantenida por la Reglamentación del Software Libre que la incorporación de masa crítica era uno de los puntos centrales de la implementación de la Ley.

El programa Conectar Igualdad comenzó a partir del año 2010, a destinar distintos tipos de recursos para fomentar el uso de tecnologías dentro de las aulas. Entre los recursos puestos a disposición de las escuelas se encuentran, por un lado, las netbooks distribuidas a alumnos y docentes, y por otro los recursos necesarios para asegurar la conectividad y el uso de este equipamiento: servidores para disponer de

servicio centralizado de internet, servicio de "desbloqueo de las netbooks", dispositivos de red, etcétera. Para esto se requiere también de personal capacitado en estas tecnologías en particular, y cuyas funciones no necesariamente apuntan a la colaboración con el docente. Es de destacar que las computadoras que entrega el programa Conectar Igualdad están equipadas con distribuciones de Linux y paquetes libres orientados a su uso en el aula. Es en este campo donde la TUSL tiene el papel de fortalecer el rol del técnico, colaborando con la formación de profesionales que estén en condiciones de asistir a los docentes para utilizar toda la potencialidad de las herramientas libres dentro del aula.

En base a estas consideraciones la Tecnicatura Universitaria en Software Libre plantea tres orientaciones: desarrollo, administración y software educativo.

2. 2. Estructura del Plan de Estudios:

La estructura curricular de la Tecnicatura Universitaria en Software Libre consta de dos ciclos: el primero está formado por diez asignaturas y el segundo está formado por cinco asignaturas. Además contempla un espacio para el Proyecto Final de carrera. El primer ciclo es común y el segundo ciclo está organizado en tres trayectos formativos diferenciados: Administración, Desarrollo y Educación. El estudiantado deberá elegir uno de estos trayectos. Cada trayecto tiene cuatro asignaturas fijas y una optativa. La asignatura optativa es parte de los otros trayectos formativos que conforman la carrera. El Proyecto Final está asociado al trayecto de formación elegido por el estudiantado.

Los trayectos formativos programados para el segundo ciclo ofrecerán cuatro materias que destinadas a brindar los conocimientos, competencias y habilidades vinculados con cada desempeño: administración, desarrollo y software educativo. La realización de una quinta asignatura (optativa cruzada), perteneciente a alguno de los otros dos trayectos formativos que no hayan sido elegidos por el estudiantado permite un acercamiento a conocimientos vinculados a los otros desempeños y a ampliar los marcos de referencia desde los cuales se piensa el objeto de estudio.

La realización de un Proyecto Final con el propósito de integrar los conocimientos y que el estudiantado pueda tener una experiencia próxima al desempeño profesional que sea guiado por un equipo docente.

El primer ciclo organiza el conocimiento básico para poder participar dentro de las comunidades de Software Libre; se enseñan los conceptos y los procedimientos

básicos requeridos para dominar técnicamente sistemas de GNU/Linux y comprender las tecnologías utilizadas en diferentes desarrollos y entornos de Software Libre.

En el trayecto formativo correspondiente al de Administración de Software Libre ofrece los conocimientos específicos para que un técnico entienda las infraestructuras actuales que existen tanto en ambientes privados como también poder tener una visión de migración hacia SL en ambientes públicos, teniendo en cuenta, en ambos casos, los aspectos vinculados a la seguridad de la información. (distintos aspectos a lo que hace la seguridad de la información).

En el trayecto formativo correspondiente al de Desarrollo en Software Libre ofrece contenidos más vinculados a la programación WEB y a las metodologías de trabajo en desarrollos colaborativos. También se incorporan contenidos referidos a la seguridad priorizando la seguridad de la información de los usuarios

El trayecto formativo relativo a la educación ofrece conocimiento sobre los procesos educativos y la gestión de los softwares de manera que las y los graduados puedan colaborar con docentes de distintas áreas en todos los niveles de enseñanza en cuanto a la utilización de soportes informáticos.

2.3. Contenidos mínimos

Primer Ciclo

Introducción al Software Libre: Introducción al software libre: Definición de la libertad en el software. Tipos de software según las libertades que ofrece. Modelos y procesos de desarrollo de software libre. Entornos y tecnologías de desarrollo de SL. Casos de estudio de desarrollo de SL

Software libre: Ofimática en las organizaciones: Utilización de un live cd. Distribuciones de GNU/Linux. Entornos gráficos de escritorio. Modificación y personalización, plugins. Instalación y configuración de escritorio y gestión de paquetes en forma gráfica

Introducción al Desarrollo de Software: Conceptos básicos (variables, declaraciones, tipos de datos, operadores, expresiones lógicas, estructuras de control, estructuras repetitivas, procedimientos y funciones) para la programación. Introducción a la programación. Programación estructurada, modular y orientada a objetos, tipos de lenguajes

Software Libre y Sociedad: Caracterizaciones de la sociedad actual: sociedad de la información, sociedad del conocimiento y capitalismo cognitivo. La ciencia y la tecnología en las sociedades periféricas. El surgimiento del SL y su difusión en el mundo. Los valores promovidos por la comunidad de SL. Aspectos legales y políticos del SL. El software libre como movimiento social. El Estado y las administraciones públicas y el SL.

Redes: Principios de redes de computadoras, usando el Modelo de Referencia OSI como marco. Arquitecturas de red. Redes de LAN, MAN, WAN y distintos protocolos. Configuración de red en Linux.

Administración GNU / Linux I: Comandos básicos. Gestión de procesos. Administración de usuarios y grupos. Sistema de archivos, montado manual y automático. Gestión de Software. Comandos de filtros. Configuración de impresoras y sistemas de archivos compartidos. Introducción a la programación shell.

Comunidades de Software Libre: La estructura de las comunidades de SL y la organización de las diferentes distribuciones. La organización de proyectos de desarrollo de SL. Trabajos de control de versión. La creación de repositorios: sourceforge. Las articulaciones de las comunidades de SL con otros movimientos sociales.

Base de Datos: Introducción a bases de datos, tipos de modelos de bases de datos. Lenguaje SQL. Introducción al diseño de base de datos. Instalación, configuración, conexión y características de motores de bases de datos libres.

Tecnologías WEB: Infraestructura de servicios web: instalación y configuración. Lenguaje HTML, conexión con Base de Datos. Otros entornos de lenguajes de última generación. Monitorización y análisis.

Culturas digitales libres: Software libre y la libre difusión del conocimiento. Contenidos libres y las formas de protección del conocimiento público. Las licencias y el movimiento copyleft, creative common. Construcción cooperativa de conocimientos y herramientas colaborativas. Los procesos creativos y el diseño de herramientas en entornos libres. Narrativas transmediales.

Segundo Ciclo

Trayecto Administración

Administración GNU / Linux II: Arquitectura del sistema operativo GNU/Linux. Arranque del sistema. Acceso remoto. Sistemas de logs. Backups definición y tipos. Programación de tareas sincrónicas y asíncronas. Compilación y personalización del kernel y de módulos externos.

Políticas de Migración y Seguridad: Políticas de migración hacia el SL. Migración en el Estado. Los roles y la conformación del equipo de instrumentación. La conformación de un programa de formación y capacitación de los agentes públicos. Seguridad en Redes. Exploración de puertos. Conceptos básicos de criptografía. Mecanismos de Detección, protección y prevención.

Administración GNU / Linux III: Sistemas de almacenamiento de datos (RAID y LVM). Configuración y políticas de Firewall. Redes privadas virtuales. Monitorización de servicios y recursos. Virtualización. Administración centralizada.

Proyectos para la administración del estado con SL: Los procesos de análisis de demanda de software. Procesos de selección de SL. Viabilidad funcional, operativa y técnica. Toma de decisiones sobre las áreas estratégicas de desarrollo de SL. El caso de los SIG.

Trayecto Programación

Estándares para desarrollo WEB: Desarrollo de aplicaciones WEB bajo Unix. Frameworks y librerías. La W3C. Principios. Intercambio de datos en formatos estándar: HTML y XML. Recomendaciones y lineamientos. CSS. DOM (Document Object Model). Introducción a Javascript.

Programación I: Arquitectura Cliente-Servidor. Arquitectura de programación en tres capas. Lenguaje de marcado HTML. Programación en PHP. Implementación de aplicaciones web con Apache+PHP. MVC. Desarrollo de aplicaciones utilizando el framework.

Programación II: Gestión de transacciones sobre base de datos. Integración utilizando el framework. Casos de estudio. Desarrollo de aplicaciones completas. Desarrollo de software orientado a objetos con UML.

Seguridad en el desarrollo de software: Desarrollo de aplicaciones seguras: riesgos inherentes al entorno web, diferentes tipos de ataques (DOS, inyección de código, etc.). Buenas prácticas en seguridad (gestión de permisos y privilegios, almacenamiento y manipulación de contraseñas, etc.). Normativas para el manejo de datos (habeas data). Licencias de software: consideraciones éticas y legales en el uso de bibliotecas y software de terceros, y en la publicación de software propio.

Trayecto Educación

Procesos educativos y Software Libre: Elementos de análisis de los procesos de enseñanza y aprendizaje. Procesos de selección de software educativo: problemas y contexto. Integración curricular del SL: de la enseñanza y aprendizaje del software libre a la enseñanza y aprendizaje con SL. Gamificación de la educación. Comunidades de aprendizaje. Integración de dispositivos móviles a las instituciones educativas.

Diseño Digital Educativo: Diseño de contenidos educativos con SL. Diseño de objetos de aprendizaje: descripción, creación y distribución. Naturaleza de los textos en la Web2.0. Tecnologías libres para educación: animaciones interactivas, actividades de resolución de consignas, mapas mentales multimedia, etc.

Entornos de aprendizaje: Plataformas de contenidos libres. Modelos de aulas virtuales. Metodologías de aprendizaje a través de EVA. Trabajo colaborativo y cooperativo. Trabajo en RED. Repositorios Educativos..

Alfabetización digital: Políticas públicas de alfabetización digital y SL. El software libre en las instituciones educativas: políticas de migración y de formación de la comunidad educativa. Estudio de casos: el Proyecto ALBA, Plan Ceibal, Conectar Igualdad.

Trabajo Final: El proyecto final podrá estar orientado a diseñar intervenciones en los diferentes ámbitos de alcance de la carrera. Elementos de producción y gestión de proyectos. Identificación de los ámbitos de intervención del proyecto. Determinación del rol del diseñador y de los actores del ámbito de intervención. Las etapas del proyecto y el establecimiento de objetivos y logros. Producción del proyecto. Etapas y fases de implementación. Pasaje del Proyecto al Producto. Evaluación de los resultados.

3. Desafíos de la implementación de la Tecnicatura Universitaria en Software Libre.

Los desafíos están vinculados con la organización de una carrera que esté en consonancia con la cultura propia del Software y la Cultura Libre. Ello implica, al menos las siguientes dimensiones:

3.1 Cátedras colaborativas

El trabajo docente en la TUSL no sólo se basa en transmitir conocimiento, sino también articularse de forma coherente la filosofía del Software y la Cultura Libre. Ello implica producir, adaptar y reorganizar los materiales de estudio de manera colaborativa y publicarlos con alguna licencia reconocida por las comunidades de Software y de Cultura libre. A su vez, la construcción de las primeras materias también se trabajará de manera colaborativa; en este sentido, las primeras asignaturas ya están siendo trabajadas de este modo. La determinación de los contenidos definitivos - tanto la estructura y la didáctica de los materiales - se discuten entre el grupo de docentes de la Tecnicatura. Se considera que sólo es posible transmitir el valor del trabajo y el aprendizaje colaborativo si la misma fue parte de la experiencia de los docentes de la carrera.

3.2 Enseñanza

La Tecnicatura Universitaria en Software Libre se impartirá a distancia; este otro gran desafío. Los docentes deben ser capaces de convertir las aulas virtuales en verdaderos espacios de circulación y producción colaborativa de conocimientos. Esto implica hacer del entorno de aprendizaje que la Universidad Nacional del Litoral tiene un espacio de experimentación. La plataforma a utilizar para el dictado de las clases es Moodle, un software con licencia libre que permite la incorporación de distintos recursos para el trabajo colaborativo como wikis, foros y otros recursos. También brinda la posibilidad de incorporar servicios de videoconferencia, anexas recursos externos como videos y documentaciones varias.

También se usarán otras tecnologías como etherpad, un sistema de creación colaborativa de tiempo real tanto de documentos como de desarrollos, con la posibilidad de que los participantes interactúen mediante el chat y poder evaluar los distintos aportes de los alumnos de manera individual y grupal, servicios de repositorios públicos como gitlab u otras que logren comunicar y realizar producciones comunes.

Otra característica de este tipo de enseñanza es la resolución de problemas prácticos para poder comprender la naturaleza de los conceptos que intervienen en el mismo, la otra característica que se pueden observar de los llamados autodidactas del SL es que generalmente analizan conceptos y saberes a través la resolución de problemas reales, no sólo conociendo los conceptos teóricos de los mismos sino también analizando el funcionamiento práctico de artefacto.

La dimensión de la enseñanza implica también pensar nuevas modalidades de evaluación. A las formas individuales propias de la universidad hay que reemplazarlas por formas colaborativas. A los trabajos que tienen un cierre sobre sí mismos y que no circulan, trabajos que se articulen con los proyectos en marcha y puedan circular.

En términos generales, la Tecnicatura Universitaria en Software Libre requiere de la conformación de una Didáctica del Software Libre.

3.3 Articulación con las comunidades

Es muy importante y valorable que estudiantes y docentes participen individual o institucionalmente en los distintos eventos que las comunidades de Software Libre propone. La participación en las comunidades es una exigencia en tanto no hay desempeños aislados, lo que se considera conocimiento relevante depende de estos intercambios con los demás miembros de las comunidades.

Uno de los mayores desafíos es la de la conformación de una cultura de software libre vinculada con la carrera donde los procesos de circulación de conocimiento puedan trascender las fronteras de la universidad y distribuirse socialmente.

Referencias

1. Cf. Fraser, N.: (1997) *Iustitia interrupta. Reflexiones críticas desde la posición 'postsocialista'*. Traducción de Magdalena Holguín e Isabel Jarmillo. Universidad de los Andes. 1997.
2. Cf. Castells, M.: (1997) *La era de la información. Economía, sociedad y cultura. Vol. II: El poder de la identidad*. México. Siglo XXI. Traducción de Carmen Martino. 1999.
3. <http://www.portalprogramas.com/software-libre/ranking-universidades/>
4. <https://vimeo.com/30812111>
5. McGaw, J.: (1996) "Reconceiving Technology: Why Feminine Technologies Matter!". En Rita Wright: *Gender and Archaeology*. University of Pennsylvania Press, pp. 52-75.
6. Wenger, E., McDermott, R. y Snyder, W.: (2002) *Cultivating communities of practice : a guide to managing knowledge*. Boston. Harvard University Press.
7. <http://libroblanco.org.ar/index2.php?provincia=santafe>