

EXPERIENCIAS DE USO DE SOFTWARE LIBRE Y FREEWARE PARA LA REPRESENTACIÓN GRÁFICA DE INGENIERÍA

Rubén Darío Morelli, Universidad Nacional de Rosario, rdm@fceia.unr.edu.ar

Hernán Alfredo Pangia Ctenas, Universidad Nacional de Rosario, pangia@fceia.unr.edu.ar

Luis Sebastián Nieva, Universidad Nacional de Rosario, 5e6anieva@gmail.com

Resumen— Luego de cuatro años de investigación sobre las posibilidades de uso de Software Libre en el área de los Sistemas de Representación Gráfica para las carreras de Ingeniería de nuestra Universidad, hemos consolidado una experiencia que nos permite afirmar y demostrar a través de la práctica que existen poderosas aplicaciones de diseño desarrolladas con Software Libre o Freeware que permiten al docente, al estudiante, a las instituciones públicas o al profesional, trabajar dentro del marco legal y ético que deviene de utilizar software libre y gratuito. Pero más allá de la importancia de la legalidad de las licencias, el Software Libre conlleva en sí mismo un paradigma o filosofía diferente a lo que la mayoría de nosotros, que venimos del mundo del software privativo, estamos acostumbrados. En este trabajo formulamos en primer lugar el marco conceptual y estado actual del tema de nuestra investigación. Luego, con la intención de compartir y sumar conocimiento, presentamos una amplitud de temáticas referidas a nuestra disciplina en base a la potencialidad del trabajo con software libre: un trabajo de diseño de ingeniería, contenidos para el aprendizaje de la asignatura Representación Gráfica y un vídeo didáctico.

Palabras clave— *Software Libre, CAD, Freeware, Enseñanza, Representación Gráfica.*

1. Introducción, marco conceptual y estado actual del tema.

Desde el año 2012 en que comenzamos nuestro proyecto de investigación sobre Software Libre y gratuito para la enseñanza de la Representación Gráfica tenemos el objetivo central de buscar alternativas a los programas CAD tradicionales ya que el valor de sus licencias estaba (y sigue estando) fuera del alcance de las instituciones, alumnos y docentes. El Software Libre permite trabajar dentro del marco legal en cuanto a licenciamiento, hecho que consideramos fundamental y además es lo que corresponde. Y como docentes universitarios es nuestro deber orientar a nuestros alumnos en el sentido ético de la legalidad y la libertad. Tratamos de generar conciencia y en 2012 ya decíamos: “consideramos como principio rector que como Universidad Pública debemos sostener el libre acceso a la información y al conocimiento, y evitar caer en una dependencia tecnológica a la hora de utilizar software dedicado a la enseñanza” [1]. Es que cuando en una institución de enseñanza como lo es una Facultad se trabaja con un software propietario específico, por ejemplo AutoCAD, se crea una cierta dependencia tecnológica en favor de ese software y de esa marca comercial, ya que docentes y alumnos terminan acostumbrándose a esa única opción y es lógico pensar que el joven profesional egresado preferirá seguir trabajando con el único software de diseño que aprendió. Además, el problema del costo de la licencia legal volverá a aparecer al egreso de la Universidad cuando el joven profesional necesite equiparse de software para ejercer su profesión, y estará siempre el riesgo de recurrir a la copia ilegal de programas. Sin embargo,

muchos aún suelen considerar que el uso de copias ilegales de un programa es algo inevitable para poder estar en la cresta de la ola del mundo digital con el mejor software. En este sentido, se debe tener en cuenta que en cualquier país cuando la licencia de un software no permite su copia y libre distribución, hacerlo es violar la ley. Al respecto coincidimos con lo expresado por Brocca y Casamiquela [2] cuando dicen que “se puede percibir un desconocimiento generalizado entre los usuarios de computadoras acerca de que la utilización, copia o reproducción de productos de software no siempre es un acto lícito. Si estas acciones se llevan a cabo sin respetar los términos establecidos en la licencia - o sin contar con ella - se está cometiendo un delito englobado dentro de lo que se ha dado en denominar piratería informática”.

A medida que fuimos progresando en la tarea de investigación comprobamos que el Software Libre va mucho más allá de resolver el problema económico y legal relativo a las licencias, ya que su filosofía se inscribe en un nuevo paradigma ético y social dentro de la sociedad del conocimiento, con su propio marco político y legal. Un nuevo paradigma que se basa en premisas como la solidaridad, la cooperación y la libertad. Tal como lo definió Stallman [3], “con software libre nos referimos a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software”. Un software para ser libre debe cumplir con todas estas condiciones sin excepción. Software Libre no significa que necesariamente deba ser gratuito. El término Libre debe entenderse en el sentido de *libertad*, como lo define Stallman. Puede tener un costo pero una vez adquirido el software uno tiene el derecho y la libertad de hacer lo que quiera con él y, si se reescribe el código, el software debe seguir siendo libre en sus nuevas versiones mejoradas o modificadas.

Hoy podemos hablar de que el movimiento de Software Libre ha crecido hacia nuevos conceptos: el de Hardware Libre [4] y el de Cultura Libre [5]. Hardware libre son aquellos dispositivos con controladores de código abierto, con especificaciones y diagramas de libre acceso, gratis o con algún costo y con la misma filosofía del Software Libre. La Cultura Libre va en el mismo sentido aplicado al principio del “contenido libre: libertad para compartir, distribuir o modificar trabajos y obras creativas”, y es lo contrario a las restricciones de los derechos de autor. Las licencias de obras del tipo Creative Commons son un ejemplo.

Un referente internacional actual del movimiento del Software Libre y sus nuevos ecosistemas como el Hardware Libre y la Cultura Libre es el Dr. en Robótica e Ing. Juan González [6] conocido como Obijuan, quien afirma:

“El conocimiento de la tecnología tiene que estar disponible para todos nosotros. Ha llegado el momento de redefinir: ¿por qué no creamos entre todos un Patrimonio Tecnológico de la Humanidad? Un compendio de la información tecnológica a la que todos tengamos acceso, que tengamos la libertad de estudiar, de poderla entender y construir objetos a partir de esa información. Esto que parece una utopía ya ha empezado a ocurrir...a partir de 1980 con Richard Stallman y el Software Libre”.

Obijuan se autodefine como un “Maker”. Los Makers son personas apasionadas por construir sus ideas como objetos físicos y así experimentar, ensayar y aprender. También se aplica a construir lo que otros han diseñado para comprender y de esa forma asimilar el conocimiento.

“El máximo exponente del Hardware Libre es el proyecto Arduino, que ha conseguido llevar esta idea a todas partes del mundo. Pero el hito más importante sin duda se produjo en el año 2008, cuando un investigador inglés, Adrian Bowyer, el fundador del proyecto Rep-Rap, construyó y terminó la primera máquina que podemos considerar como la primera de nuestro Patrimonio Tecnológico de la Humanidad. Una máquina que era una impresora 3D que está disponible absolutamente toda la

información; hasta el último detalle de todo lo podemos estudiar: las tuercas, el Firmware, el Software, la Electrónica, absolutamente todo. ¿Y qué fue lo que ocurrió?, pues que miles de entusiastas Makers comenzamos a estudiarla y empezamos a replicarlas, y se crearon cientos de modelos diferentes de esta máquina. Diferentes versiones. Las impresoras 3D son esas maravillosas máquinas, algo mágico que nos permiten crear objetos de la nada” [6].

En el gráfico de la Figura 1 se representa el pensamiento de Obijuan: crear e incrementar un Patrimonio Tecnológico de la Humanidad ubicando el Software Libre como centro y ensamblando los distintos ecosistemas a partir del GNU-Linux (o simplemente sistema operativo Linux), el Hardware Libre, el proyecto Rep-Rap (Replicating Rapid prototyper. Tecnología Open Source que permitió que una máquina impresora 3D sea auto-replicable mediante su propia impresión 3D), los Printbots (robots imprimibles con impresoras 3D a partir de difundir la información para que puedan ser copiados y redefinidos en nuevas adaptaciones) y desembocando en nuevos proyectos. Coincidimos con Obijuan [6] cuando dice que “es el momento de redefinir: hay que reinventar la rueda, pero libre”.

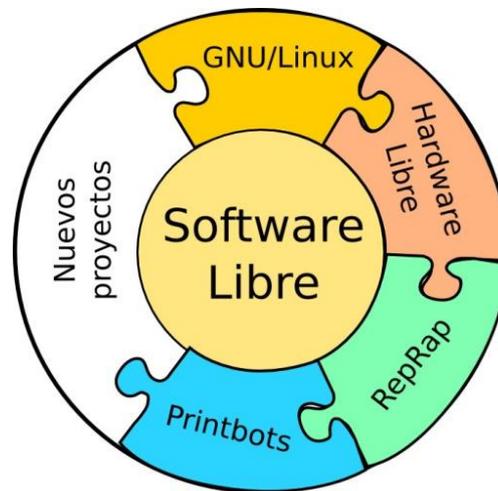


Figura 1. Creación de un Patrimonio Tecnológico de la Humanidad.

Fuente: Obijuan en TEDx

Lo contrario del Software Libre es el Software Privativo, que no se puede adquirir como producto material sino que lo que se adquiere es un permiso de uso o licencia conocida como EULA (End-User License Agreement) que plantea restricciones de uso y tiempo de vigencia del producto, y no se puede acceder al código fuente. En general no permite la copia ni la libre distribución. Dentro del Software Privativo está el Freeware, que son aquellos programas de licencia gratuita que se pueden descargar aceptando ciertas condiciones que decide el propietario (por ejemplo DraftSight o SketchUp Make). El Freeware, que por supuesto nos interesa por ser gratuito, normalmente es restringido al uso personal no profesional o para uso educativo y siempre no comercial.

Las grandes empresas desarrolladoras de software propietario fueron liberando licencias de sus productos insignia para uso educativo. Autodesk fue una de las últimas que las liberó hace un par de años. Entendemos que no es una concesión porque sí, sino por una intencionalidad o estrategia para no perder mercado en el ámbito educativo, no sólo por el desarrollo de software propietario por parte de la competencia, sino porque en la práctica el sistema de software privativo, legal y de pago se torna imposible de sostener por la mayoría de las universidades públicas en todo el mundo y muchas han migrado hacia el Software Libre para sus distintas áreas funcionales. Además, lógicamente estas empresas saben de los avances permanentes en el desarrollo del Software Libre.

El ámbito universitario es el que está mejor preparado para poder investigar y aprender a utilizar Software Libre (aunque en general tiene una curva de aprendizaje más extendida en el eje del tiempo). Imaginemos una universidad donde todos sus docentes y alumnos dominen alternativas diversas en Software Libre y a lo largo de la carrera, donde sea necesario utilizar software, se logren las competencias de cada especificidad con el uso de software libre. Estos jóvenes profesionales así formados, al incorporarse al mundo productivo, estarían en mejores condiciones de manejar cualquier tipo de software ya sea libre o privativo, incluso podrían asesorar a las empresas en las mejores alternativas y más económicas. Esto por suerte ya ocurre en muchas universidades. Por ejemplo, en la carrera de Licenciatura en Ciencias de la Computación de nuestra Universidad se utiliza solamente software libre, habiendo emitido esta carrera una resolución al respecto (ver <http://www.fceia.unr.edu.ar/lcc/resoluciones.php>).

A poco de comenzar con nuestro proyecto de investigación, ya estábamos conscientes de que nos enfrentaríamos a nuevos desafíos, ya que como nuevo paradigma de pensamiento, el Software Libre plantea un cambio filosófico que demanda un esfuerzo y compromiso de trabajo diferente a lo que estamos acostumbrados. Como afirmábamos en [1] “el aprendizaje del Software Libre requiere una fuerte capacidad de autoaprendizaje. Es bien conocido que los cursos que se ofrecen apuntan a los productos comerciales. Foros de ayuda, blogs, manuales, tutoriales, etc., son recursos que siempre están disponibles en Internet puestos a disposición por la comunidad de usuarios y desarrolladores. Los conceptos de autoaprendizaje y aprendizaje colaborativo, concuerdan con la formación que debe brindar una Universidad Pública como la nuestra”. Pese a que el Software Libre permite acceder al código fuente, siempre tuvimos en claro, como docentes que somos, que nuestro abordaje de la temática del Software Libre era desde el rol principal de usuarios finales, no de desarrolladores. Al respecto, coincidimos con Mochi Alemán [7] cuando plantea que “el código fuente puede convertirse sólo en una ambición para aquéllos que tienen necesidades y capacidades suficientes para leerlo, estudiarlo y mejorarlo. La mayoría de las personas, sin embargo, sólo quieren que el software funcione y no se preocupan demasiado sobre cómo lo hace...”

El concepto de que el Software Libre es sólo para expertos en informática, es un prejuicio o mito que pudo haber tenido validez hace 10 o 20 años, pero en la actualidad no presenta impedimentos de uso para alguien que está habituado a trabajar en el medio digital. Tal como ya hemos dicho [8], “pretendemos sumar conocimiento y derribar mitos respecto del Software Libre, poner en juego la capacidad de aprender y auto-aprender (tan importante en la producción del conocimiento en la universidad) y ante un mismo problema saber desenvolvernos con distintas alternativas de solución basadas en programas CAD diferentes y elegir la mejor o más adecuada”. Sabemos que “la filosofía del Software Libre, como nuevo paradigma de esta era, genera en muchos una resistencia al cambio” [9]. Tal vez esto suceda porque este paradigma es antagónico al paradigma comercial del Software Privativo que domina el mundo digital, y al cual el usuario común está acostumbrado ya que la gran mayoría, en la que nos incluimos, estamos formados en el universo del Software Privativo. En este sentido, muchos colegas nos plantean dudas que coinciden con estos interrogantes: “¿Cómo se puede vivir produciendo software libre?, ¿se arrepentirán los programadores en el futuro de darlo gratis?, ¿conseguirá Linux competir con Microsoft?, ¿cómo hacer para que los grandes productores de software no se apoderen de Linux y lo privaticen?, ¿serán cada vez más fáciles las herramientas para usar Linux, o quedará confinado al uso de programadores y fanáticos de la informática?” [7].

Estas cuestiones nacen desde la visión del paradigma del software privativo, y pierden importancia cuando se las considera desde el paradigma del Software Libre, que existe y va a seguir existiendo, y cuya comunidad de desarrolladores sigue creciendo en todo el mundo.

Un ejemplo claro de conflictividad privativo-libre es cuando faltan controladores de hardware para Linux. Dicho de otra manera, algunos dispositivos como impresoras, placas gráficas, etc., funcionan con menor rendimiento o no funcionan en sistemas Linux porque los fabricantes de ese hardware no crean los controladores para este sistema. Solo lo hacen para Windows o Mac, ya sea por motivos económicos (el mercado de sistemas Linux es pequeño ya que la mayoría de sus clientes tienen Windows) y/o porque consideran que liberar código es dar su know-how a la competencia. Así, los voluntarios del software libre deben crear sus propios controladores para sistemas Linux. Esto complica al usuario común.

Otro ejemplo de conflictividad es la falta de estándares para los programas de diseño. AutoCAD guarda en el formato privativo DWG (Drawing) que debido a la gran cuota de uso que tiene AutoCAD se ha convertido en un formato estándar, pero de facto. Para poder trabajar con un mismo archivo de dibujo DWG desde programas libres se debe recurrir a métodos de exportación que suelen ser engorrosos y pueden ocasionar pérdida de datos. Programas como AutoCAD o SolidWorks no están desarrollados para el sistema operativo Linux. Las empresas que desarrollan Software Privativo en general sólo lo hacen para sistemas privativos como Windows o Mac. Por el contrario, la mayoría de los programas libres que utilizamos son multiplataforma (corren en Linux y en Windows), y esto nos da la opción de utilizarlos en uno u otro sistema operativo según nos convenga. Ver Tabla 1.

Tabla 1. Software Libre y Freeware multiplataforma que más utilizamos.

Tipo	Software Libre y gratuito	Freeware	Sitio Web oficial	
CAD 2D	LibreCAD		http://librecad.org/	
		DraftSight	www.3ds.com/es/productos-y-servicios/draftsight/	
Modelado paramétrico 3D	FreeCAD		http://freecadweb.org/	
Modelado 3D		SketchUp	http://www.sketchup.com/es	
Modelado 3D, iluminación, render y animación	Blender		http://www.blender.org/	
Edición de imágenes y Diseño Gráfico	Gimp		http://www.gimp.org/	
	Inkscape		http://www.inkscape.org/	
Paquete de Oficina	LibreOffice	Docs. de Texto	Writer	https://es.libreoffice.org/
		Diseño Gráfico	Draw	
		Presentaciones	Impress	
		Planilla cálculo	Calc	

Fuente: elaboración propia

Como consecuencia de nuestra actividad de investigación, nos constituimos como Grupo SoLCAD (Investigación sobre Software Libre y Freeware CAD) y creamos un sitio Web donde puede conocerse nuestro trabajo de investigación desde que comenzamos en el año 2012 (www.fceia.unr.edu.ar/solcad).

2. Metodología de este trabajo

Con la intención de mostrar una amplitud de temáticas referidas a nuestra disciplina donde utilizamos Software Libre, a continuación presentaremos en primer lugar un trabajo de diseño de ingeniería, en segundo lugar contenidos para el aprendizaje de la asignatura Representación Gráfica donde vamos a mostrar y comparar la resolución de un mismo ejercicio de modelado sólido hecho con AutoCAD y con FreeCAD, y en tercer lugar el proceso de elaboración de un vídeo didáctico sobre el tema Vistas a partir de combinar distintos programas libres y gratuitos. Finalmente pondremos a disposición las conclusiones y recomendaciones, e indicaremos cómo pensamos seguir avanzando en futuros trabajos con eje en la temática que nos ocupa.

Los programas que más utilizamos son FreeCAD y Blender, de los que haremos una breve descripción. Luego existe un conjunto de Software Libre y Freeware que consideramos complementarios pero fundamentales a la hora de concretar un proyecto. Creemos que es mucho más enriquecedor para el aprendizaje combinar e interactuar con distintos programas que con uno solo. La Figura 2 es una infografía que muestra la interacción del software que más utilizamos y explica nuestra metodología de trabajo a la hora de resolver un proyecto. Dicha infografía fue confeccionada con el programa Draw de la suite LibreOffice.

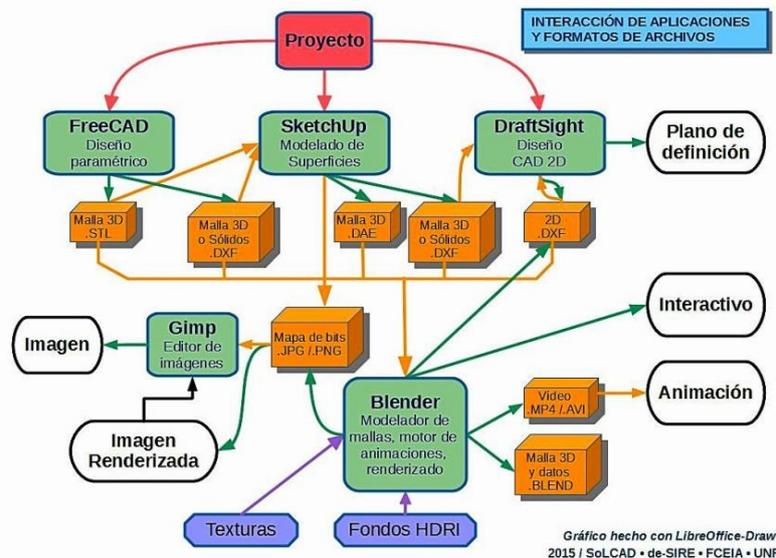


Figura 2. Infografía. Interacción de aplicaciones libres y formatos de archivos.

Fuente: elaboración propia [8]

2.1 Descripción de FreeCAD

FreeCAD [9,10] es una aplicación de modelado paramétrico CAD/CAE de arquitectura modular. Está orientado a la ingeniería mecánica y al diseño de producto, pero también se ajusta a un amplio rango de usos de otras especialidades de ingeniería, y arquitectura. Sus herramientas son similares a las de Catia, SolidWorks o Solid Edge, por lo tanto también pertenece a la categoría de MCAD, PLM, CAx y CAE. La interfaz de FreeCAD está organizada con escenarios de trabajo o “workbenchs” (Sketcher, Drawing, Part, Part Design, Mesh Design, Arch, RayTraicing, etc.) que permiten seleccionar las barras de herramientas necesarias en base a la tarea a desarrollar. Sus capacidades se basan en el kernel OpenCascade, un núcleo CAD de nivel profesional que cuenta con la manipulación de geometría 3D avanzada y objetos. Este programa está aún en una etapa de fuerte desarrollo.

2.2 Descripción de Blender

Blender [9,11] es un potente programa para modelado 3D, iluminación, render, animación y creación de gráficos tridimensionales. También es para composición digital utilizando la técnica procesal de nodos, edición de vídeo, escultura y pintura digital. Se pueden desarrollar videojuegos ya que posee un motor de juegos interno. Tiene la capacidad de representar simulaciones físicas como deformación de sólidos, acciones con fluidos, humo, etc.

La interfaz predeterminada se compone de Editores, Cabeceras, Botones contextuales, Paneles y Controles. Se llaman Ventanas a la parte del programa que responde a una función específica (ventana de Vista 3D, de Propiedades, de Edición de secuencias de vídeo, de Editor de nodos, etc.). Cada ventana tiene su propia cabecera superior o inferior. Rigen tres principios invariables para la interfaz: 1) Nada se superpone: no hay ventanas ni herramientas flotantes, todo está en un mismo plano; 2) No hay bloqueos: ninguna herramienta espera que el usuario haga algo, no hay área de comandos; 3) No modal: deberá ser siempre tan consistente y predecible como sea posible sin cambiar métodos comunes (mouse, teclado). La interfaz del usuario es un entorno multitareas, simultáneo, automatizable y 100% personalizable. Al igual que con la ventana principal, cada ventana nueva se puede subdividir.

3. Resultados y Discusión

3.1 Diseño de ingeniería: impresora 3D

Antes del diseño CAD se diseñaba dibujando manualmente en el papel las vistas o proyecciones ortogonales de los objetos con el respectivo dimensionamiento, para luego construir lo diseñado a partir de esos *planos*. A partir de 1980, con la llegada del CAD, se procedía igual, pero digitalizando los planos en el ambiente computacional. Con el nuevo siglo llegó la tecnología del modelado sólido 3D con la posterior generación automática de los planos de fabricación, invirtiéndose la secuencia del proceso de diseño. Así se pasó de una lógica de pensamiento bidimensional (esquemáticamente: $2D \rightarrow 3D$), a una lógica de pensamiento tridimensional, (esquemáticamente: $3D \rightarrow 2D$), generando los planos al final del proceso. Este hecho hoy convive con la llegada de la tecnología de impresión 3D, con la que entramos en el paradigma de pasar directamente del modelado 3D a la fabricación del producto sin necesidad de confeccionar los planos (esquemáticamente: $3D \rightarrow \text{producto final}$). En muchos procesos industriales hoy se utilizan estas tecnologías CAD-CAM (Diseño y Manufactura Asistida por Computadora) y CNC (Control Numérico Computarizado) para la fabricación de un producto, modelo o prototipo directamente desde el ambiente CAD 3D. Con las impresoras 3D se imprimen en plástico una variedad de objetos, como ser prototipos y matricería o prefabricación en diseño industrial o arquitectura, hasta prótesis médicas, etc.

En lo referido al Prototipado Rápido (PR) por impresión 3D, existen diferentes tecnologías que pueden consultarse en [12]. El enfoque común en todas las técnicas actuales de PR por adición de material consiste en el modelado sólido en sistema CAD, la exportación en malla geométrica STL (Stereo Lithograph), la división del modelo en capas muy delgadas y paralelas en el plano x-y que se van adicionando en el sentido del eje z. Una de las técnicas más difundidas de PR en nuestro medio es el FDM (Modelado por Deposición Fundida). Se basa en la extrusión en caliente de un filamento de plástico que puede ser ABS (Acrilonitrilo-Butadieno-eStireno) o PLA (Poliácido Láctico). El hilo de plástico pasa a través de un dispositivo, denominado extrusor, que se calienta hasta que lo funde sin llegar a derretirlo por completo. El dispositivo extrusor va expulsando este hilo casi fundido y lo va depositando en una base con una temperatura definida, y se enfría y solidifica rápidamente para contribuir a la correcta conformación de cada capa de la pieza.

Nos propusimos modelar una impresora 3D y hacer un vídeo de animación para tomar el conocimiento necesario sobre la impresora y su funcionamiento. Pero para esto era necesario obtener los archivos digitales de las partes de la impresora. Esto nos orientó hacia el proyecto Rep-Rap y el concepto del Hardware Libre. Nos basamos en la metodología de colaboración habitual en foros de FOSS (Free Open Source Software) que es compartir las partes de un proyecto en archivos listos para su uso. En una aplicación se pueden compartir el código fuente y el código compilado para su ejecución. En el caso de las partes de una máquina, como por ejemplo la impresora, se comparten los archivos para ser impresos directamente en otra impresora 3D. La ausencia total de planos de definición es una característica muy notoria. Se comparten las piezas listas para ser replicadas.

Elegimos un modelo de impresora que trabaja por FDM, la conocida como M Prime ONE [13]. Descargamos las partes de la M Prime ONE en formato de archivo para ser abiertos en FreeCAD, donde se hizo el montaje del conjunto para su revisión. Ver la Figura 3.

En esta oportunidad continuamos con la línea de trabajo de interacción entre los programas FreeCAD y Blender, que pusimos en práctica en [9], y se resume en dos etapas:

1ª etapa: Con FreeCAD hacemos el modelado paramétrico de cada componente del diseño y posteriormente el montaje del conjunto en un nuevo archivo que luego se exporta en formato STL, un formato CAD que define la geometría de objetos 3D excluyendo información como color, texturas o propiedades físicas.

2ª etapa: Con Blender logramos la imagen fotorrealista del objeto. Primero abrimos el archivo del conjunto en formato STL con todas las partes. Luego asignamos los materiales y atributos de diseño a cada parte, terminando la modelización con render en imágenes fotorrealistas y, en esta oportunidad, hicimos un vídeo del funcionamiento mecánico de la impresora.

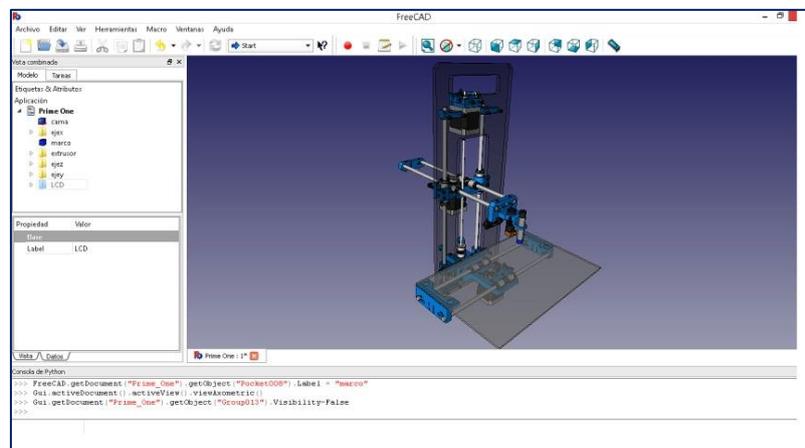


Figura 3. Impresora M Prime ONE: montaje en FreeCAD.
Fuente: elaboración propia

En la Figura 4 mostramos un collage de vistas en perspectiva logradas en FreeCAD. Luego de cumplida esta tarea, exportamos el conjunto desde FreeCAD al formato STL (archivo único).

Continuamos la modelización 3D en Blender con el propósito final de realizar un vídeo o película donde se vean los movimientos mecánicos de la impresora. Decidimos importar el conjunto como una sola malla aprovechando la tarea de ensamble hecha en FreeCAD. Gracias a las diversas herramientas de selección muy precisas que posee Blender, procedemos a separar las partes principales que se moverán en la animación y las que son de diferentes materiales (plástico, metal brillante, metal opaco, etc.). Ver Figura 5.

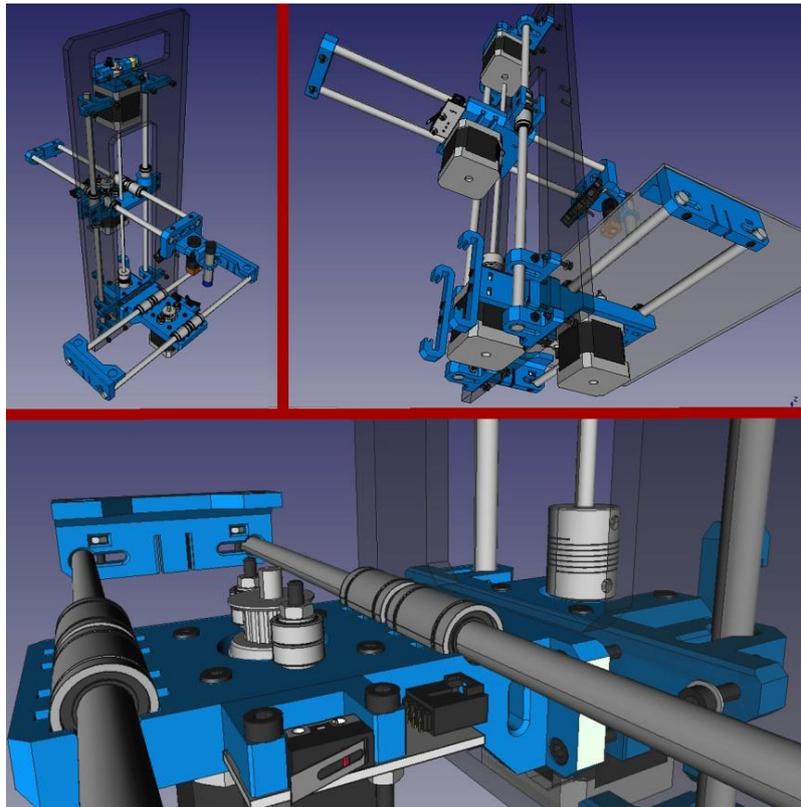


Figura 4. Perspectivas de la Impresora Prime ONE en FreeCAD.

Fuente: elaboración propia

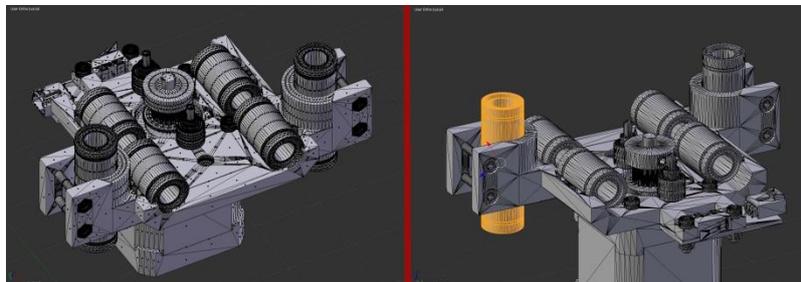


Figura 5. Edición de mallas, separación de partes y selección.

Fuente: elaboración propia

Creamos estos materiales en el editor, pero se pueden crear en otro archivo y se importan a un nuevo proyecto. Luego los asignamos a cada parte y comenzamos el proceso de iluminación de la escena. Cada material es probado en diferentes entornos de luz. Ver Figura 6.

Aplicamos anisotropía, método de filtrado de luz para mejorar la calidad de la textura en una superficie, que define la dirección más probable en que se dispersa la luz. Ver Figura 7.

En este caso utilizamos algunas lámparas spots difusas y el modelo Sol, que logra sombras duras y tiene rayos paralelos. Para ver los efectos de luces y sombras dividimos la pantalla en múltiples vistas como se observa en la Figura 8, con el motivo de definir las formas a través de sus sombras. La perspectiva es la vista de la cámara donde se aprecia el efecto de las luces. Las vistas planas sirven para mover las luces aplicadas.

Ajustamos la cámara con una lente normal de 35 mm y con profundidad de campo total. En la Figura 9, a la izquierda, se muestra una prueba de cámara.



Figura 6. Asignación de Materiales. Render en tiempo real.
Fuente: elaboración propia

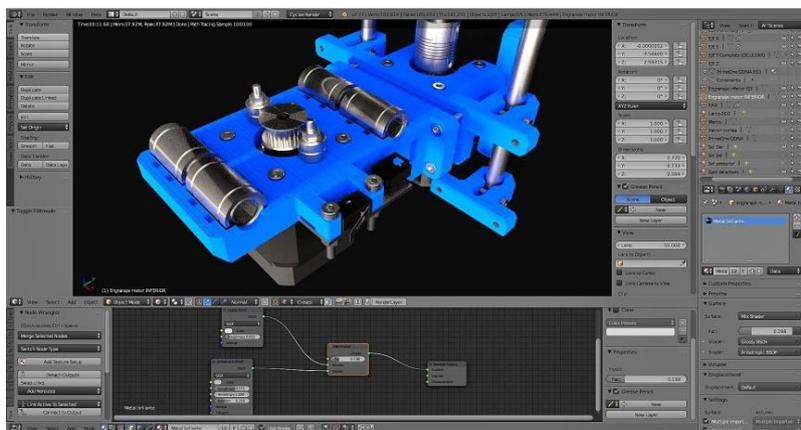


Figura 7. Anisotropía. Vista perspectiva.
Fuente: elaboración propia

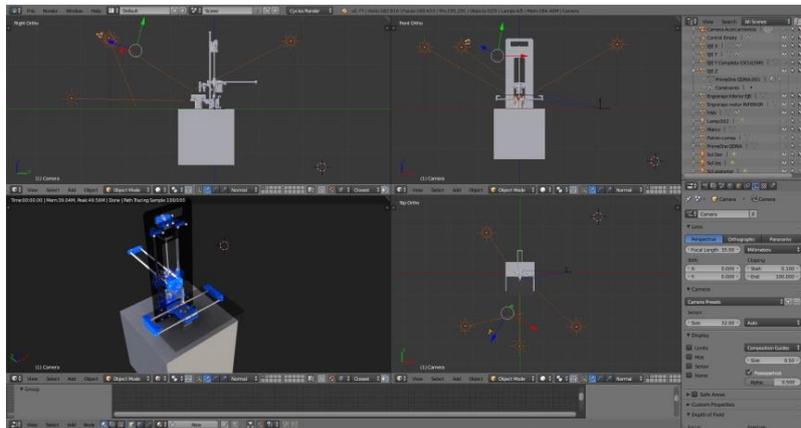


Figura 8. Prueba de luces en múltiples vistas.
Fuente: elaboración propia

Para realizar el proceso de animación creamos un objeto invisible llamado Empty (vacío) al que se le asocian los movimientos según cada eje de la impresora (ver Figura 9 a la derecha). Restringimos dichos movimientos con respecto a los recorridos máximos de cada eje. Moviendo el Empty logramos que la impresora siga la misma dinámica.

En la línea de tiempo creamos claves de la posición del Empty, para la cámara y para algunas luces. Una vez hecho eso se procedimos a renderizar imágenes del proceso y la animación.

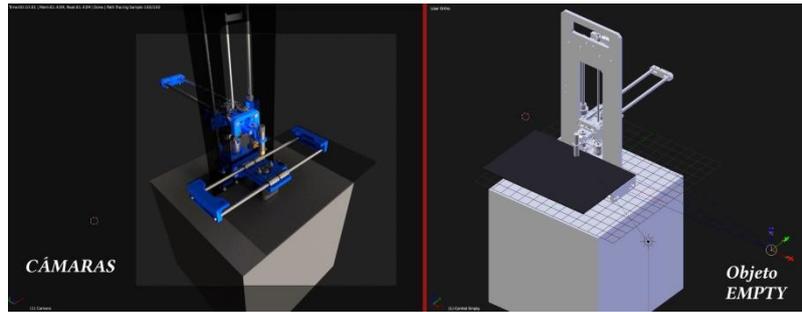


Figura 9. Prueba de cámaras y creación de Objeto Empty
Fuente: elaboración propia

Todo el proyecto en Blender lo hicimos en una PC dedicada y optimizada para gráfica digital con las últimas tecnologías de procesamiento gráfico: GPU multinúcleos (Graphics Processor Unit); SLI (Scalable Link Interfaz) y CUDA (Arquitectura Unificada de Dispositivos de Cómputo). El vídeo puede verse en nuestro canal YouTube [14]. La música fue compuesta por Hernán Pangia (miembro del grupo SoLCAD y músico) utilizando el programa LMMS (Linux Multimedia Studio), que también es software libre y gratuito.

3.2 Contenidos para el aprendizaje de Representación Gráfica

Uno de los ejercicios de modelado sólido que damos a los alumnos de nuestra comisión es el de la Figura 10: hacer el modelado 3D de una pieza mecánica. El programa con que se trabaja en todas las comisiones de nuestra materia (1° año de Ingeniería) es AutoCAD.

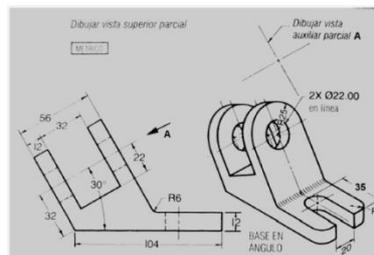


Figura 10. Datos para el modelado.
Fuente: Libro “Dibujo Técnico” – Spencer, Dygdon y Novak

La Figura 11 muestra el proceso de modelado con AutoCAD. La Figura 12 es un collage del proceso de modelado paramétrico con FreeCAD. El proceso no es idéntico, pero la lógica de diseño es bastante parecida. Y el resultado al que se llega es el mismo.

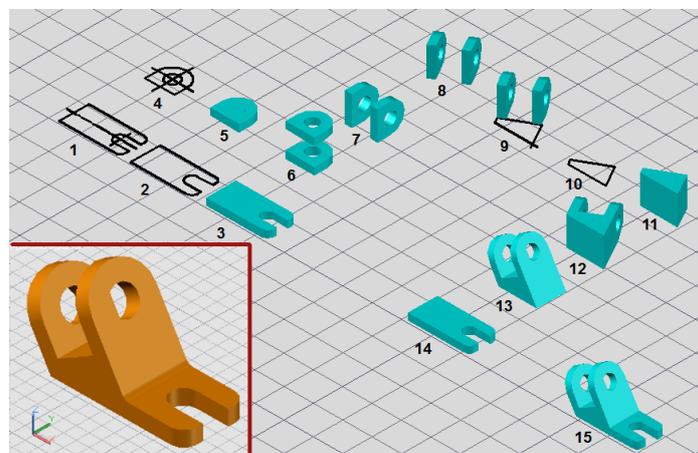


Figura 11. AutoCAD – Proceso de Modelado
Fuente: elaboración propia

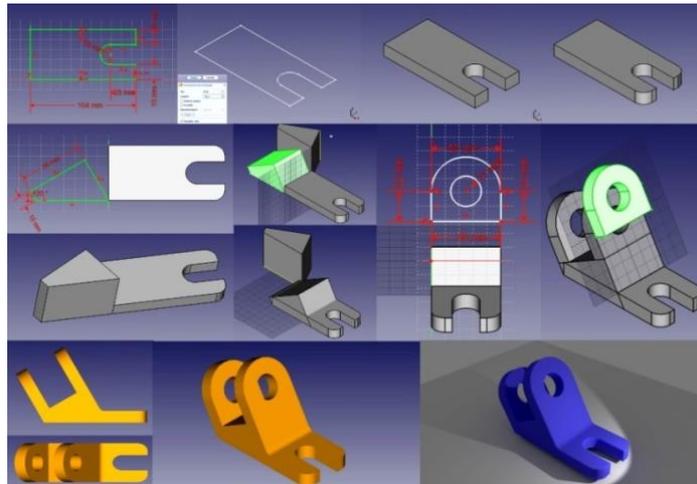


Figura 12. FreeCAD: proceso de Modelado y un render en Blender
Fuente: elaboración propia

Planteamos también el cálculo del volumen de la pieza que en AutoCAD puede obtenerse desde el Menú o desde línea de comandos mientras que en FreeCAD se calcula desde la consola de Python donde se da una sentencia. En la Tabla 2 se indican estos comandos y los resultados obtenidos, que son prácticamente iguales, con un error menor que 0,01 en mm^3 .

Tabla 2. Cálculo de volumen.

Software	Comando utilizado	Resultado en mm^3
AutoCAD	Menú Herramientas / Consultar / Volumen	131480,2965
FreeCAD	App.ActiveDocument.ActiveObject.Shape.Volume	131480,2942

Fuente: elaboración propia.

Ambos programas se comportaron satisfactoriamente, aunque debemos resaltar que FreeCAD por ser un modelador paramétrico tiene un proceso de diseño más comparable con SolidWorks o SolidEdge. AutoCAD incluye características paramétricas en sus últimas versiones y es un programa que demanda más recursos de sistema que FreeCAD.

3.3 Vídeo didáctico sobre tema Vistas

Una gran dificultad para los estudiantes de primer año de Ingeniería es comprender el espacio y la representación de los objetos en el plano, la asociación de las vistas ortogonales con la perspectiva de los objetos. Elaboramos un vídeo de apoyo para comprender el espacio y las vistas de un objeto sencillo, a partir del primer trabajo práctico de la materia Representación Gráfica consistente en representar las vistas y perspectiva isométrica de un modelo físico o maqueta como se ve en la Figura 13. Primero los alumnos deben dibujar las vistas y perspectiva isométrica con lápiz a mano alzada [15] y luego hacer en AutoCAD el modelado 3D y las vistas automáticas inclusive la perspectiva isométrica [16].

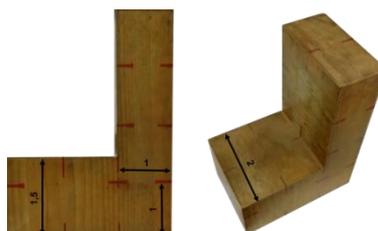


Figura 13. Fotos del modelo físico.
Fuente: elaboración propia

La elaboración del vídeo didáctico fue una experiencia valiosa porque combinamos el uso de dos sistemas operativos: Windows y Linux. Esto fue debido a que los programas utilizados no eran multiplataforma, salvo Gimp. Ver listado en la Tabla 3.

Tabla 3. Software utilizado en la elaboración del vídeo didáctico.

Software	Tipo	Sist. Operativo	Usado para
SketchUp Make	Freeware educativo	Windows	Modelado, animación y vídeo
AutoCAD	Freeware educativo		Dibujos complementarios
Snagit	Privativo – V. Trial		Cortes de pantalla y edición
Gimp	Software Libre y gratuito	Multiplataforma	Edición de imágenes
KSnapshot		Linux	Cortes de pantallas
Kdenlive			Edición de vídeo

Fuente: elaboración propia.

SketchUp Make es un programa excelente para el modelado 3D y la creación y animación de escenas con posterior salida de vídeo que se hizo en formato AVI. Ver Figura 14.

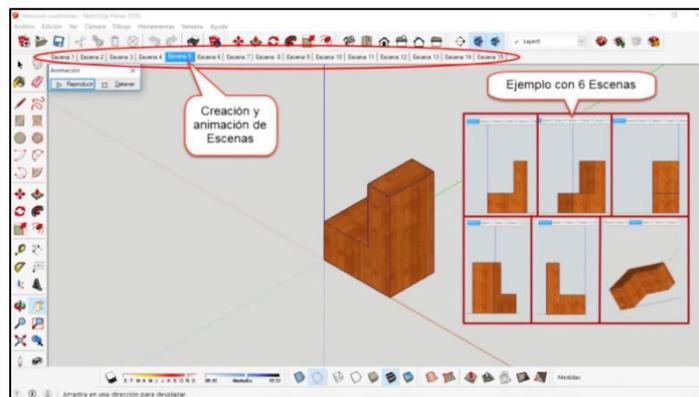


Figura 14. Modelado 3D y Animación de Escenas con SketchUp Make, en Windows.

Fuente: elaboración propia

Con el vídeo base AVI y elaboradas y editadas las imágenes complementarias, trabajamos en Linux en la edición final del vídeo con el programa Kdenlive, un editor de código abierto fácil de usar, potente, con muchos efectos creativos (comparable con el privativo Adobe Premiere). La Figura 15 muestra la interfaz gráfica de Kdenlive.

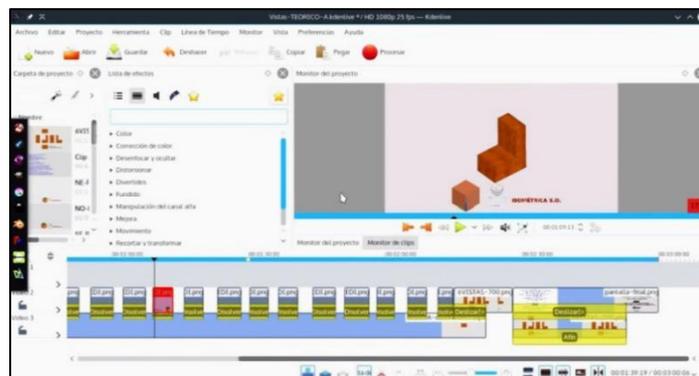


Figura 15. Edición de vídeo en Kdenlive en Sistema GNU-Linux.

Fuente: elaboración propia

La Figura 16 muestra miniaturas capturadas del vídeo final logrado con Kdenlive, que se puede ver en nuestro canal YouTube [17], publicado bajo licencia Creative-Commons.



Figura 16. Capturas de pantalla del vídeo.
Fuente: elaboración propia

4. Conclusiones y recomendaciones

Hemos presentado una amplitud de temáticas referidas a nuestra disciplina gráfica para seguir difundiendo nuestra experiencia y así animar a otros a migrar hacia las aplicaciones libres. No vamos en detrimento de los excelentes programas privativos que existen, pero entendemos que en nuestra actividad debemos darle la oportunidad al Software Libre. Aprendimos que la migración a Software Libre no es sencilla. Una cosa es la migración de usuarios que investigan y se comprometen con esta tecnología, y otra cosa es la migración de una organización, como una institución educativa o una empresa. El gran escollo es la resistencia al cambio. Una migración brusca es perjudicial. Debe ser formativa y gradual, seleccionando y capacitando líderes tecnológicos entre las personas más competentes y reconocidas por sus propios colegas, que sepan comunicar muy bien todo lo atinente a ese proceso de cambio.

Entendemos que hablar de ventajas o potencialidades del uso de Software Libre sobre el Software Privativo sin considerar el paradigma y su filosofía es caer en un análisis que puede llevar a conclusiones erróneas. Y no es nuestra intención. Porque tanto los programas privativos como los libres que se destinan a un uso similar, puede ocurrir que alguno se desempeñe mejor, igual, o peor que el otro. Las ventajas o potencialidades se dan justamente en el marco de su filosofía: legalidad y libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software. Además los programas libres son más fáciles de descargar e instalar. Y si a esto le sumamos la gratuidad, es algo incontrastable (hoy una licencia individual por dos años de AutoCAD o de 3D Studio Max cuestan alrededor de u\$s 3000). En general, el Software Libre hace un uso más eficiente de los recursos de hardware. Por ejemplo Blender posee drivers dedicados para trabajar dentro de su entorno. El entorno completo de instalación de Blender no supera 0.5 Gb cuando AutoCAD ocupa un DVD. Aclaremos que es mejor comparar Blender con 3D Studio Max y otros del estilo. Ver la tabla de comparación hecha por Saint-Moulin [18].

Recomendamos instalar GNU-Linux en una partición del disco, para aprender a trabajar 100% en Software Libre desde el mismo sistema operativo, y dejar Windows para aquellos programas que necesitamos forzosamente y sólo se desempeñan en Windows. *Aprender a aprender* programas libres y nuevas formas de pensar y de hacer enriquece nuestro conocimiento y nuestra cultura digital. Como continuidad de nuestro trabajo, pretendemos profundizar el tema de las impresoras 3D dentro del proyecto de Hardware Libre con la intención de adquirir una y comenzar a producir prototipos didácticos.

5. Referencias

- [1] MORELLI R.D.; MORELLI L.R. (2012). El Software Libre en la Enseñanza de la Representación Gráfica. *Gráfica del Diseño. Tradición e innovaciones*, EGRAFIA, 2012. La Plata, p. 219-223.
- [2] BROCCA, J.C.; CASAMIQUELA, R. (2005). Las licencias de software desde la perspectiva del usuario final. *Revista Pilquen – Univ. Nac. Comahue*, Viedma, n. 7, p. 5.
- [3] STALLMAN, R.M. (2004). *Software libre para una sociedad libre*, Madrid: Traficantes de Sueños, p. 59-60.
- [4] WIKIPEDIA (2016). Hardware libre. *Wikipedia, la enciclopedia libre*. URL: https://es.wikipedia.org/wiki/Hardware_libre
- [5] WIKIPEDIA (2016). Cultura libre. *Wikipedia, la enciclopedia libre*. URL: https://es.wikipedia.org/wiki/Cultura_libre
- [6] GONZÁLEZ J. (2013). Yo, Maker. *Charla en TEDx, evento mundial de Tecnología, Entretenimiento y Diseño*. Valladolid. URL: [youtube.com/watch?v=94_uafCR0w](https://www.youtube.com/watch?v=94_uafCR0w)
- [7] MOCHI ALEMÁN P.Ó. (2002). El movimiento del software libre. *Revista Mexicana de Ciencias Políticas y Sociales*, México D.F., v. XLV, n. 185, p. 73-89
- [8] MORELLI R.D. (2015). Opciones para la Enseñanza de la Representación Gráfica a partir de Software Libre y Gratuito. *RBEG - Revista Brasileira de Expressão Gráfica*. Revista Online – Brasil, v. 3, n. 1, p. 16-40.
- [9] MORELLI R.D.; PANGIA CTENAS H.A.; NIEVA L.S. (2015). Modelado Paramétrico 3D, Render y Animación con Software Libre: Interacción Freecad + Blender. *Geometrias & Graphica 2015 Proceedings*, Lisboa, v.1, c.1, p.023-036.
- [10] FREECAD WIKI (2016). Características generales. *Sitio Web Oficial de FreeCAD*. URL: www.freecadweb.org
- [11] BLENDER (2016). Contenidos del Manual de Referencia de Blender. *Manual Web Oficial de Blender*. URL: www.blender.org/manual
- [12] BORTOLATO M.G.; LENTI C.A.; MORELLI R.D.; VERGER G.I. (2008). El Prototipado Rápido en plástico ABS como herramienta didáctica. *EGRAFIA – Universidad Nacional de San Juan*. EGRAFIA, San Juan, p. 2-4
- [13] REP-RAP WIKI (2015). M Prime One. *Impresora 3D libre*. URL: http://reprap.org/wiki/M_Prime_One/es
- [14] PANGIA H.A. (2016). Movimientos de una impresora 3D M Prime ONE. *Canal YouTube del Grupo SoLCAD*. URL: https://www.youtube.com/channel/UCUuUzry2Jhlt_bAd3c4SzXg
- [15] MORELLI R.D. (2015). Teórico A. *Croquis de vistas*. FCEIA, Rosario. URL: <http://www.fceia.unr.edu.ar/de-sire/TEORICO-A.pdf>
- [16] MORELLI R.D. (2015). TP 03. *Modelado 3D y Vistas automáticas*. FCEIA, Rosario. URL: <http://www.fceia.unr.edu.ar/de-sire/TP-03-DATOS.pdf>
- [17] MORELLI R.D. (2016). Vídeo de apoyo para el tema Vistas. *Canal YouTube del Grupo SoLCAD*. URL: <https://youtu.be/CdPmgjoXvz8>
- [18] SAINT-MOULIN B. (2007). 3D softwares comparisons table. *Revista online TDT3D*. Bélgica, online 101/0. URL: http://www.tdt3d.be/articles_viewer.php?art_id=99