

**V CONGRESO INTERNACIONAL DE EXPRESIÓN GRÁFICA  
XI CONGRESO NACIONAL DE PROFESORES DE EXPRESIÓN GRÁFICA  
EN INGENIERÍA, ARQUITECTURA Y ÁREAS AFINES  
EGraFIA 2014  
Rosario, ARGENTINA  
1, 2 y 3 de octubre de 2014**

**DATOS DEL/ LOS AUTOR/ES DEL TRABAJO**

Pangia Ctenas, Hernán Alfredo - Nieva, Luis Sebastián  
Universidad Nacional de Rosario  
Facultad de Ciencias Exactas, Ingeniería y Agrimensura  
Av. Pellegrini 250–Rosario–Te 0341-153-28-4496  
E-Mail: hernanpangia@gmail.com - 5e6anieva@gmail.com  
Rosario – Argentina

**DESARROLLO DE UNA EXTENSIÓN RUBY PARA AUTOMATIZAR  
LA REPRESENTACIÓN DE VISTAS EN SKETCHUP**

Disciplina: Ingeniería

Eje de interés: Investigación: Gráfica Analógica y Gráfica Digital – Nuevas Herramientas.

**ABSTRACT**

SketchUp is very popular CAD software, especially in architectural projects and it has a Freeware version. PRO version has many extra functions but using extensions, Sketchup does the almost same work. These extensions are written in Ruby open source programming language and these could be shared in a “plugins warehouse”. Anyone with programming knowledge could write a plugin and share it. Our investigation group is about free alternatives to licensed CAD applications. We developed an extension that allows representing automatic views of models. In this paper, we show how to use it, and we explain the method of plugins development.

**RESUMEN**

SketchUp es un programa CAD para diseño por modelado 3D muy difundido, especialmente en el ambiente del diseño arquitectónico, pero que también es apto para representar muchos desarrollos en proyectos de ingeniería y cualquier otra disciplina que utilice la representación gráfica técnica. SketchUp es un software de tipo Freeware, es decir, privativo pero de licencia gratuita. La empresa que lo provee también ofrece su versión de pago, SketchUp Pro, que lógicamente tiene mayores funcionalidades, aunque la versión gratuita es muy buena y además permite aumentar sus funcionalidades a través de la instalación de complementos o extensiones. ¿Qué son, en concreto, las extensiones en el programa SketchUp? Son programas complementarios diseñados en un lenguaje específico, que se añaden al programa principal como herramienta para permitir a un usuario realizar uno o varios tipos de trabajos, por ejemplo para la automatización de ciertas tareas complicadas. Las extensiones para SketchUp se hacen con el lenguaje de programación interpretado y orientado a objetos llamado “Ruby”, creado en el año 1993 por el programador japonés Yukihiro Matsumoto, y su implementación oficial es distribuida bajo una licencia de software libre. Esto significa que puede conocerse el código fuente de una extensión Ruby. Cualquier usuario que sepa programar en Ruby puede desarrollar una extensión para SketchUp y utilizarla en su computadora para potenciar las posibilidades del programa, y también puede compartirla con la comunidad libremente, como de hecho lo son las muchísimas extensiones disponibles. Para tal fin, SketchUp dispone en la Web de un “Almacén de extensiones”, donde pueden descargarse gratuitamente las que a cada usuario le interese.

En el marco de nuestro Proyecto de Investigación en el que investigamos la utilización de software libre y gratuito en la enseñanza de la disciplina Sistemas de Representación, presentaremos en este trabajo una extensión que hemos desarrollado para SketchUp que permite representar vistas automáticas de cuerpos geométricos, piezas o modelos, para lo cual se intentará probar la funcionalidad de la extensión a través de ejemplos prácticos.

## 1.- INTRODUCCIÓN

Como se menciona en [1] debido a que el costo de las licencias de los programas informáticos es un tema relevante en el marco de nuestra Universidad que es pública y gratuita, se convierte en eje principal de nuestra actividad la utilización de software libre y gratuito. Este posee funcionalidad limitada respecto de sus análogos privativos, nos ocupa el intento de disminuir esta brecha o eliminarla a través de agregar comportamiento por medio de extensiones.

En la utilización del programa Sketchup en su versión gratuita en áreas como el desarrollo de proyectos arquitectónicos e ingenieriles, la misma potencialidad y sencillez que el programa exhibe para modelado en 3D puede reprochársele para trabajo en 2D, esto nos motiva a generar un nexo entre este y un programa altamente especializado en 2D como puede ser Draftsight, que es un software privativo de licencia gratuita similar a AutoCAD en su lógica, su interface y además trabaja con la extensión “.dwg” y “.dxf”. Con esta asociación de competencias esperamos obtener un resultado aceptablemente comparable a un software de licencia paga.

## 2.- METODOLOGÍA

- Realizaremos una breve mención de que documentación de apoyo puede ser útil en el desarrollo de una extensión de Sketchup.
- Mencionaremos que es un almacén de extensiones.
- Ilustraremos el proceso de incorporación de una extensión a Google Sketchup gratuito hasta la versión 8 y Trimble Sketchup 2013 (a partir del año 2013 Sketchup fue adquirido por la empresa Trimble).
- Se realizará el análisis de la mecánica de funcionamiento de la extensión “Vistas automáticas”.
- Llevaremos a cabo algunas pruebas de funcionamiento de la extensión en ejercicios de vistas de modelos tridimensionales.

## 3.- DESARROLLO

### Documentación de apoyo.

Un muy buen compendio de información acerca del programa Ruby orientado al desarrollo de extensiones de Sketchup puede consultarse en [3]. Así como también existen portales con abundante información, tutoriales y foros [4].

### Almacén de extensiones.

Los almacenes de extensiones son espacios en la web donde hay disponibles extensiones gratuitas para descargar. Del mismo modo que la documentación, la proliferación de oferta de extensiones en la red es considerable. Por eso nos inclinamos al sitio oficial de la empresa titular del software [5], no obstante, también es posible encontrar

extensiones en páginas de desarrolladores o sus blogs.

### Incorporación de una extensión a Sketchup.

Las extensiones son archivos de extensión “.rb” o “.rbz” que pueden o no estar acompañados de una carpeta y/o archivos accesorios. Se colocan en el directorio donde está instalado el Sketchup en la carpeta Plugins, y la próxima vez que se abra el programa ya estarán en funcionamiento. En este caso usaremos las extensiones:

- *Skp\_to\_dxf.rb*: que permite exportar el espacio de trabajo a un archivo “.dxf”.
- *As\_rubyeditor\_3-2.rbz*: [2] que es una consola que permite probar fragmentos de línea de código (Figura 1) agilizando la producción del mismo.
- *Workplane.rb*: extensión que permite definir un plano de proyección sobre el cual se pueden proyectar modelos tridimensionales.
- *Flattentoplane.rb*: extensión que realiza la proyección ortogonal de dichos modelos tridimensionales sobre el plano generado con *Workplane*. *FlattenToPlane* y *WorkPlane* trabajan en conjunto.

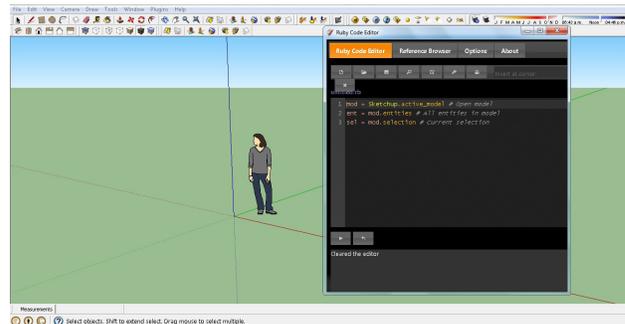


Figura 1. Consola editora de código.

### Diseño y funcionamiento de la extensión.

La denominada extensión “Vistas automáticas” fue ideada poniendo en consideración el hecho de que Sketchup ofrece la posibilidad, por medio de extensiones ya existentes, de proyectar de manera ortogonal un modelo tridimensional sobre un plano y de exportar un archivo “.dxf” (drawing exchange format) formato de intercambio de información entre archivos CAD muy versátil que permite la interacción con AutoCAD, Draftsight, etc... Esta exportación puede ser útil para realizar acotaciones, láminas de presentación, etc...

Teniendo en cuenta lo mencionado pretendemos a partir de un modelo en Sketchup, y por medio de la extensión Vistas automáticas lograr la visualización de las vistas normalizadas de manera automática. Eventualmente, en caso de ser necesario realizar acotaciones, u obtener láminas de presentación se puede exportar una representación en dos dimensiones.

Por medio de programación, se manipula el modelo en cuestión creando un número de copias igual al número de vistas normalizadas, en este caso las seis vistas que corresponden al desarrollo del cubo de proyecciones de los sistemas ISO-A o ISO-E, se ubican en el espacio de manera conveniente y se le realizan rotaciones de forma que los modelos desde un punto de vista particular nos muestren una aparente obtención de las vistas.

El funcionamiento de la extensión requiere solo un modelo en el espacio de trabajo con ubicación indistinta. En primer lugar ofrece al usuario la posibilidad de elegir el sistema ISO-E o ISO-A, indicado en la Figura 2 de obtención de las vistas y a continuación las produce.

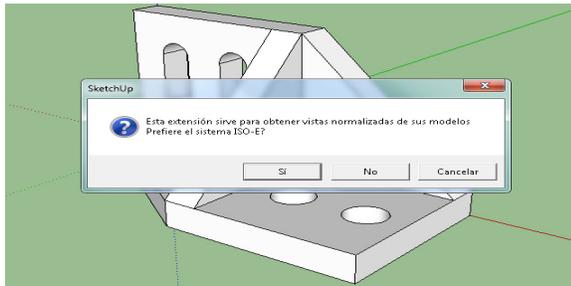


Figura 2. Cuadro de diálogo para elegir sistema.

Ejemplo del código que produce la rotación del modelo alrededor de un vector en el origen del sistema de coordenadas, con dirección al eje y una cantidad de grados guardada en la variable a se muestra en las siguientes dos líneas de código:

```
rr = Geom::Transformation.rotation origen,j,-a
new_group = group.transform!(rr)
```

Al concluir las rotaciones y algunas tareas accesorias se obtiene una disposición como la de la Figura 3 que obviamente no es visible para el usuario. En esta situación nos encontramos en posibilidad de elegir de donde miramos el conjunto. Cuestión que resuelven las siguientes líneas de código:

```
my_camera = Sketchup::Camera.new
my_camera.perspective = false
my_camera.height = 5*bounds.height
view = Sketchup.active_model.active_view
view.camera = my_camera
```

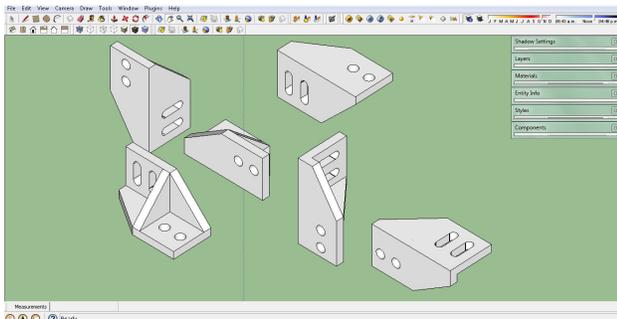


Figura 3. Situación intermedia de la resolución.

Que crean y posicionan una cámara mirando desde el eje z, en este caso vertical, a una distancia que permita ver toda la información de interés (Figura 4).

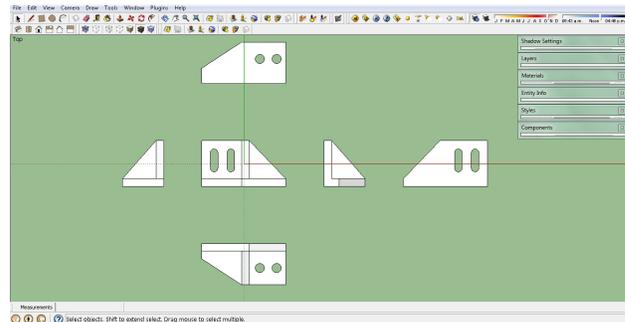


Figura 4. Vistas resultantes en sistema ISO-E.

En este momento tenemos aún un resultado en 3D, haciendo uso de las extensiones, en primer lugar de la extensión Workplane que ejecutamos desde el menú Herramientas -> WorkPlane -> New Plane. Al darle tres puntos esta extensión genera un plano como se observa en la Figura 5.

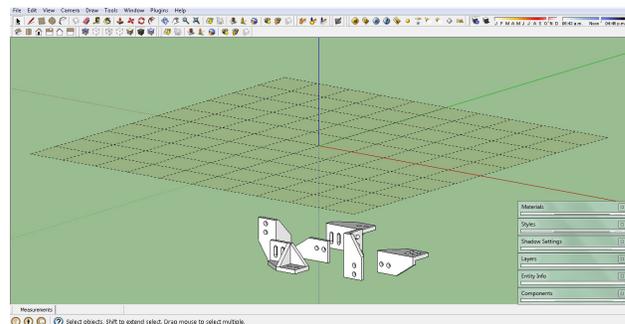


Figura 5. Plano de trabajo.

En esta situación seleccionamos simultáneamente los modelos y el plano y desde el menú Plugins -> Flatten to Plane obtenemos las proyecciones ortogonales como muestra la Figura 6.

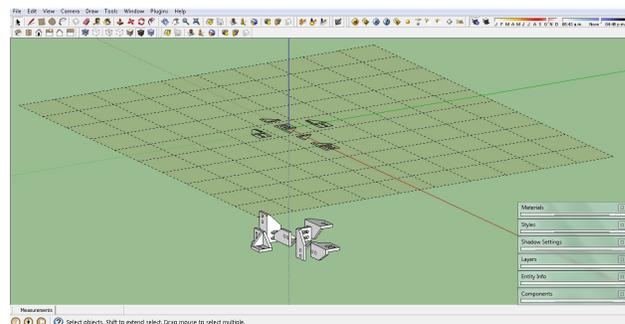


Figura 6. Obtención de la proyección ortogonal.

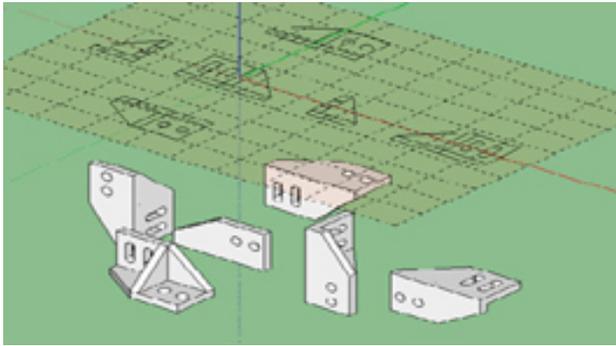


Figura 7 - Detalle Figura 6.

Antes de realizar la exportación se puede analizar la visibilidad, ya que todas las aristas serán proyectadas sobre el plano de trabajo.

A continuación por medio del menú Herramientas -> Exportar a DXF se logra crear un archivo de extensión “.dxf” que abierto desde Draftsight, por ejemplo, muestra el resultado de la Figura 7.

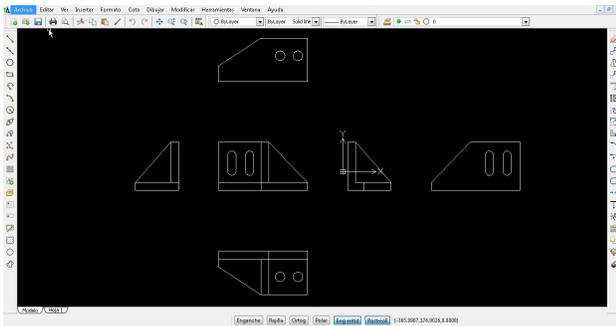


Figura 8. Vistas en Draftsight.

**Otro ejemplo de uso.**

Nos proponemos obtener las vistas del modelo de la Figura 6.

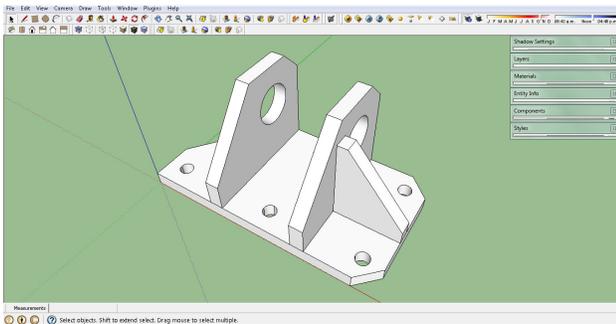


Figura 9. Modelo de prueba.

Una vez ejecutado la extensión tenemos el resultado de la Figura 7. En esta oportunidad elegimos el sistema ISO-A.

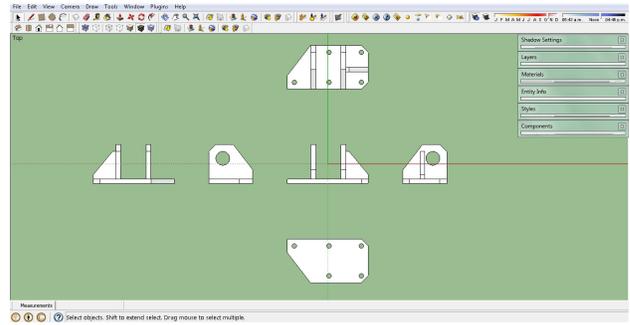


Figura 10. Vistas resultantes en sistema ISO-A.

Haciendo uso de las extensiones, en primer lugar de la extensión Workplane que ejecutamos desde el menú Herramientas -> WorkPlane -> New Plane. Al darle tres puntos esta extensión genera un plano como se observa en la Figura 10.

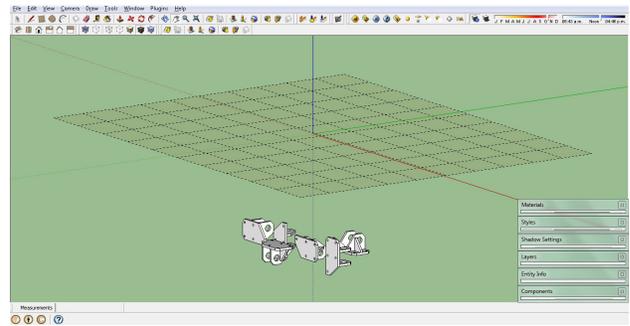


Figura 11. Plano de proyección.

A continuación seleccionamos los modelos y el plano simultáneamente, y desde el menú Plugins -> Flatten to Plane, obtenemos las proyecciones ortogonales como muestra la Figura 9.

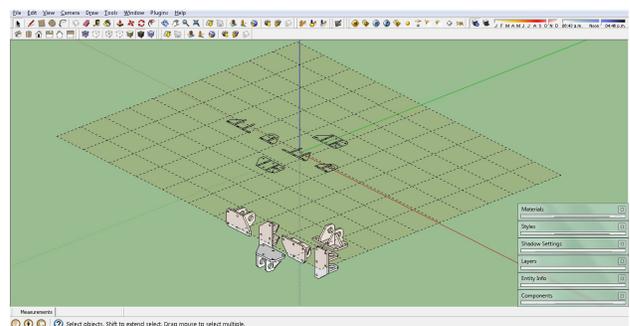


Figura 12. Obtención de la proyección ortogonal.

Repetimos el procedimiento en lo relativo a exportar el resultado por medio de la extensión a un archivo .dxf que luego abrimos desde Draftsight obteniendo lo expuesto en la Figura 10.

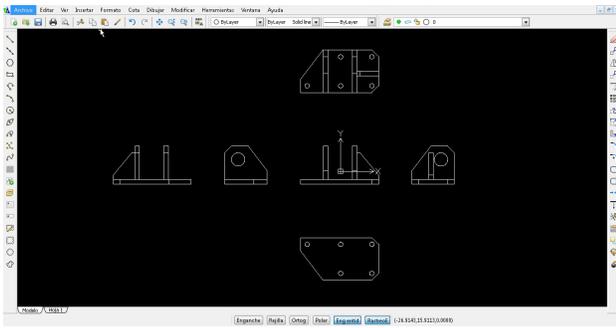


Figura 13. Vistas en Draftsight.

## CONCLUSIONES:

Se concluye que es posible programar con relativa facilidad una extensión que incremente el comportamiento de un software que por su condición de gratuidad posee limitaciones en algunos aspectos. Nuestro caso de éxito lo constituye la extensión "Vistas automáticas" que genera las proyecciones ortogonales de un modelo tridimensional.

Se valoró la contribución de extensiones también gratuitas ya existentes para la realización de nuevas, en nuestro caso en particular, tanto en la fase de desarrollo como la consola para editar código, como en la fase de pos desarrollo, con la extensión que permite exportar el resultado del proceso para visualizarlo en otro programa CAD.

Queda como objetivo en la continuidad de este proyecto la posibilidad de ampliar el funcionamiento de la extensión, para poder optar solo por las tres vistas fundamentales, y obtener perspectivas axonometrías con el correspondiente enriquecimiento de la interfaz de usuario en virtud del incremento de funcionalidad.

## 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, Congreso Egrafía, ISBN N° 978-987-1494-25-5, 203-207.

[2] <http://www.alexschreyer.net/projects/sketchup-ruby-code-editor/>

[3] [www.sketchup.com/intl/en/developer](http://www.sketchup.com/intl/en/developer)

[4] [www.sketchucation.com](http://www.sketchucation.com) y [www.sketchando.net](http://www.sketchando.net)

[5] [www.extensions.sketchup.com](http://www.extensions.sketchup.com)

[6] YUKIHIRO MATSUMOTO. (1993). Guía del usuario Ruby.