



## 3.2 Importación CAD a Gazebo

César Omar Alvarado Contreras

Jonathan Fonseca Camarena

Marcos Manzo Torres

Eduardo Robles Vázquez

Víctor Gabriel Tapia Casillas

Universidad Politécnica de la Zona Metropolitana de Guadalajara

Profesor: Carlos Enrique Morán Garabito

8 de noviembre del 2019

---

# Índice general

---

<b>1</b>	<b>Blender</b>	<b>3</b>
<b>2</b>	<b>Gazebo</b>	<b>4</b>
<b>3</b>	<b>Desarrollo</b>	<b>5</b>
<b>4</b>	<b>conclusión</b>	<b>9</b>
	<b>Bibliografía</b>	<b>10</b>

## Capítulo 1

# Blender

Blender es un programa informático multi plataforma, dedicado especialmente al modelado, iluminación, renderizado, animación y creación de gráficos tridimensionales. También de composición digital utilizando la técnica procesal de nodos, edición de vídeo, escultura (incluye topología dinámica) y pintura digital.

El programa fue inicialmente distribuido de forma gratuita pero sin el código fuente, con un manual disponible para la venta, aunque posteriormente pasó a ser software libre. Actualmente es compatible con todas las versiones de Windows, macOS, GNU/Linux (incluyendo Android), Solaris, FreeBSD e IRIX.

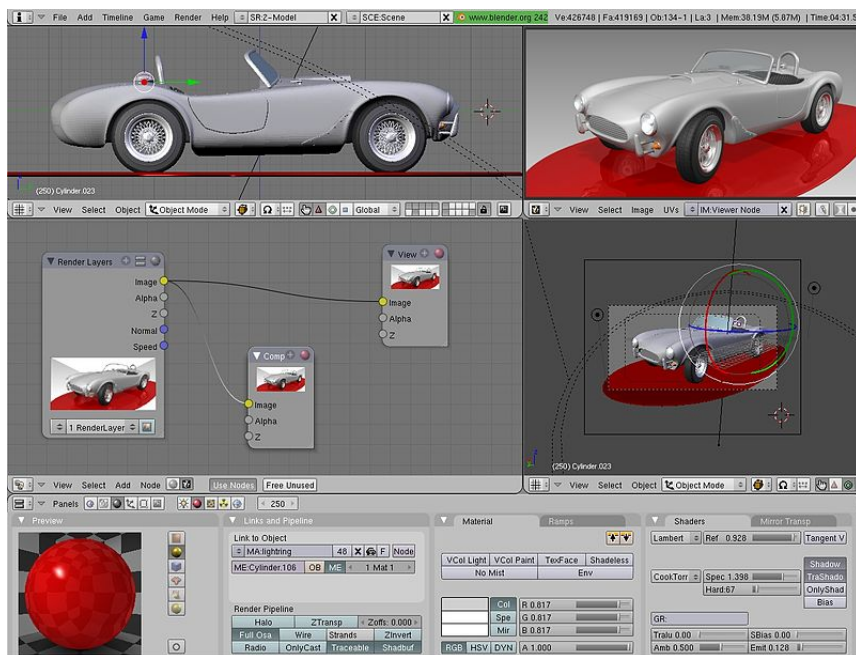


Figura 1.1: Ejemplo de las capacidades de Blender.

:

## Capítulo 2

# Gazebo

Gazebo es un simulador de robótica en 3D de código abierto. Gazebo integró el motor físico ODE de renderizado de OpenGL y código de soporte para la simulación y control de actuadores. En 2011, Gazebo se convirtió en un proyecto independiente apoyado por Willow Garage.

En 2012, la Fundación de robótica de código abierto (OSRF por sus siglas en inglés) se convirtió en el administrador del proyecto Gazebo. Posteriormente, la OSRF cambió su nombre a Open Robotics en 2018.

Gazebo puede utilizar múltiples motores físicos de alto desempeño, como ODE, Bullet, etc (Por defecto viene siendo ODE). Este provee de renderizados realistas de ambientes, incluyendo iluminaciones, sombras y texturas, todas estas de gran calidad. Puede modelar sensores que pueden "ver" el ambiente simulado, como buscadores láser de rango, cámaras y sensores del estilo Kinect entre otros.

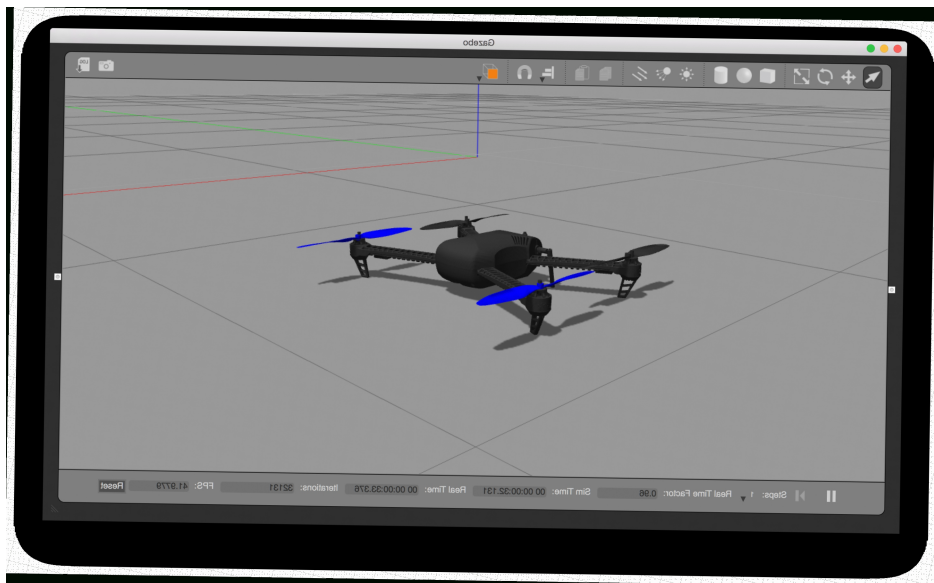


Figura 2.1: Ejemplo de las capacidades de Gazebo.

:

## Capítulo 3

# Desarrollo

Primeramente y partiendo del ensamble realizado en SolidWorks, exportamos dicho ensamble como .STL con lo cual el archivo podrá importarse en Blender de manera sencilla por medio de mallas.

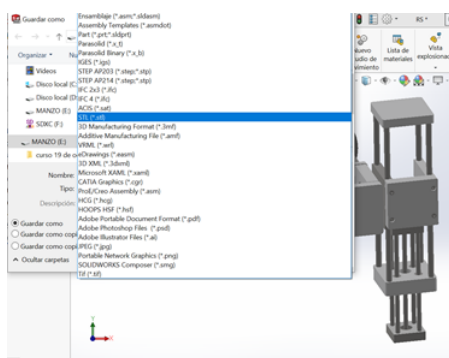


Figura 3.1: Exportación en SolidWorks

Iniciamos nuestra máquina virtual, donde trabajaremos con el archivo .STL

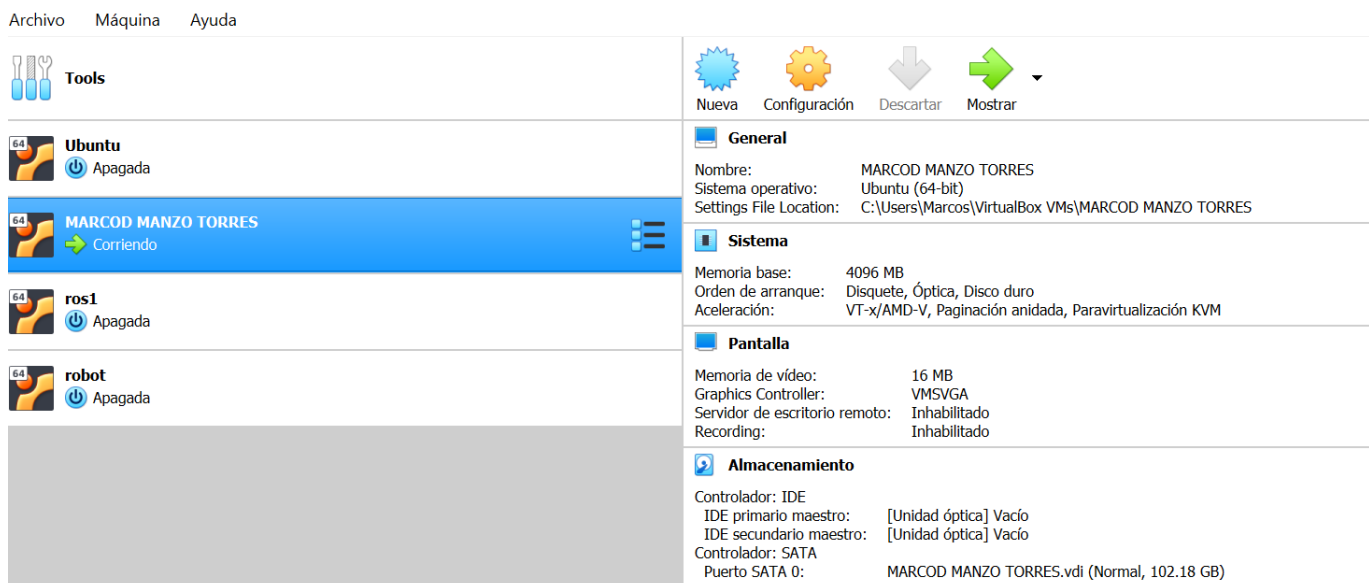


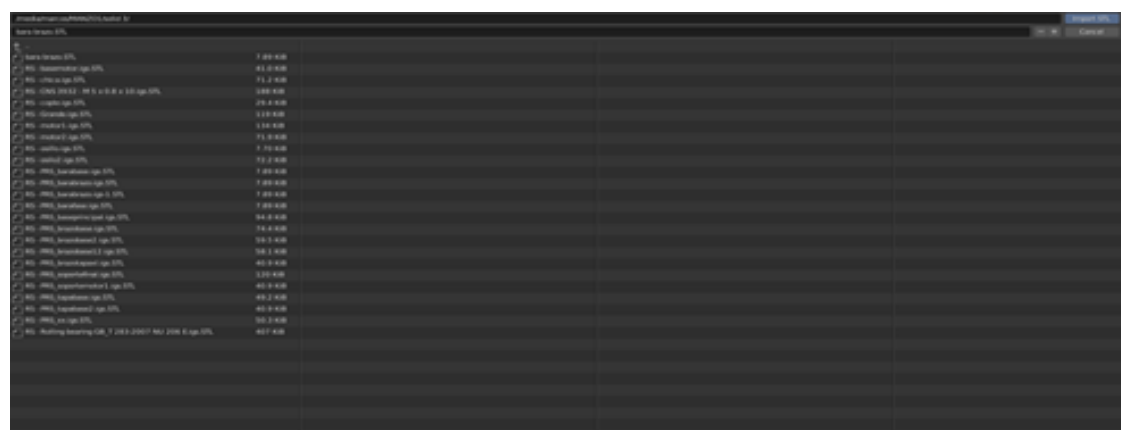
Figura 3.2: Inicio de máquina virtual

A screenshot of a terminal window titled "marcos@marcos-VirtualBox: ~". The terminal shows the execution of several commands and their outputs:  

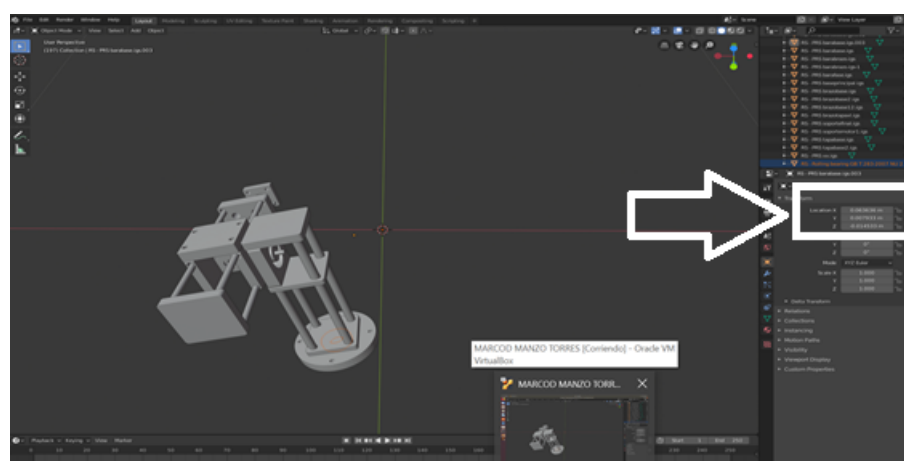
```
marcos@marcos-VirtualBox: ~$ Archive Editor via Secure Terminal - Apache  
Saved session recovery to '/tmp/quitt.blend'  
marcos@marcos-VirtualBox: ~$ blender  
Received X11 Error:  
error code: 179  
request code: 155  
minor code: 34  
error text: GLXBadFBConfig  
Received X11 Error:  
error code: 179  
request code: 155  
minor code: 34  
error text: GLXBadFBConfig  
Received X11 Error:  
error code: 179  
request code: 155  
minor code: 34  
error text: GLXBadFBConfig  
Received X11 Error:  
error code: 179  
request code: 155  
minor code: 34
```

  
At the bottom of the terminal, there are status indicators: "0% CPU" on the left, "CPU: 98%, Memory usage: 16%" in the center, and "40 x 400" on the right.

Tras haber exportado el archivo .STL de SolidWorks, lo importamos en Blender y observamos que el ensamble está subdividido por cada unas de las piezas que lo conforman



Una vez abrimos el ensamble, observamos que nos faltan algunas piezas las cuales se colocan de manera manual por medio de la opción transformar y por medio de coordenadas, se colocan en el lugar deseado.



6

Ahora que finalizamos el ensamble importado en Blender, guardamos el mismo como archivo .DAE y lo colocamos en nuestra carpeta launcher de Gazebo

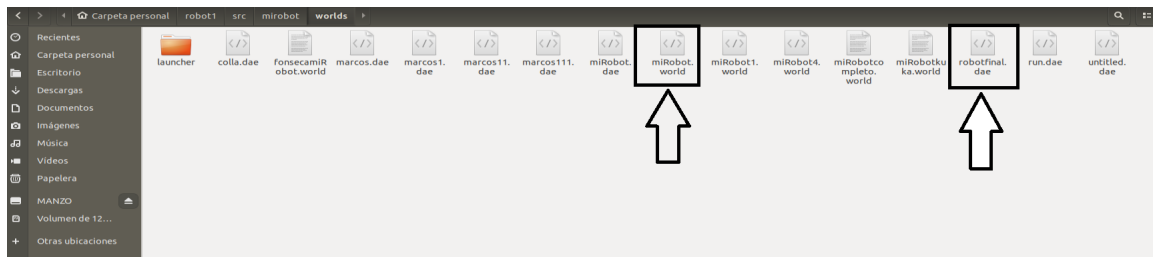


Figura 3.6: Guardar ensamble

Observamos la información almacenada en nuestro archivo .DAE, donde nuestro ensamble fue guardado por medio de mallas similar a un archivo .USDF que es el archivo base de Gazebo. Encontramos cómo se declara el entorno y las uniones para que estas aparezcan a la hora de correr el mismo.

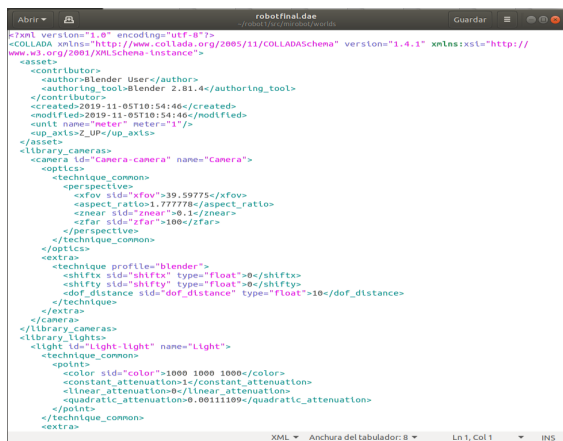


Figura 3.7: Archivo .DAE

Para iniciar el contenido en Gazebo, necesitamos declarar un archivo .World el cual funciona como launcher para iniciar el contenido o arrancar el contenido especificado. solo basta con colocar el nombre del archivo .DAE que deseamos iniciar. Ahora solo basta abrir un terminal, donde iniciaremos gazebo y

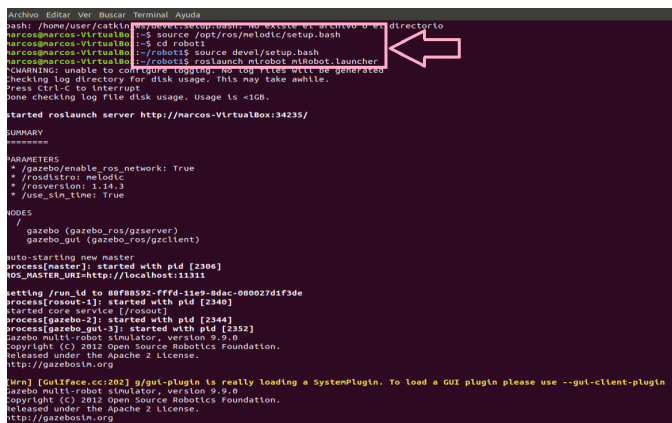


Figura 3.8: Arrancador

nos abrirá nuestro archivo colocado en el arrancador y en nuestra carpeta de trabajo.

```
Archivo Editar Ver Buscar Terminal Ayuda
bash: /home/user/catkin_ws/devel/setup.bash: No existe el archivo o el directorio
marcos@marcos-VirtualBox: ~$ source /opt/ros/melodic/setup.bash
marcos@marcos-VirtualBox: ~$ cd robot1
marcos@marcos-VirtualBox: ~/robot1$ source devel/setup.bash
marcos@marcos-VirtualBox: ~/robot1$ roslaunch mirobot miRobot.launcher
RCWARNING: unable to configure logging. No log files will be generated
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://marcos-VirtualBox:34235/

SUMMARY
=====
PARAMETERS
* /gazebo/enable_ros_network: True
* /roscpp: melodic
* /rosversion: 1.14.3
* /use_sim_time: True

NODES
/
  gazebo (gazebo_ros/gzserver)
  gazebo_gui (gazebo_ros/gzclient)

auto-starting new master
process[master]: started with pid [2306]
ROS_MASTER_URI=http://localhost:11311

setting /run_id to 88f88592-fffd-11e9-8dac-080027d1f3de
process[rosout-1]: started with pid [2340]
started core service [/rosout]
process[gazebo-2]: started with pid [2344]
process[gazebo_gui-3]: started with pid [2352]
gazebo multi-robot simulator, version 9.9.0
Copyright (C) 2012 Open Source Robotics Foundation.
Released under the Apache 2 License.
http://gazebo.org

[Warn] [GuiFace.cc:202] g/gui-plugin is really loading a SystemPlugin. To load a GUI plugin please use --gui-client-plugin
gazebo multi-robot simulator, version 9.9.0
Copyright (C) 2012 Open Source Robotics Foundation.
Released under the Apache 2 License.
http://gazebo.org
```

Figura 3.9: Iniciar Gazebo

Tras haber iniciado Gazebo en nuestra terminal, podemos observar una ventana nueva donde nos aparece el contenido y su entorno, además de poder observar a detalle la simulación de nuestro robot.

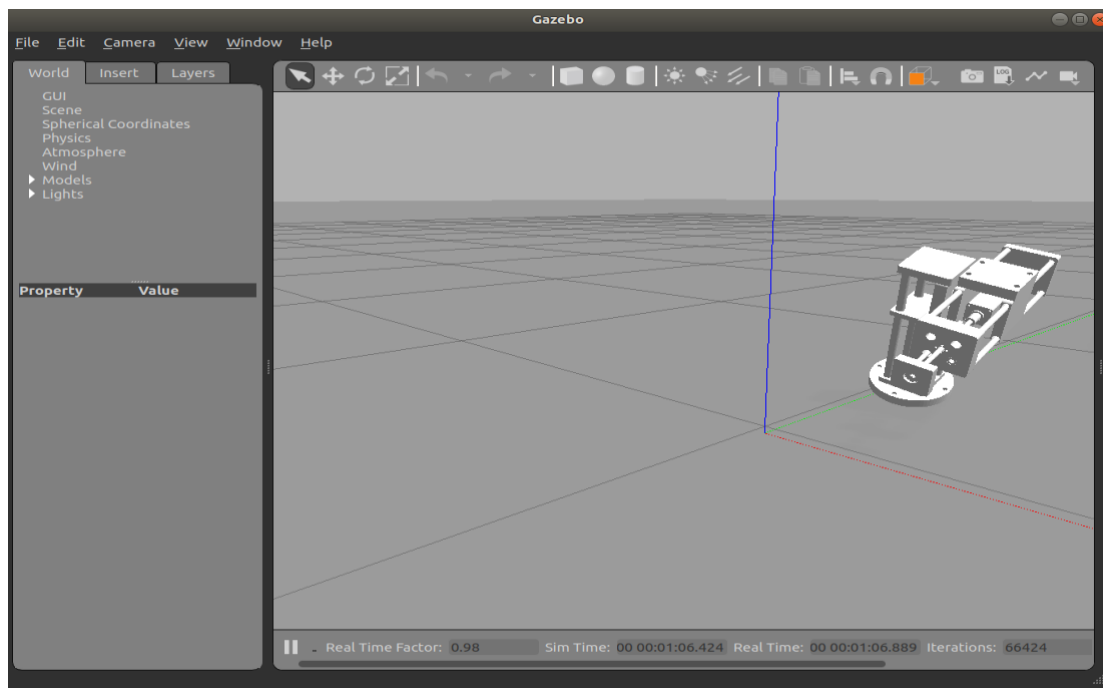


Figura 3.10: Gazebo



## **conclusión**

---

Con esta práctica hemos aprendido a realizar el "traslado" de un modelado tridimensional no solo de un programa a otro, sino que también de un sistema operativo a otro.

Esta herramienta nos será de utilidad en prácticas posteriores, donde será necesario simular el funcionamiento de nuestro robot esférico polar de manera virtual y corroborar su correcto funcionamiento sin el temor de algún desastre en el modelo físico, esto de la mano con ROS, para posteriormente poder aplicar nuestra programación en el modelo físico final.

---

# Bibliografía

---

- [1] Lentin Joseph. Gazebo y su entorno de trabajo. In *Robot Operating System (ROS) for Absolute Beginners*, pages 127–170. Springer, 2018.
- [2] Jonathan Ruiz de Garibay Pascual. Solidworks to blender, export. *Universidad de Deuston. Número. Fecha*, page 54, 2006.