



**Universidad de Panamá**  
**Facultad de Informática, Electrónica y Comunicación**  
**Escuela de Ingeniería en Informática**

**Asignatura**  
**Sistemas Expertos**

**Integrantes:**

**Jorge Ruiz 6-723-428**  
**Erika Gonzalez 8-954-2161**  
**Milagros Maturana 8-966-1319**  
**Zaydeth Lewys 8-946-431**  
**Gilberto Muñoz 8-960-1326**

**Profesor**  
**Ariel Vernaza**

**Proyecto Final**  
**Gundam 2022**

## INDICE

|   |    |
|---|----|
| Investigaciones Preliminares - GUNDAM RX78 .....              | 3  |
| Historia .....  | 3  |
| Connotación en la Cultura Japonesa.....                       | 3  |
| Gundam Challenge .....  | 3  |
| Objetivo del Proyecto .....                                   | 4  |
| Bases Del Proyecto .....                                      | 4  |
| Creación de Sprint Backlog .....                              | 4  |
| Definición de Stakeholders.....                               | 4  |
| Implementación del Código Base y Desarrollo del Proyecto..... | 5  |
| Paquetes de ROS para robots de GUNDAM.....                    | 5  |
| Configuración del Ambiente de Trabajo .....                   | 6  |
| Instalación de Mesh y URDF .....                              | 6  |
| Visualizar modelo URDF .....                                  | 7  |
| Mapa de Panamá.....   | 8  |
| Ejecución del Gazebo .....                                    | 8  |
| Para controlar los ángulos de las articulaciones .....        | 9  |
| Simulación: .....   | 10 |
| Comandos: .....   | 11 |
| joint_trajectory_client_csv.py .....                          | 11 |
| Video de YouTube:.....  | 11 |
| Referencias .....   | 12 |

## **Investigaciones Preliminares - GUNDAM RX78**

### **Historia**

El RX-78-2 Gundam (en japonés: RX-78-2 ガンダム, Hepburn: Āru Ekkusu Nanajū Hachi no Ni Gandamu) es un robot (mecha) tripulado ficticio, introducido en 1979 en la serie de anime Mobile Suit Gundam de Yoshiyuki Tomino y Sunrise. [1]

El RX-78-2 fue el segundo de los tres mobile suits prototipos Gundam contruidos en Side 7 en el año UC 0079, como parte del Proyecto V de la Federación Terrestre. Construido para complementar a la unidad de largo alcance Guntank y la unidad de apoyo de alcance medio Guncannon, el Gundam fue diseñado para ser una máquina de corto alcance, altamente móvil con avanzado armamento más allá de los usados por las maquinas Zeón. [2]

### **Connotación en la Cultura Japonesa**

Se han realizado investigaciones que revelan cómo el personaje RX-78-2 Gundam se presenta como un héroe en el contexto de la cultura japonesa presentada en el anime Mobile Suit Gundam.

Los resultados de este estudio son que el personaje RX-78-2 es una encarnación o una representación de un héroe japonés del pasado, a saber, el samurái que fue revivido por el director y el diseñador en una historia ficticia en forma de un robot gigante; en el que hay principios de vida que hasta el día de hoy los japoneses siguen. Además, hace connotación al espíritu de un samurái o lo que se llama el espíritu de Bushido para seguir luchando incansablemente en defensa de la justicia, principio que se muestra en la figura del personaje ficticio RX -78-2 Gundam como héroe de Japón. [3]

### **Gundam Challenge**

Durante el año 2009 apareció una estatua de GUNDAM de tamaño natural. La vista de la estatua con los pies firmemente plantados en el suelo capturó los corazones de muchas personas y representó la posibilidad de que los sueños se hicieran realidad.

En el año 2019, GUNDAM, que celebra su 40 aniversario conmoverá Japón y conmoverá al mundo.

Se creó un desafío para mover el GUNDAM de tamaño real de 18 m de altura.

### **Objetivo del Proyecto**

El objetivo es encontrar una forma que permita a Gundam tener un combate, manteniendo la estabilidad de este.

### **Bases Del Proyecto**

Sistema Operativo: Linux, Ubuntu, Sistema Operativo Robótico.

Robot Modelo: [https://github.com/gundam-global-challenge/gundam\\_robot](https://github.com/gundam-global-challenge/gundam_robot)

Referencias: [https://github.com/HumaRobotics/darwin\\_gazebo](https://github.com/HumaRobotics/darwin_gazebo)

### **Creación de Sprint Backlog**

Como siguiente punto desarrollamos las tareas de las que estaría compuesta el Backlog.

Backlog:

1. Estudiar la información provista por el Stakeholder Ariel Vernaza
2. Realizar Investigación Preliminar
3. Definir Metodología de Trabajo
4. Configurar el ambiente de trabajo
5. Desarrollar el Código de Programación para ROS
6. Probar el código en una Simulación
7. Grabar un video de la simulación de GUNDAM RX78
8. Editar el video de la simulación
9. Subir a YouTube el video de la simulación
10. Crear el Reporte de Avances del Semestral
11. Subir al Github el código de programación.

### **Definición de Stakeholders**

Además, definimos nuestros Stakeholders.

Stakeholder Principal: Profesor de Sistemas Expertos Ariel Vernaza

Stakeholder Secundarios:

- Departameto de informatica,
- Facultad de Informatica, Electronica y comunicación.
- Universidad de Panamá.

Una vez de acuerdo con la metodología comenzamos a desarrollar el backlog, leyendo toda la información provista por nuestro Stakeholder Principal Ariel Vernaza.

Video Adjunto por el professor:

<https://www.youtube.com/watch?v=XIucUB23zZ4>

Ahora que hemos terminado con la lectura procedemos a realizar la investigación preliminar que se puede observar en este informe.

Procedemos a configurar el ambiente de trabajo Gazebo y a ejecutar el simulador como se puede observar en la sección de desarrollo del proyecto.

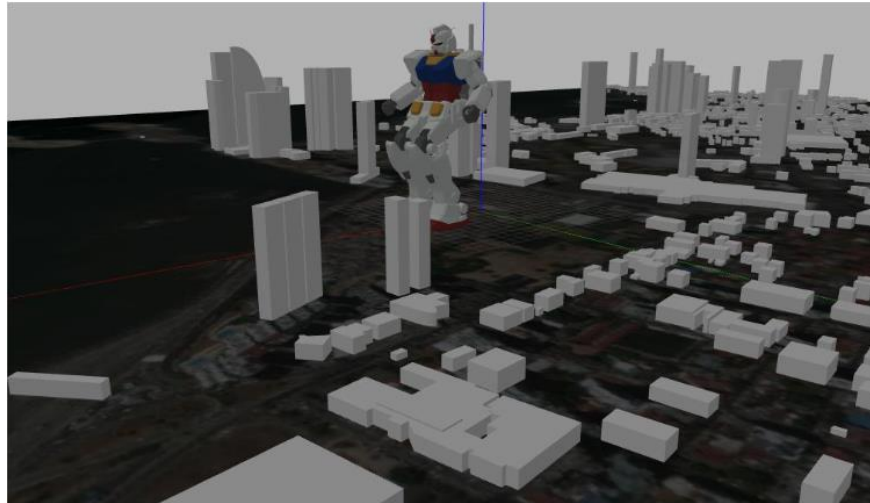
## **Implementación del Código Base y Desarrollo del Proyecto**

El sistema operativo utilizado es Ubuntu 18.04 Es necesario tener ROS instalado, con el entorno de trabajo configurado. Los pasos para la instalación y configuración pueden verse en:

<http://wiki.ros.org/melodic/Installation/Ubuntu>

### **Paquetes de ROS para robots de GUNDAM**

original: [https://github.com/gundam-global-challenge/gundam\\_robot.git](https://github.com/gundam-global-challenge/gundam_robot.git)



## Configuración del Ambiente de Trabajo

```
$ mkdir -p catkin_ws/src
$ cd catkin_ws
$ wstool init src
$ wstool merge -t src https://raw.githubusercontent.com/gundam-global-challenge/gundam_robot/.gundam.rosinstall
$ wstool update -t src
$ source /opt/ros/$ROS_DISTRO/setup.bash
$ rosdep install -y -r --from-paths src --ignore-src
$ catkin build
$ source devel/setup.bash
```

## Instalación de Mesh y URDF

Primero, descargamos el archivo Gundam Collada

(p. ej., GGC\_TestModel\_rx78\_20170112.DAE) en el directorio gundam\_rx78\_description.

Luego, ejecutamos el archivo ./scripts/dae\_to\_urdf.py con el nombre del archivo descargado como argumento. Esto creó archivos de malla en el directorio meshes/ y creó el archivo URDF en el directorio urdf/.

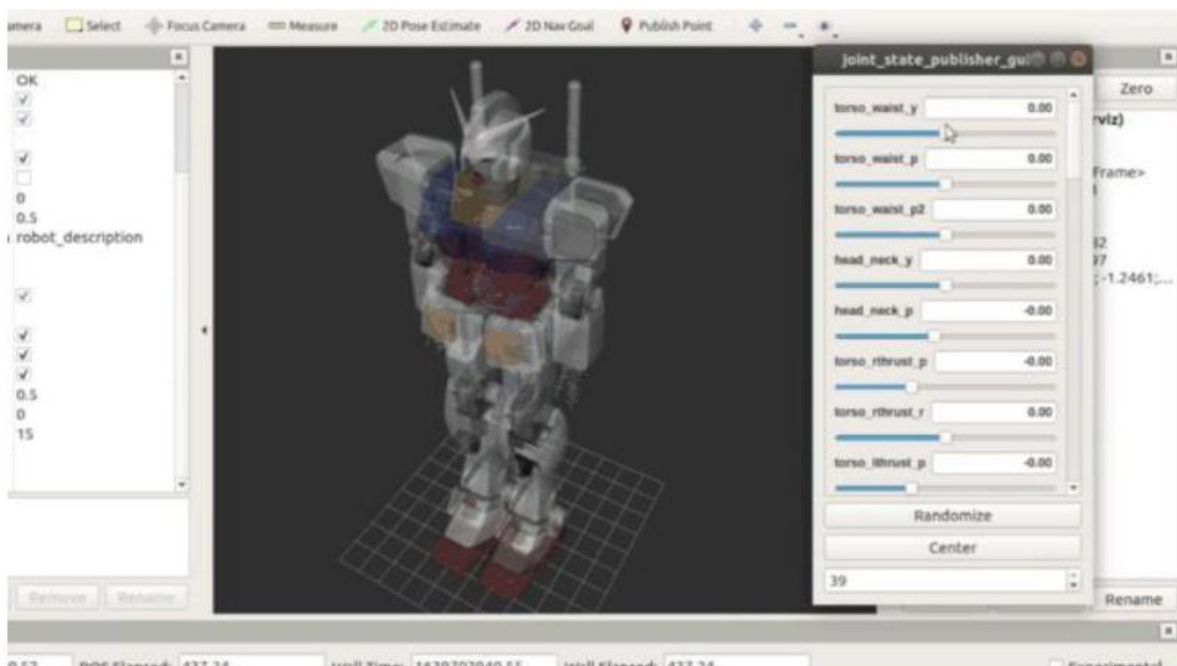
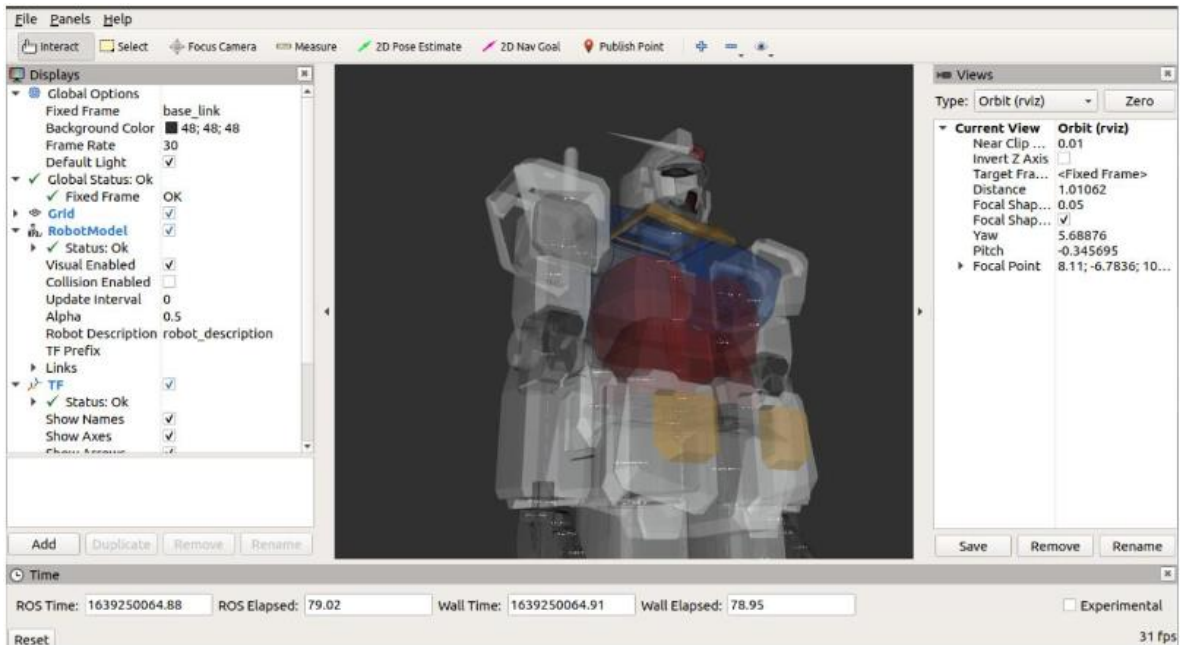
```
$ roscd gundam_rx78_description
$ python ./scripts/ggc_dae_to_urdf.py GGC_TestModel_rx78_20170112.DAE
--
write_mesh
```

```
$ mv urdf/GGC_TestModel_rx78_20170112.urdf urdf/gundam_rx78.urdf
```

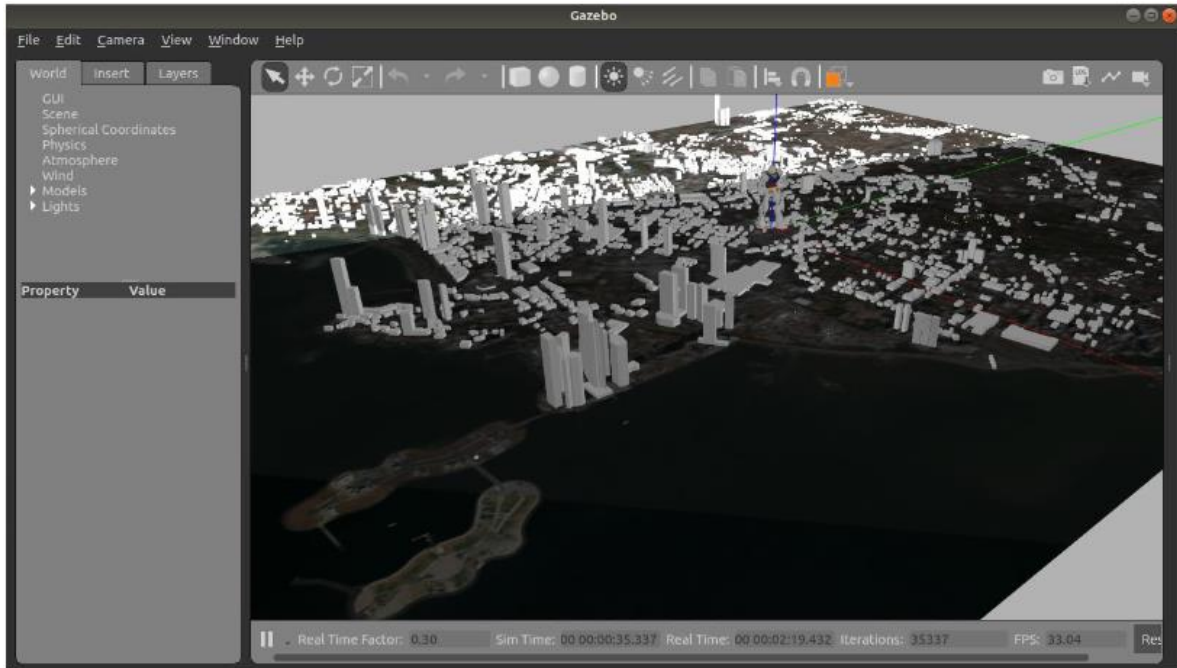
## Visualizar modelo URDF

Archivo URDF en rviz, puede usar el archivo display.launch.

```
$ roslaunch gundam_rx78_description display.launch
```



## Mapa de Panamá



## Ejecución del Gazebo

En esta sección nos disponemos a instalar el ambiente de trabajo Gazebo en nuestro Sistema operativo a través de la terminal.

```
$ roslaunch gundam_rx78_gazebo gundam_rx78_world.launch
```

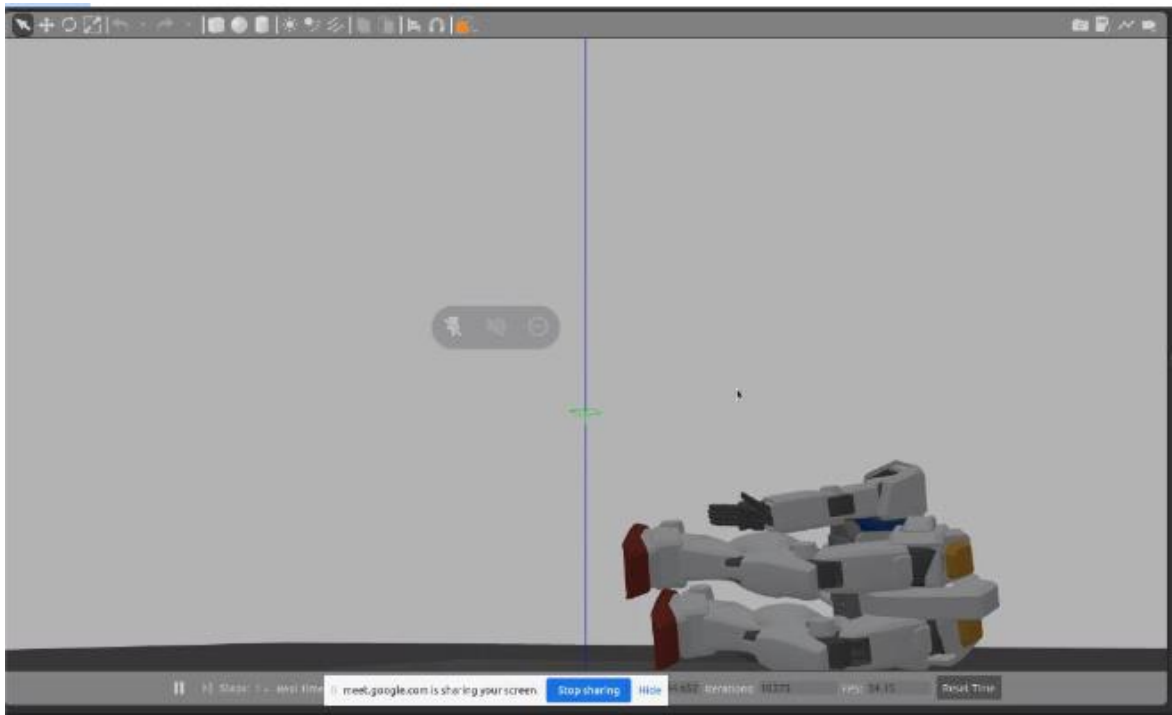




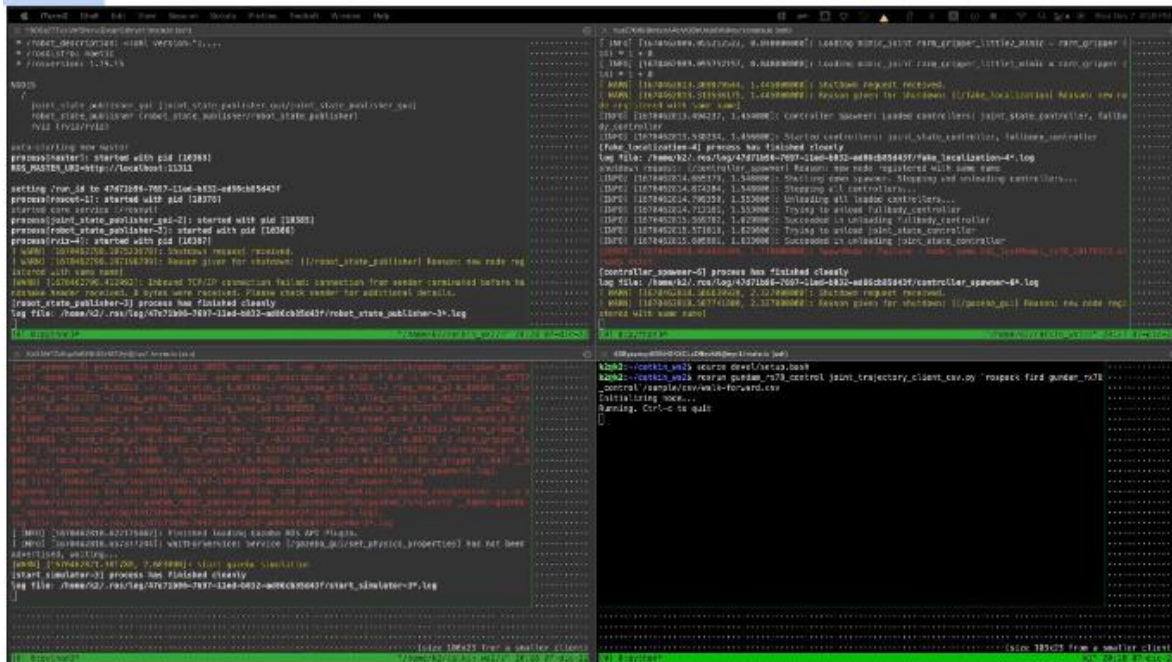
```
$ rosrund gundam_rx78_control joint_trajectory_client_csv.py `rospack find  
gundam_rx78_control`/sample/csv/turn-right.csv  
# Girar a la izquierda  
$ rosrund gundam_rx78_control joint_trajectory_client_csv.py `rospack find  
gundam_rx78_control`/sample/csv/turn-left.csv  
# Levantar los brazos y caminar  
$ rosrund gundam_rx78_control joint_trajectory_client_csv.py `rospack find  
gundam_rx78_control`/sample/csv/up.csv
```

Realizamos las primeras pruebas con el robot. Pero el gran reto que se tuvo fue evitar que el robot cayese al vacío, apenas se abre el entorno.

### Simulación:



## Comandos:



Tengamos en cuenta que actualmente tenemos varias limitaciones en esta simulación, solo tenemos un controlador de posición etc.

También se pueden encontrar archivos de control de movimiento de muestra en el directorio `gundam_rx78_control/sample`.

**joint\_trajectory\_client\_csv.py**

Usando el control de gundam con joint\_trajectory\_client.csv.py e ingresando un archivo .csv

Este utilizará los patrones indicados en el archivo para simular el movimiento del gundam según los valores de los ángulos de cada componente.

```
# mover la parte superior del cuerpo
```

```
$ rosrn gundam_rx78_control joint_trajectory_client_example.py
```

**Video de YouTube:**

<https://youtu.be/-TdqdOPxIps>

## **Referencias**

- [1] «Wikipedia.com,» 9 Noviembre 2022. [En línea]. Available: [https://en.wikipedia.org/wiki/Gundam\\_\(fictional\\_robot\)](https://en.wikipedia.org/wiki/Gundam_(fictional_robot)). [Último acceso: 2022 Diciembre 2022].
- [2] G. Fandom, «gundam.fandom.com,» [En línea]. Available: [https://gundam.fandom.com/es/wiki/RX-78-2\\_Gundam#Historia](https://gundam.fandom.com/es/wiki/RX-78-2_Gundam#Historia). [Último acceso: 2022].
- [3] W. S. Pramono, «Kajian Semiotika Gundam RX-78-2 Sebagai Hero Dalam Anime Mobile Suit Gundam (1979-1980,» de UPT Perpustakaan ISI Yogyakarta, 2017.