

Configuración del Entorno con Docker, ROS2 y Gazebo

Paula Sandoval

Abril 2025

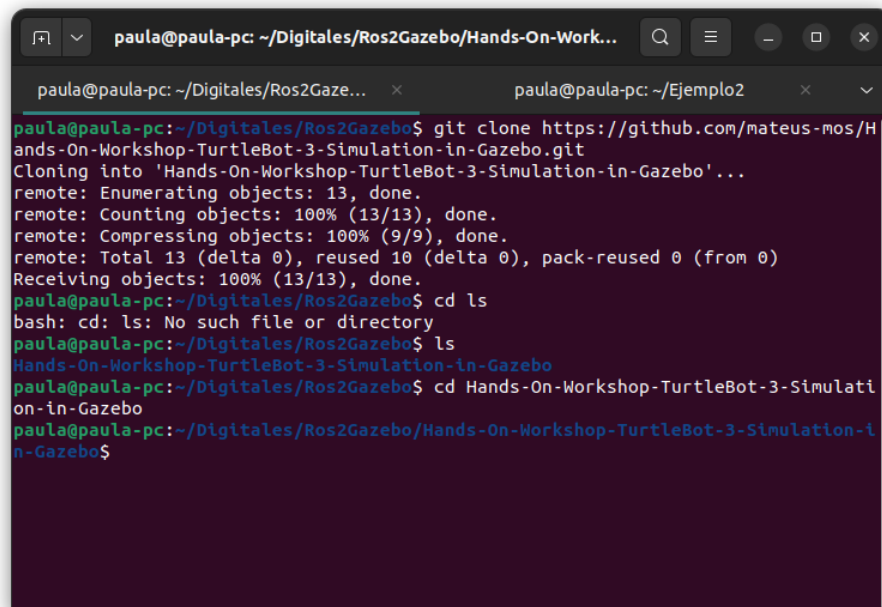
Este documento muestra cómo configurar un entorno para simulaciones robóticas utilizando Docker, ROS2 y Gazebo. Este tutorial muestra la instalación de las herramientas necesarias y la ejecución de la configuración del entorno para visualizar simulaciones de diferentes robots incluidos en Gazebo. Basado en el repositorio de mateus-mos:

Repositorio Hands-On-Workshop-TurtleBot-3-Simulation-in-Gazebo

Pasos de instalación

1. Clonar el Repositorio del Proyecto

Inicialmente se crea una carpeta para el proyecto, al ingresar al directorio elegido, de ejecuta el comando de **Git clone** junto a la dirección del repositorio desde la terminal:



```
paula@paula-pc: ~/Digitales/Ros2Gazebo/Hands-On-Work...
paula@paula-pc: ~/Digitales/Ros2Gazebo$ git clone https://github.com/mateus-mos/Hands-On-Workshop-TurtleBot-3-Simulation-in-Gazebo.git
Cloning into 'Hands-On-Workshop-TurtleBot-3-Simulation-in-Gazebo'...
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 13 (delta 0), reused 10 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (13/13), done.
paula@paula-pc:~/Digitales/Ros2Gazebo$ cd ls
bash: cd: ls: No such file or directory
paula@paula-pc:~/Digitales/Ros2Gazebo$ ls
Hands-On-Workshop-TurtleBot-3-Simulation-in-Gazebo
paula@paula-pc:~/Digitales/Ros2Gazebo$ cd Hands-On-Workshop-TurtleBot-3-Simulation-in-Gazebo
paula@paula-pc:~/Digitales/Ros2Gazebo/Hands-On-Workshop-TurtleBot-3-Simulation-in-Gazebo$
```

Figura 1: Enter Caption

2. Instalar Docker Engine

A continuación, se debe instalar Docker Engine. Este paso es opcional ya que es este caso ya se contraba instalo en la maquina de manera anticipada. Se puede ejecutar el contenedor Hello world para confirmar la correcta instalación de la herramienta.

Repositorio Instalación de docker para ubuntu

<https://docs.docker.com/engine/install/ubuntu/>

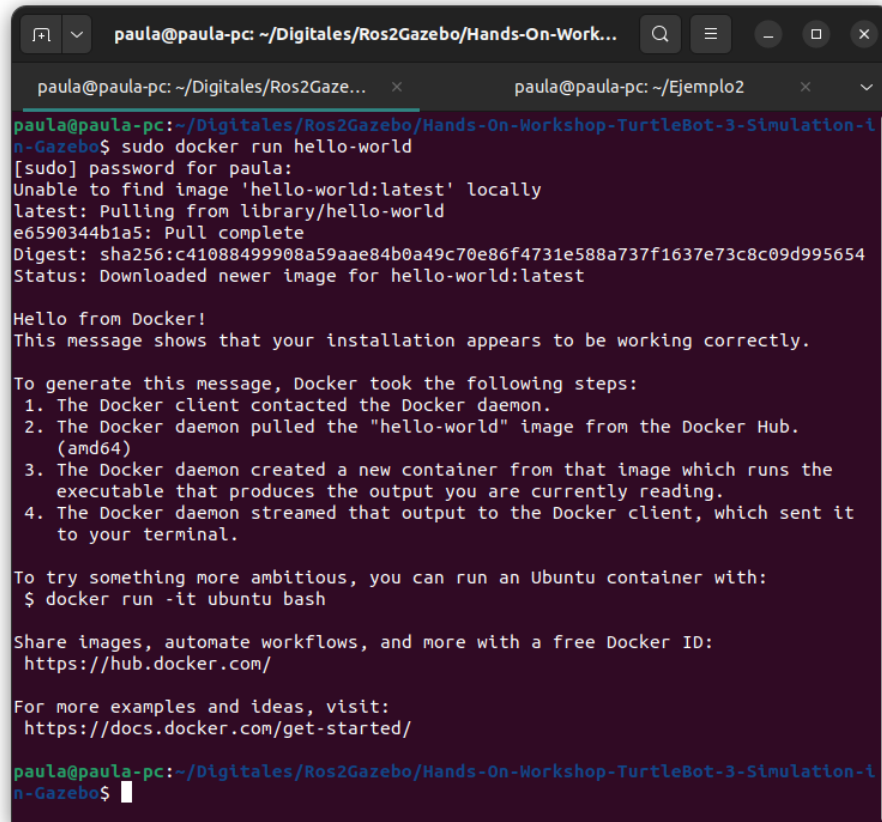
A screenshot of a terminal window with a dark background. The window title is 'paula@paula-pc: ~/Digitales/Ros2Gazebo/Hands-On-Workshop-TurtleBot-3-Simulation-1'. The terminal shows the command 'sudo docker run hello-world' being executed. The output includes a password prompt, a message about finding the image locally, the pulling of the 'hello-world:latest' image from Docker Hub, the completion of the pull, the digest, and the status. The output then displays 'Hello from Docker!' and a message stating that the installation appears to be working correctly. It lists the steps Docker took to generate the message: 1. The Docker client contacted the Docker daemon. 2. The Docker daemon pulled the 'hello-world' image from the Docker Hub. (amd64) 3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading. 4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal. It then suggests trying something more ambitious by running an Ubuntu container with 'docker run -it ubuntu bash'. Finally, it provides links to share images and for more examples. The prompt returns to 'paula@paula-pc: ~/Digitales/Ros2Gazebo/Hands-On-Workshop-TurtleBot-3-Simulation-1'.

Figura 2: Docker - Hello world

3. Crear el Grupo Docker

Crear un grupo `docker` y agregar a los usuarios, para facilitar el uso de comandos sin necesidad de privilegios `sudo`. **Repositorio** Instalación de docker group

4. Instalar Docker Compose

El siguiente paso en la instalación es el Docker Compose, es una herramienta necesaria para manejar aplicaciones Docker con múltiples contenedores.

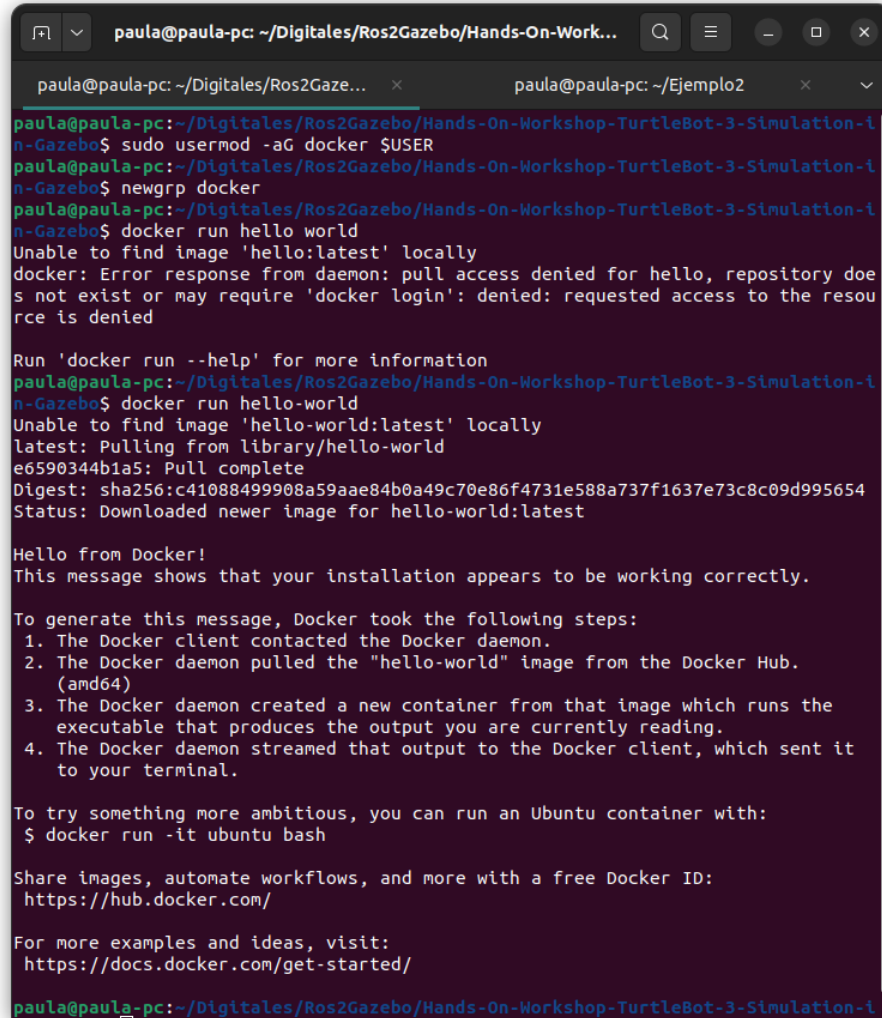
A screenshot of a terminal window with a dark background and light-colored text. The window title bar shows the user 'paula' on a machine named 'paula-pc' in the directory '~/Digitales/Ros2Gazebo/Hands-On-Work...'. The terminal shows the following commands and output:
1. `sudo usermod -aG docker $USER`
2. `newgrp docker`
3. `docker run hello world`
 Output: `Unable to find image 'hello:latest' locally`
 Output: `docker: Error response from daemon: pull access denied for hello, repository does not exist or may require 'docker login': denied: requested access to the resource is denied`
4. `docker run --help` (for more information)
5. `docker run hello-world`
 Output: `Unable to find image 'hello-world:latest' locally`
 Output: `latest: Pulling from library/hello-world`
 Output: `e6590344b1a5: Pull complete`
 Output: `Digest: sha256:c41088499908a59aae84b0a49c70e86f4731e588a737f1637e73c8c09d995654`
 Output: `Status: Downloaded newer image for hello-world:latest`
6. Output: `Hello from Docker!`
 Output: `This message shows that your installation appears to be working correctly.`
7. Text: `To generate this message, Docker took the following steps:`
 List:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub. (amd64)
 3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.
8. Text: `To try something more ambitious, you can run an Ubuntu container with:`
 Command: `$ docker run -it ubuntu bash`
9. Text: `Share images, automate workflows, and more with a free Docker ID:`
 URL: `https://hub.docker.com/`
10. Text: `For more examples and ideas, visit:`
 URL: `https://docs.docker.com/get-started/`
The terminal ends with the prompt `paula@paula-pc:~/Digitales/Ros2Gazebo/Hands-On-Workshop-TurtleBot-3-Simulation-i`.

Figura 3: Docker group

5. Construir y Ejecutar el Contenedor Docker

Tras instalar Docker y Docker Compose, se debe construir el contenedor y ejecutarlo, evidenciando la ejecución de ROS2 y Gazebo para este caso.

* Para versiones recientes se debe hacer uso de docker compose, en cambio de docker-compose.

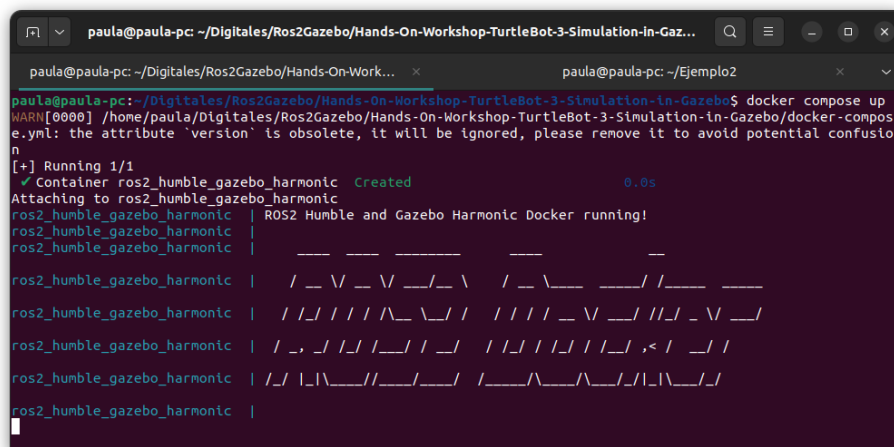
```
paula@paula-pc: ~/Digitales/Ros2Gazebo/Hands-On-Work...
paula@paula-pc: ~/Digitales/Ros2Gazebo/Hands-On-Workshop-TurtleBot-3-Simulation-in-Gazebo$ sudo apt-get install docker-compose
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
docker-compose is already the newest version (1.29.2-1).
The following packages were automatically installed and are no longer required:
  linux-headers-6.2.0-26-generic linux-hwe-6.2-headers-6.2.0-26
  linux-image-6.2.0-26-generic linux-modules-6.2.0-26-generic
  linux-modules-extra-6.2.0-26-generic
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
paula@paula-pc:~/Digitales/Ros2Gazebo/Hands-On-Workshop-TurtleBot-3-Simulation-in-Gazebo$
```

Figura 4: Docker Compose

```
paula@paula-pc: ~/Digitales/Ros2Gazebo/Hands-On-Work...
paula@paula-pc: ~/Digitales/Ros2Gazebo/Hands-On-Workshop-TurtleBot-3-Simulation-in-Gazebo$ docker compose build
WARN[0000] /home/paula/Digitales/Ros2Gazebo/Hands-On-Workshop-TurtleBot-3-Simulation-in-Gazebo/docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
Compose can now delegate builds to bake for better performance.
To do so, set COMPOSE_BAKE=true.
[+] Building 153.2s (5/16)                                docker:default
[+] Building 153.4s (5/16)                                docker:default
=> => transferring context: 2B                             0.0s
=> [ros2_humble_gazebo 1/13] FROM docker.io/library/ros:humble@sha256: 58.5s
=> => resolve docker.io/library/ros:humble@sha256:d62c38c84a37900f373d1c 0.0s
=> => sha256:e3cbd214e98aed196ef5d3b4e323c10af6bf5342ef1 3.10kB / 3.10kB 0.0s
=> => sha256:d62c38c84a37900f373d1cb3a8cb222c0ccdd67ede5 2.69kB / 2.69kB 0.0s
=> => sha256:f3c5d1f659ac7727135201b389078359ae7767d2d04 5.59kB / 5.59kB 0.0s
=> => sha256:9a7e736ce0baf9d51b44d5ead6000c34fa440489dcf 1.21MB / 1.21MB 5.6s
=> => sha256:fc958827a8b499b348349b560b1c2798dda767c9420 3.63MB / 3.63MB 5.3s
=> => sha256:30a9c22ae099393b0131322d7f50d8a9d7cd06c5 29.53MB / 29.53MB 14.6s
=> => sha256:c0be3271a91fea9ec7a9ed9e2e17342524d3c263ee0 2.00kB / 2.00kB 5.6s
```

Figura 5: instalación Ros2 y gazebo

6. Acceder al Contenedor en Ejecución

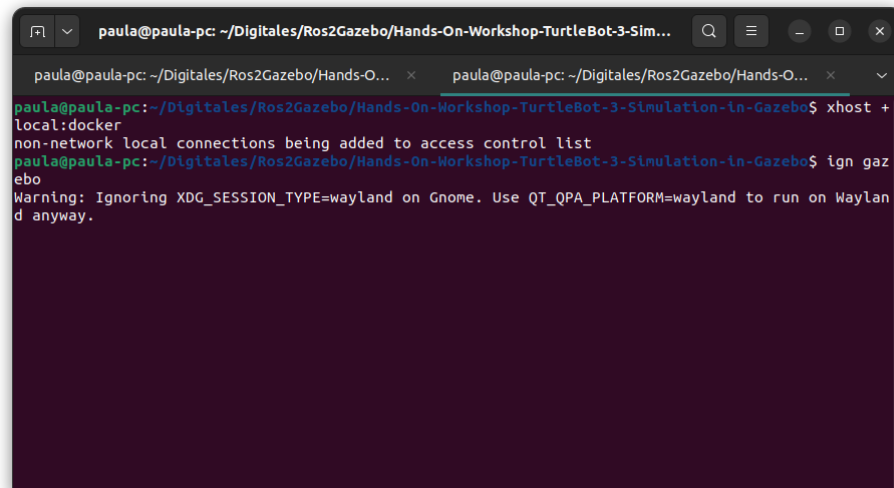


```
paula@paula-pc: ~/Digitales/Ros2Gazebo/Hands-On-Workshop-TurtleBot-3-Simulation-in-Gazebo$ docker compose up
WARN[0000] /home/paula/Digitales/Ros2Gazebo/Hands-On-Workshop-TurtleBot-3-Simulation-in-Gazebo/docker-compos
e.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusio
n
[+] Running 1/1
✓ Container ros2_humble_gazebo_harmonic Created 0.0s
Attaching to ros2_humble_gazebo_harmonic
ros2_humble_gazebo_harmonic | ROS2 Humble and Gazebo Harmonic Docker running!
ros2_humble_gazebo_harmonic |
ros2_humble_gazebo_harmonic |  _ _ _ _ _
ros2_humble_gazebo_harmonic | / _ \ _ \ _ \ _ \ / _ \ _ \ _ \ _ \
ros2_humble_gazebo_harmonic | / / / / / \ \ \ \ \ / / / _ \ _ \ / / _ \
ros2_humble_gazebo_harmonic | / _ \ / / / / / _ \ / / / / / _ \ / _ \
ros2_humble_gazebo_harmonic | / / _ \ \ \ \ \ \ / _ \ \ \ \ \ \ / _ \
ros2_humble_gazebo_harmonic |
```

Figura 6: Ros 2

7. Habilitar la Interfaz Visual para Gazebo

Para propósitos de simulación, se configura el acceso a la interfaz visual del host desde dentro del contenedor Docker, esencial para el funcionamiento de Gazebo.



```
paula@paula-pc: ~/Digitales/Ros2Gazebo/Hands-On-Workshop-TurtleBot-3-Simulation-in-Gazebo$ xhost +
local:docker
non-network local connections being added to access control list
paula@paula-pc: ~/Digitales/Ros2Gazebo/Hands-On-Workshop-TurtleBot-3-Simulation-in-Gazebo$ ign gazebo
Warning: Ignoring XDG_SESSION_TYPE=wayland on Gnome. Use QT_QPA_PLATFORM=wayland to run on Wayland anyway.
```

Figura 7: Acceso a la interfaz de Gazebo

8. Lanzar Gazebo para Simulaciones

Finalmente, se muestra cómo lanzar Gazebo dentro del contenedor y cómo confirmar una configuración exitosa al ejecutar una simulación de robot.

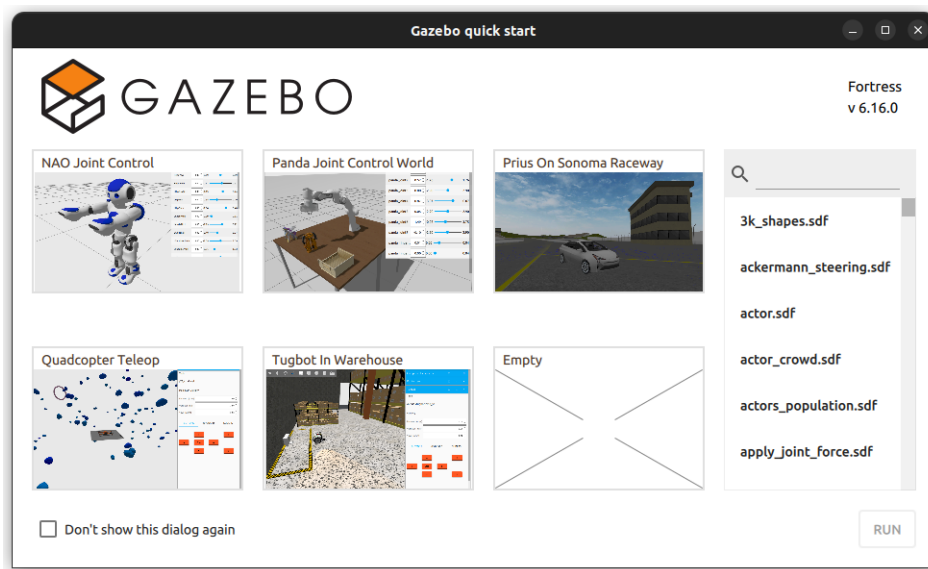


Figura 8: Gazebo

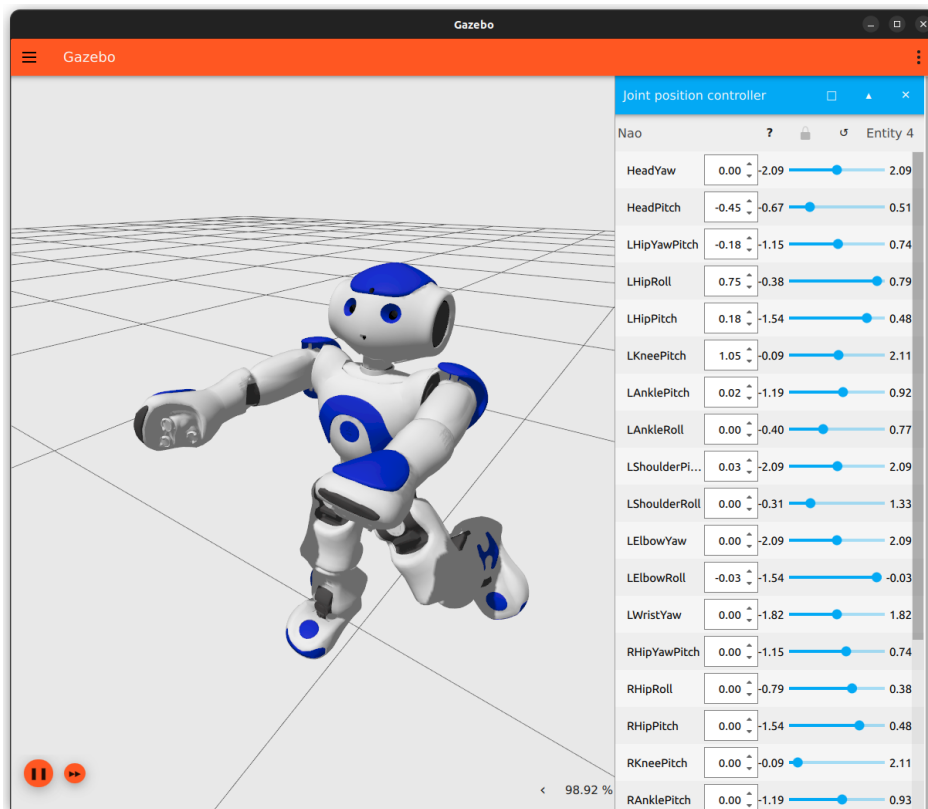


Figura 9: NAO - Pepper Simulation