

# Cheatsheet: la CLI de Gazebo Jetty (comando gz)

Referencia rápida de **Robótica Fácil con ROS 2** — ROS 2 Lyrical Luth · Gazebo Jetty (gz-sim 10) · Ubuntu Resolute 26.04

## Antes de empezar

El binario gz no está en el PATH del sistema hasta que sourceas el entorno de ROS 2 en cada terminal nueva:

```
source /opt/ros/lyrical/setup.bash
```

Si no, verás gz: command not found.

## gz sim — arrancar y gestionar el simulador

Servidor y GUI de Gazebo. Sustituye <mundo> por la ruta a tu archivo .sdf.

```
gz sim -r --render-engine ogre mundo.sdf
# abre un mundo y arranca la física ya; motor ligero (ideal en WSL2)
gz sim -s mundo.sdf
# solo servidor, sin GUI (headless / CI)
gz sim --versions
# versión instalada (10.4.0 en Jetty)
gz sim --help
# ayuda completa del comando
```

## gz service — listar e invocar servicios

El mecanismo de petición/respuesta de Gazebo (independiente de los servicios de ROS 2).

```
gz service -l
# listar todos los servicios disponibles
gz service -i -s /world/<mundo>/control
# inspeccionar el tipo de petición/respuesta de un servicio
gz service -s /world/<mundo>/control \
--reqtype gz.msgs.WorldControl --reptype gz.msgs.Boolean \
--timeout 2000 --req 'pause: true'
# pausar la simulación (pause: false para reanudar)
gz service -s /world/<mundo>/set_physics \
--reqtype gz.msgs.Physics --reptype gz.msgs.Boolean \
--timeout 2000 --req 'real_time_factor: 2.0'
# cambiar la velocidad de simulación
```

## gz topic — listar e inspeccionar topics

El mecanismo de publicación/suscripción de Gazebo (bus gz-transport, distinto del de ROS 2).

```
gz topic -l
# listar todos los topics
gz topic -i -t <topic>
# información de un topic (tipo de mensaje, publicadores...)
gz topic -e -t /world/<mundo>/stats -n 1
# leer 1 mensaje y salir (en vez de escuchar en bucle)
gz topic -e -t <topic>
# escuchar en bucle (Ctrl+C para salir)
```

## gz model — información de modelos en la escena

Consulta modelos ya cargados en una simulación en marcha.

```
gz model --list
# listar los modelos de la escena
gz model -m <modelo> --pose
# pose (posición/orientación) del modelo
gz model -m <modelo> -l
# links del modelo
```

## gz fuel — gestionar recursos de Gazebo Fuel

Mundos y modelos alojados en el repositorio público de Gazebo Fuel ([app.gazebosim.org/fuel](http://app.gazebosim.org/fuel)).

```
gz fuel list -o <owner> --limit 5
# listar modelos de un propietario
gz fuel download -u <url>
# descargar un modelo/mundo al disco local
```

## gz sdf — validar y depurar archivos SDF

Comprueba un mundo o modelo sin necesidad de arrancar Gazebo.

```
gz sdf --check mundo.sdf
# validar la sintaxis ("Valid." si es correcto)
gz sdf --print mundo.sdf
# imprimir el SDF ya resuelto (útil para depurar <include>)
```

### Trampa real: campos "ausentes" en gz topic -e

El formato de texto de protobuf que imprime `gz topic -e` omite cualquier campo que valga su valor por defecto (`false` o `0`) —no imprime `paused: false` ni `real_time_factor: 0`. Que un campo no aparezca significa "vale su valor por defecto", nunca "desconocido".

Post completo con un ejemplo funcional (nodo ROS 2 que automatiza arrancar / pausar / reanudar / cambiar velocidad / consultar el mundo) en **Robótica Fácil con ROS 2** ([robotica-facil-con-ros2.es](http://robotica-facil-con-ros2.es)).