

# **Despliegue de Simulaciones Robóticas en Docker: Brazo Robótico y Pista de Carreras**

Yossed Riaño  
Jeferson Hernández  
Miguel Montaña

Octubre 2025

# Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Objetivos</b>	<b>3</b>
<b>3. Descarga e instalación de Docker</b>	<b>3</b>
3.1. Windows . . . . .	3
3.2. Linux . . . . .	3
<b>4. Desarrollo de los proyectos</b>	<b>3</b>
4.1. Pista de Carreras (Tkinter) . . . . .	3
4.2. Brazo Robótico (PyBullet) . . . . .	3
<b>5. Dockerización</b>	<b>4</b>
5.1. Pista de Carreras . . . . .	4
5.2. Brazo Robótico . . . . .	4
<b>6. Resultados</b>	<b>4</b>
<b>7. Evidencias</b>	<b>4</b>
<b>8. Conclusiones</b>	<b>5</b>
<b>9. Referencias</b>	<b>5</b>

# 1. Introducción

El presente documento describe el proceso de creación, configuración y despliegue de dos proyectos de simulación desarrollados en Python, utilizando las bibliotecas Tkinter y PyBullet, ambos ejecutados dentro de contenedores Docker.

## 2. Objetivos

- Desarrollar una pista de carreras sencilla utilizando Tkinter.
- Implementar un brazo robótico con PyBullet.
- Dockerizar ambos proyectos para garantizar su portabilidad y ejecución en cualquier entorno.

## 3. Descarga e instalación de Docker

### 3.1. Windows

1. Descargar e instalar Docker Desktop desde: <https://www.docker.com/products/docker-desktop/>
2. Descargar e instalar VcXsrv desde: <https://sourceforge.net/projects/vcxsrv/>
3. Al iniciar VcXsrv, marcar la opción *Disable access control*.

### 3.2. Linux

1. Instalar Docker usando el gestor de paquetes correspondiente.
2. Permitir acceso al servidor X11 ejecutando el comando:

```
xhost +local:docker
```

## 4. Desarrollo de los proyectos

### 4.1. Pista de Carreras (Tkinter)

Aplicación desarrollada en Python utilizando la librería Tkinter. Representa una pista en forma de pentágono por la cual se desplaza un carro de manera automática.

### 4.2. Brazo Robótico (PyBullet)

Simulación 3D de un brazo robótico KUKA IIWA controlado mediante movimientos sinusoidales de sus articulaciones principales.

## 5. Dockerización

### 5.1. Pista de Carreras

```
docker build -t pista-pentagono .  
docker run -it --rm -e DISPLAY=host.docker.internal:0.0 pista-pentagono
```

### 5.2. Brazo Robótico

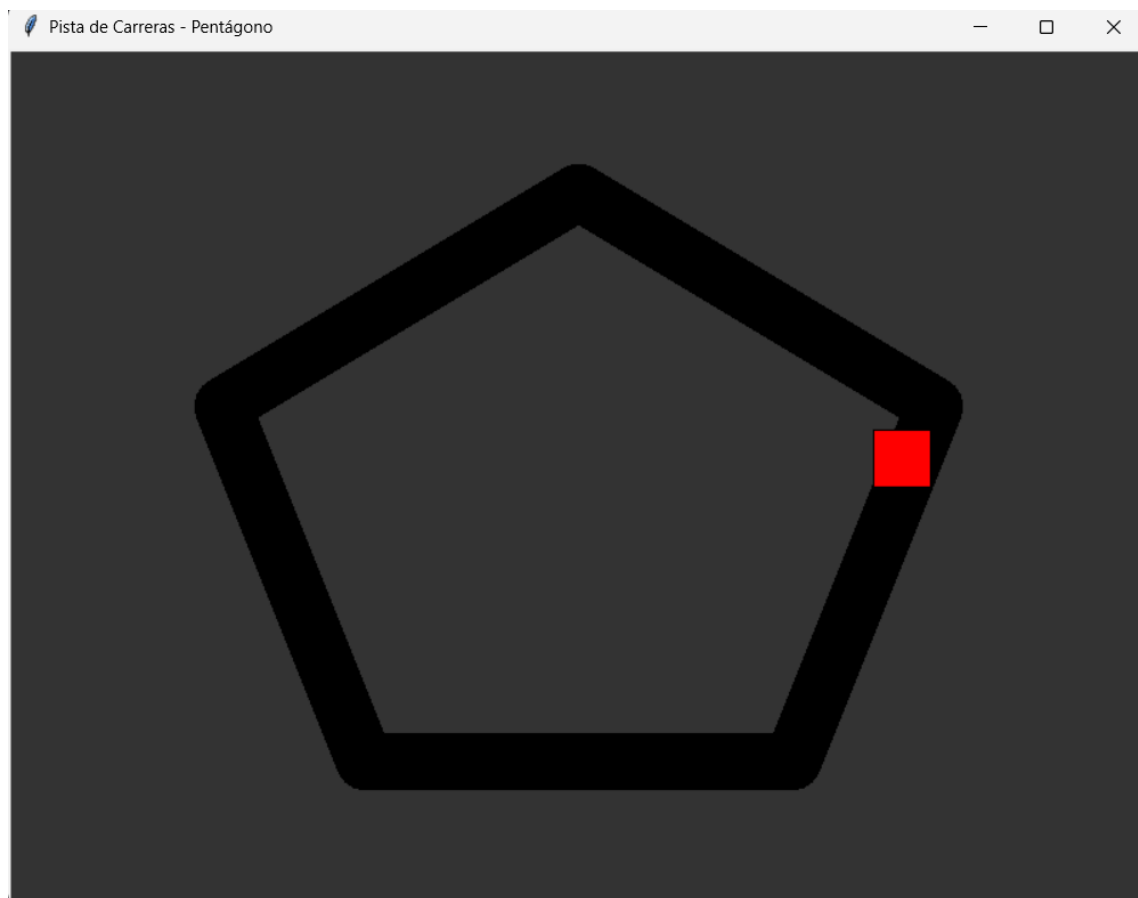
```
docker build -t brazo-pybullet -f Dockerfile.brazo .  
docker run -it --rm -e DISPLAY=host.docker.internal:0.0 brazo-pybullet
```

## 6. Resultados

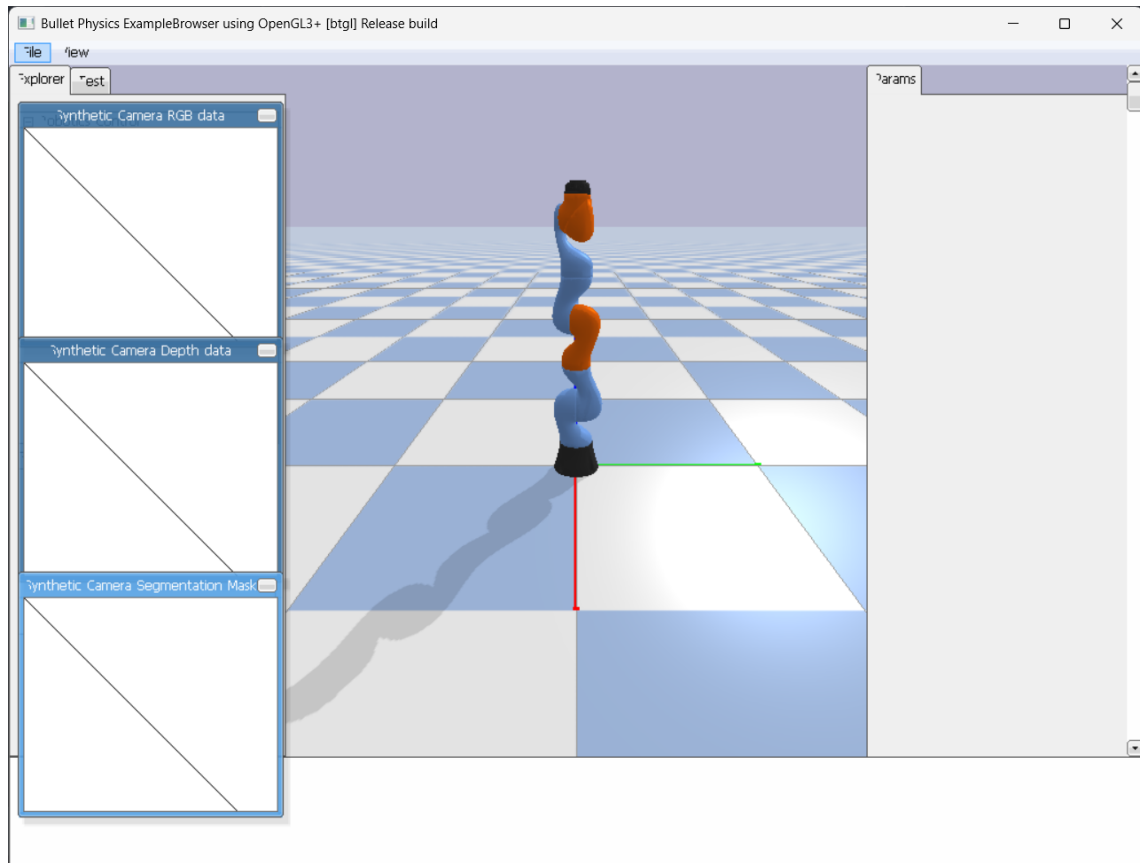
- Pista de carreras visualizada en una ventana Tkinter.
- Brazo robótico mostrado en entorno 3D PyBullet.

## 7. Evidencias

- Captura de ejecución de la pista en Docker.



- Captura del brazo robótico en PyBullet.



## 8. Conclusiones

El uso de Docker permite ejecutar proyectos gráficos de simulación sin conflictos de entorno. Tkinter proporciona una base sencilla para interfaces 2D, mientras que PyBullet ofrece un entorno físico realista en 3D. Ambas herramientas, combinadas con Docker, facilitan la portabilidad y reproducibilidad del desarrollo.

## 9. Referencias

- <https://pybullet.org/wordpress/>
- <https://docs.python.org/3/library/tkinter.html>
- <https://docs.docker.com/>