

Entrenamiento en entorno virtual para la teleoperación segura de Brazos Robóticos

Realidad Virtual - Facultad de Ingenieria UNCuyo

Juan Ignacio Quiroga

Francisco Castel

Octubre 2025

Contents

1	Resumen	2
2	Motivación y justificación	2
3	Objetivos	2
3.1	Objetivo general	2
3.2	Objetivos específicos	2
4	Alcance y exclusiones	3
5	Requisitos	3
5.1	Funcionales	3
5.2	No funcionales	4
6	Estado del arte (breve)	4
7	Metodología y tecnologías	4
7.1	Motor gráfico y XR	4
7.2	Arquitectura	4
7.3	Hardware de interfaz: Joystick RPi Pico	5
7.4	Simulación del robot	6
7.5	Teleoperación y seguridad	7
8	Plan de evaluación	7
8.1	Métricas	7
8.2	Diseño experimental	7
9	Entregables	7
10	Riesgos y mitigaciones	7
11	Consideraciones éticas y de seguridad	8
12	Plan de bibliografía	8

1 Resumen

Este anteproyecto propone el desarrollo de un **entorno virtual (VR)** para **entrenar a operarios** en el uso de un **brazo robótico** y la posterior **teleoperación** con un joystick diseñado ad hoc sobre *Raspberry Pi Pico*. El sistema combina simulación en *Unity*, medición de desempeño y un pipeline que facilite la transferencia desde el entrenamiento en VR al control del robot real.

2 Motivación y justificación

La teleoperación es pertinente cuando existen *riesgos ambientales*, se requiere *precisión manual*, hay *alta variabilidad* en la tarea o se busca *transferir expertise* humana. La VR reduce costos y riesgos de entrenamiento, habilita *iteración rápida* y define métricas objetivas (tiempo, trayectorias, colisiones, suavidad del movimiento).

3 Objetivos

3.1 Objetivo general

Desarrollar un entorno VR y un esquema de control que permitan entrenar y luego teleoperar un brazo robótico mediante un joystick propio, priorizando seguridad, usabilidad y desempeño.

3.2 Objetivos específicos

1. Implementar un **simulador** de brazo robótico en Unity con interacción XR.
2. Diseñar el **mapeo de controles** del joystick (RPi Pico) a los grados de libertad del robot.
3. Instrumentar **métricas de entrenamiento**: tiempo, errores, colisiones, suavidad/jerk.
4. Diseñar y validar **modos de teleoperación**: velocidad limitada, asistencia, seguimiento de path.

5. Incorporar **mecanismos de seguridad**: e-stop, zonas prohibidas, límites de torque/velocidad.
6. Realizar una **evaluación** con usuarios: usabilidad (SUS), aprendizaje y transferencia a robot real/simulador de mayor fidelidad.

4 Alcance y exclusiones

Alcance

- Entrenamiento VR de tareas representativas (p.ej., pick&place, posicionamiento preciso).
- Teleoperación con joystick RPi Pico (prototipo funcional).
- Integración de comunicación (serial/UDP/TCP) y capa de seguridad básica.

Fuera de alcance (por ahora)

- Retroalimentación háptica avanzada.
- *Motion planning* autónomo o visión por computadora compleja.
- Certificación normativa industrial.

5 Requisitos

5.1 Funcionales

- Control en tiempo real del robot (simulado) desde el joystick.
- Escenarios VR con objetivos, *checkpoints* y feedback.
- Registro de métricas y generación de reportes.
- Modo teleoperado con límites de seguridad (e-stop, zonas).

5.2 No funcionales

- **Latencia** end-to-end $< 60\text{--}100\text{ ms}$ (objetivo inicial).
- **Usabilidad**: $\text{SUS} \geq 70$.
- **Compatibilidad**: Windows y *headset* VR objetivo.
- **Mantenibilidad**: arquitectura modular y documentación.

6 Estado del arte (breve)

Se revisarán enfoques de simulación de robots en motores 3D, bibliotecas XR, estándares de seguridad en teleoperación y soluciones de *joysticks* personalizados.

7 Metodología y tecnologías

7.1 Motor gráfico y XR

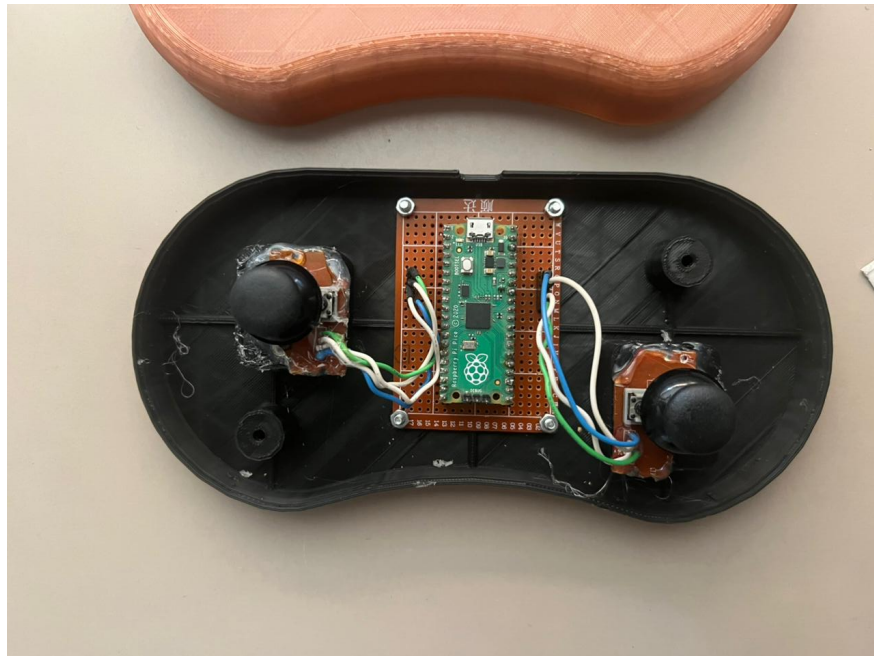
Unity (render pipeline apropiado) y sistema de *input*.

7.2 Arquitectura

- **Capa de Dispositivos**: joystick RPi Pico \rightarrow PC (USB/serial/UDP).
- **Capa de Control**: mapeo joystick \rightarrow comandos (velocidad/posición).
- **Capa de Simulación**: cinemática/dinámica y colisiones.
- **Capa de VR/UX**: escenas, HUD, feedback, tutoriales.
- **Teleoperación**: canal de comunicación + *safety layer*.

7.3 Hardware de interfaz: Joystick RPi Pico

- **Estado actual:** primer prototipo impreso 3D y firmware básico (PWM/ADC/USB HID o serial).
- **Plan:** iteraciones de ergonomía, *deadband*, *debounce*, vibración opcional.



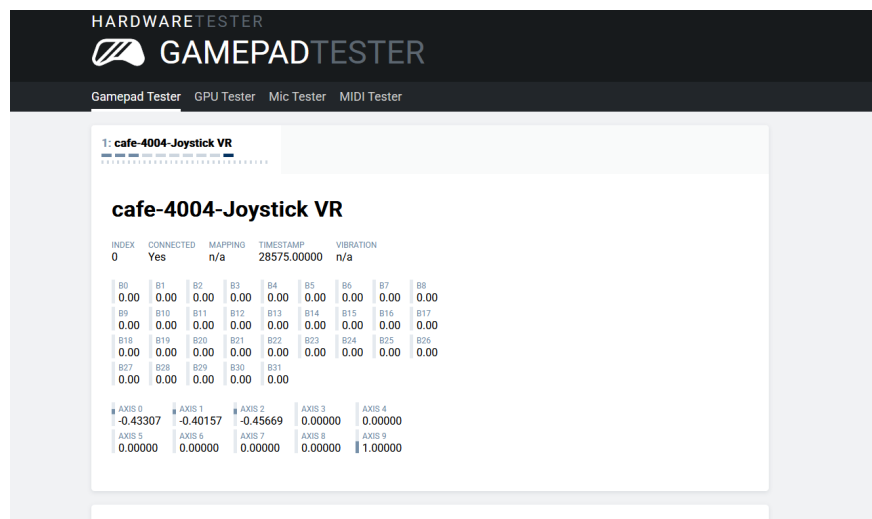


Figure 1: Se utilizó la librería TinyUSB para implementar el protocolo HID

7.4 Simulación del robot

Cinemática directa/inversa, límites articulares, velocidad y *soft stops*. Opcional: exportar URDF y/o puente ROS2 para escalabilidad futura.

7.5 Teleoperación y seguridad

Modos de control (velocity/position), *rate limiting*, filtros, geofencing, e-stop físico/virtual, *watchdog* de conexión.

8 Plan de evaluación

8.1 Métricas

Tiempo por tarea, tasa de errores/colisiones, suavidad (jerk promedio), # de reintentos, aprendizaje (*pre/post*), SUS.

8.2 Diseño experimental

$N \approx 8-12$ participantes (piloto), ensayo de 2-3 tareas con y sin asistencia, análisis de normalidad y comparaciones (p.ej., Wilcoxon/ttest).

9 Entregables

- Demo VR funcional (entrenamiento).
- Prototipo de joystick RPi Pico (STL/FW).
- Módulo de teleoperación con *safety layer*.
- Informe con resultados y repositorio con código/documentación.

10 Riesgos y mitigaciones

- Latencia alta \rightarrow *rate limiting*, compresión, simplificación de escena.
- Inestabilidad de control \rightarrow saturaciones, *low-pass*, límites articulares.
- Fatiga del usuario \rightarrow sesiones cortas, ergonomía del joystick, tutoriales progresivos.

11 Consideraciones éticas y de seguridad

Privacidad de datos de uso, seguridad funcional (e-stop), ergonomía y prevención de mareo en VR (pausas, locomoción adecuada).

12 Plan de bibliografía

Se incluirán trabajos de teleoperación, entrenamiento en VR, cinemática de robots y guías de HMI/seguridad.