



UNIVERSIDAD DEL AZUAY
FACULTAD DE CIENCIA Y TECNOLOGÍA
ESCUELA DE INGENIERÍA ELECTRÓNICA

**“Desarrollo de una Interfaz de Realidad Aumentada para el
Control Telemático a Tiempo Real de un Robot
Manipulador.”**

Trabajo de graduación previo a la obtención del título de:

INGENIERO ELECTRÓNICO

Autores:

ANGÉLICA MARÍA ABAD FERNÁNDEZ DE CÓRDOVA

MARCO ANTONIO CHICA SALAZAR

Director:

ING. ANDRÉS PATRICIO CABRERA FLOR

CUENCA, ECUADOR

2023

DESARROLLO DE UNA INTERFAZ DE REALIDAD AUMENTADA PARA EL CONTROL TELEMÁTICO A TIEMPO REAL DE UN ROBOT MANIPULADOR

En la presente investigación se realizó el desarrollo e implementación de una aplicación de realidad aumentada (AR) que sirva para realizar el control telemático y a tiempo real de un robot manipulador. El sistema propuesto consiste en generar una versión virtual interactiva de un robot manipulador e incluirla dentro de un entorno AR, de modo que el robot físico sea capaz de replicar los movimientos realizados en el modelo virtual. Se utilizaron: un dispositivo móvil que contiene la aplicación AR diseñada en la plataforma de videojuegos Unity 3D, una Raspberry Pi 4 con ROS (Robot Operating System) y un robot manipulador. Además, para establecer la comunicación entre ROS y Unity se empleó una comunicación TCP-IP sobre Wi-Fi. Como resultado, la aplicación fue capaz de posicionar cada articulación de un robot manipulador tras mover un modelo virtual del mismo, obteniendo así un control a tiempo real eficiente, preciso y amigable con el usuario.

Palabras clave: Realidad Aumentada, AR, ROS, robótica, robot manipulador, Unity 3D, interacción humano-robot.



Firmado electrónicamente por:
ANDRÉS PATRICIO
CABRERA FLOR

Andrés Patricio Cabrera Flor

Director de Tesis



Firmado electrónicamente por:
DANIEL ESTEBAN
ITURRALDE PIEDRA

Daniel Esteban Iturralde Piedra

Director de Escuela

Angélica María Abad Fernández de Córdova

Autor

Marco Antonio Chica Salazar

Autor

DEVELOPMENT OF AN AUGMENTED REALITY INTERFACE TO ACHIEVE REAL-TIME TELEMATIC CONTROL OF A ROBOTIC MANIPULATOR

This research describes the development and implementation of an augmented reality (AR) mobile application, which performs the teleoperation and real-time control of a robotic manipulator. The proposed system is based on generating an interactive virtual model of a robot arm and place it on an AR environment, so that the real robotic arm can replicate all movements made by its virtualization. A mobile device containing the AR app developed with Unity 3D, a Raspberry Pi 4 with ROS (Robot Operating System) and a robotic arm manipulator were needed to complete this project. In addition, TCP- IP communication over Wi-Fi was implemented to establish the connection between ROS and Unity. As a result, users were able to place each joint of a robotic manipulator on the desired position by only moving the virtual model of the robot through the developed AR app. The AR interface achieved an efficient and precise real-time control and was highly user-friendly.

Palabras clave: Augmented Reality, AR, ROS, robotics, robotic arm manipulator, Unity 3D, human-robot interaction.



Firmado electrónicamente por:
ANDRÉS PATRICIO
CABRERA FLOR

Andrés Patricio Cabrera Flor

Thesis Director



Firmado electrónicamente por:
DANIEL ESTEBAN
ITURRALDE PIEDRA

Daniel Esteban Iturralde Piedra

Faculty School Coordinator

Angélica María Abad Fernández de Córdova

Author

Marco Antonio Chica Salazar

Author



Translated by

Angélica María Abad

Desarrollo de una Interfaz de Realidad Aumentada para el control Telemático a Tiempo Real de un Robot Manipulador

Angélica María Abad Fernández de Córdova
Escuela de Ingeniería Electrónica
Universidad del Azuay
Cuenca, Ecuador
angelicamabad@es.uazuay.edu.ec

Marco Antonio Chica Salazar
Escuela de Ingeniería Electrónica
Universidad del Azuay
Cuenca, Ecuador
mchicas@es.uazuay.edu.ec

Resumen—En la presente investigación se realizó el desarrollo e implementación de una aplicación de realidad aumentada (AR) que sirva para realizar el control telemático y a tiempo real de un robot manipulador. El sistema propuesto consiste en generar una versión virtual interactiva de un robot manipulador e incluirla dentro de un entorno AR, de modo que el robot físico sea capaz de replicar los movimientos realizados en el modelo virtual. Se utilizaron: un dispositivo móvil que contiene la aplicación AR diseñada en la plataforma de videojuegos Unity 3D, una Raspberry Pi 4 con ROS (Robot Operating System) y un robot manipulador. Además, para establecer la comunicación entre ROS y Unity se empleó una comunicación TCP-IP sobre Wi-Fi. Como resultado, la aplicación fue capaz de posicionar cada articulación de un robot manipulador tras mover un modelo virtual del mismo, obteniendo así un control a tiempo real eficiente, preciso y amigable con el usuario.

Index Terms—Realidad Aumentada, AR, ROS, robótica, robot manipulador, Unity 3D, interacción humano-robot.

I. INTRODUCCIÓN

Incursionar en nuevas tecnologías es una tarea fundamental para el desarrollo de la sociedad y de la ciencia. La evolución de los sistemas de control a través de los años ha permitido que la interacción entre las máquinas y las personas sea cada vez más amigable, de forma que en la actualidad se han vuelto indispensables en muchas actividades de la vida cotidiana. Esto ha permitido el desarrollo de campos como la robótica, la cual ha jugado un papel muy importante en el fortalecimiento de la relación humano-robot, mediante la agilización de tareas repetitivas dentro de la industria, el manejo de sustancias corrosivas o tóxicas para el ser humano, el mantenimiento de maquinaria en lugares de difícil acceso, la investigación aeroespacial, submarina o subterránea, entre otros.

La robótica es una rama de la ingeniería con un vasto campo de aplicaciones, de modo que se ha ido adaptando según las necesidades de cada área en específico. Por ello, es posible encontrar robots de diferente naturaleza, ya sean móviles, manipuladores, humanoides, entre otros, de acuerdo a la actividad que esté destinado a realizar [1]. Los robots manipuladores son unos de los más empleados en los diferentes campos de la industria e investigación, debido a su capacidad de

movimiento y alta funcionalidad para efectuar diversas tareas: consisten en brazos robóticos articulados capaces de manejar objetos encontrados en su entorno; son de carácter sedentario y su elevado número de grados de libertad (capacidad de movimiento según sus articulaciones) le brindan al sistema mayor movilidad, de acuerdo a una trayectoria establecida.

Las interfaces de control tradicionales desarrolladas para robots manipuladores consisten en elementos físicos como joysticks, palancas y pulsantes que accionan las diferentes funciones del robot que, a su vez, se visualizan a través de una pantalla. Sin embargo, a pesar de ser operacionalmente eficientes, estos sistemas no permiten al operador interactuar y percibir los movimientos del robot adecuadamente: la única relación que se desarrolla entre el operador y el robot es mediante accionadores mecánicos y una pantalla. Esto implica que un operador debe tener experiencia para manejar la interfaz de control, de manera que se familiarice con factores como sensibilidad, precisión y respuesta del sistema ante el accionamiento de los controladores.

Adicionalmente, este problema se acentúa aún más cuando se habla de aplicaciones que involucran teleoperación. La teleoperación, definida como el control de un dispositivo de forma remota [2] de modo que el operador y el elemento a controlar se encuentran en dos locaciones distintas, es uno de los desafíos más grandes dentro de la robótica. Para lograr un control telemático eficiente de un robot, se requiere que el usuario tenga un nivel alto de conocimiento tanto del comportamiento del dispositivo así como de su sistema de control. A pesar de ello, muchas veces el operador no logra obtener la retroalimentación necesaria de parámetros como precisión y posicionamiento del robot, para conseguir un mejor dominio sobre el dispositivo. Por esta razón, es necesario aprovechar las nuevas tecnologías para mejorar la interacción humano-robot en el desarrollo de interfaces de control dedicadas a la teleoperación de robots manipuladores, siendo la realidad aumentada, o también conocida como AR por sus siglas en inglés, una de las más llamativas.

En los últimos años, la realidad virtual y sus contra partes (AR y realidad mixta) han captado la atención de muchos

investigadores y desarrolladores de distintas áreas, debido a su gran potencial de brindar al ser humano una nueva experiencia con el mundo digital: como su nombre lo indica, la realidad virtual es la creación de entornos 3D completamente virtuales en los que un usuario puede interactuar como en el mundo real. Esta, a pesar de desarrollarse inicialmente para videojuegos, ha servido mucho en distintos campos de la investigación y la industria como la medicina, la educación, la arquitectura entre otros [2]. Sin embargo, dada la necesidad de integrar objetos virtuales en escenarios reales, surgió la tecnología AR, la cual permite a una persona interactuar con objetos generados por computadora (objetos virtuales) en el mundo real, de modo que el usuario tiene una mayor noción de lo que sucede a su alrededor. Esta incorporación de elementos reales y virtuales han llamado la atención de investigadores de diversos sectores, como el turismo, la educación e incluso la robótica.

Existen diversas formas de implementar ambientes de AR, desde plataformas básicas como lo son Zappar, Apy Pie, Hp Reveal, hasta herramientas más complejas en donde se necesita más conocimiento sobre programación como lo son Vuforia, ARtoolkit o Unity. Las plataformas básicas se caracterizan por ser de carácter comercial, por lo que no ofrecen muchos servicios para desarrollo y experimentación [3]. Sin embargo, entornos como Unity, Vuforia o ARtoolkit son herramientas más versátiles y de software libre que permiten la investigación y el uso de paquetes integrados, con lo cual es posible crear texturas y modelos de figuras tridimensionales de todo tipo. Además, la ventaja de integrar la programación en el diseño de estos ambientes virtuales es que otorga la libertad de editar hasta el más mínimo detalle de los mismos, por lo que las capacidades de aplicación de la tecnología AR aumentan.

La plataforma Unity, a pesar de ser dedicada al desarrollo de videojuegos, posee funciones que en la actualidad son muy usadas para el diseño de sistemas de carácter científico e investigativo. Como se mencionó anteriormente, Unity es uno de los entornos de desarrollo de videojuegos más empleados en la actualidad, siendo juegos como Pokémon Go, Heartstone o Mobius Final Fantasy algunos de los productos más populares creados a través de esta plataforma. Como se observa, Unity ofrece una vasta cantidad de elementos y herramientas de desarrollo, lo que demuestra su gran capacidad de creación de entornos virtuales con alto nivel de detalle. Además, recientemente Unity se ha integrado más en el diseño en AR, permitiendo a desarrolladores crear aplicaciones y entornos AR para sus videojuegos y otros propósitos, a través del paquete «AR Foundation».

El sistema operativo robótico, o mejor conocido como ROS por sus siglas en inglés, es un sistema de acceso abierto, diseñado para el control y desarrollo de aplicaciones relacionadas con robótica. Su funcionamiento se basa en una variedad de herramientas de software que permiten construir sistemas robóticos complejos, mediante la codificación, envío y recepción de mensajes entre «nodos», o puntos del sistema, a través de «tópicos». Un nodo ROS puede funcionar como publicador («publisher» o emisor), suscriptor («subscriber» o receptor) o jugar ambos papeles, facilitando así la comunicación entre

partes distintas de un sistema robótico. Por otro lado, un tópico ROS sirve como medio de transporte de datos para que los mensajes se transmitan entre dos o más nodos. Para que ROS opere adecuadamente, es necesario la instalación de una distribución compatible con el robot a utilizar en un computador a base Linux [4].

Debido a la problemática que actualmente se experimenta en el campo de la robótica con respecto a las interfaces de control de robots manipuladores y por las ventajas que la tecnología AR brinda para esta clase de aplicaciones, la presente investigación expone el diseño e implementación de una interfaz basada en AR para realizar la teleoperación de robots manipuladores. Para ello, se utilizó la plataforma Unity 3D y el sistema operativo dedicado a robótica ROS, lo cual facilitó la puesta en marcha del sistema.

II. ESTADO DEL ARTE

Dentro de la robótica, la AR promete innovar los sistemas de control tradicionales para brindar al operador una mejor interacción con el dispositivo robótico, mejorando así la respuesta del sistema a las instrucciones enviadas por el usuario [5]. De esta forma, funciones como la optimización de trayectorias, la teleoperación o el desarrollo de interfaces de control más eficientes han sido producto de la inmersión de AR en la robótica.

Como se evidencia en la investigación de Rodríguez [6], una de las ventajas en comparación a la AR con respecto a la realidad virtual es el grado de inmersión del usuario en la misma; sin embargo, esto podría llevar a un problema paradójico en el cual la interfaz sea sobrecogedora para el usuario, produciendo confusión al momento de operar el sistema. Un ejemplo de ello se refleja en el sistema de realidad virtual de [3], en el cual se diseñaba un entorno como el de un GUNDAM para operar un robot manipulador tipo pinza; por ende, depende mucho del diseño final del entorno AR para que este tipo de aplicaciones sean efectivas o no.

Cuando se habla de teleoperación, principalmente se hace referencia a dos clases en particular, la teleoperación directa y la teleoperación asistida [7]. En el método asistido, el sistema utiliza herramientas como inteligencia artificial, sensores y redondeos para corregir errores que podrían producirse durante el movimiento del robot. Sin embargo, si no se realiza un buen acondicionamiento de señales, el robot puede efectuar movimientos totalmente distintos a los esperados, aumentando el error del sistema. Por otro lado, en la teleoperación directa, el robot efectúa exactamente los mismos movimientos que aquellos que se producen en el sistema de control a distancia, por lo cual si existe un error en el comando enviado al robot, éste se verá reflejado de forma inmediata en el dispositivo. Este método de teleoperación no contiene ningún mecanismo que contrarreste o corrija la falla por completo, por lo cual este sistema exige una mayor exactitud al generar la dinámica y la cinemática del robot a controlar para evitar errores de control.

La teleoperación presenta grandes desafíos cuando se trata de robótica, en especial en sistemas a tiempo real. Según

Walker [8], existen tres problemas principales con respecto a los paradigmas de la teleoperación que representan un alto riesgo para el usuario y para el sistema: (1) que el operador se familiarice con los controles del sistema; (2) identificar cómo responde el robot a cada comando dado por el operador, (3) que el operador deba manipular el sistema directamente para lograr su objetivo. Estos problemas se dan, en parte, debido a que las interfaces comunes de control no permiten al usuario tener una completa retroalimentación del sistema, lo cual limita el dominio del operador sobre el robot. Por ende, debido a la distancia entre el robot y el usuario, es difícil para el operador identificar los movimientos exactos del sistema a tiempo real, razón por la cual los controladores actuales no son tan eficientes en estas situaciones. Gracias a la AR es posible tener un monitoreo más versátil y a tiempo real sobre la actividad del robot, sin necesidad de que el operador se encuentre en la misma habitación, como se demuestra en el software desarrollado en [9].

Adicionalmente, varias investigaciones acerca del control mediante gestos se han realizado, las cuales se relacionan con el reconocimiento de expresiones faciales y el movimiento de las manos permiten al robot realizar ciertas actividades, de acuerdo al comando dado por el cuerpo. En [10] se presenta un sistema de tele-control mediante AR, excepto que en este caso se realiza el control del robot a través de los gestos de las manos, utilizando un sensor Leap Motion. Además, si esta tecnología acopla inteligencia artificial con autonomía de aprendizaje, tomaría una gran fuerza en la industria, ya sea a baja o gran escala [11] [12].

Uno de los objetivos del uso de AR en robótica es mejorar la interacción humano-máquina a través del desarrollo de interfaces que proponen una comunicación versátil y variada. Esto permite que exista una mayor libertad de control y un mejor flujo de información, ya sea de humano a máquina o viceversa. A diferencia de los controladores tradicionales, la tecnología AR permite que los comandos e instrucciones enviados al robot actúen con mayor rapidez que en métodos tradicionales. Esto permite introducir al usuario en un enfoque más interactivo e intuitivo mediante visores, sets AR y el desarrollo de aplicaciones AR en plataformas como Unity.

Las interfaces AR de control más comunes utilizadas en investigación son aquellas en las que el robot imita los movimientos de un robot virtual similar, como en [13] o [14]. Sin embargo, existen interfaces que se basan solo en generar controladores virtuales sin necesidad de tener un enfoque virtual del robot [15]. En el caso de la presente investigación, la concentración se colocará en el desarrollo de una interfaz AR dedicada a la teleoperación a tiempo real de un robot manipulador con el objetivo de obtener un mejor seguimiento y dominio sobre un sistema robótico.

III. METODOLOGÍA

Como se mencionó anteriormente, el objetivo principal del presente proyecto es desarrollar una interfaz AR capaz de controlar a distancia un robot manipulador; la interfaz propuesta consiste en una aplicación móvil basada en AR

que genere un modelo 3D interactivo del robot manipulador que se pretende controlar, de manera que el usuario pueda seleccionar cada pieza individual del robot y moverla a la posición deseada. Una vez que se efectúe este movimiento, el robot físico procederá a replicar en tiempo real toda acción que realice el modelo virtual. Para ello, el proyecto se estructuró de forma que la comunicación entre la interfaz AR y el robot manipulador se realice como se observa en el diagrama de la Figura 1.

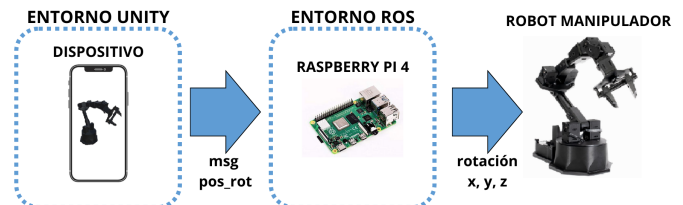


Figura 1: Diagrama de bloques del sistema.

Como parte de los equipos de hardware y software utilizados están:

- **Robot manipulador Interbotix PincherX 150**

Es un brazo robótico dedicado para desarrollo e investigación. Posee 5 grados de libertad y capacidad máxima de carga de 50 gramos. Se puede programar a través del framework ROS, en el cual se incluyen paquetes, librerías, APIs (conexión Python-Ros y Matlab-ROS), etc., lo cual brinda mayor facilidad en el control y manejo del robot [16].

- **Celular compatible con AR**

Teléfono celular capaz de soportar aplicaciones AR. En [17] se encuentra una lista de todos los dispositivos compatibles con esta tecnología; en este caso, se usó un Samsung Galaxy Note 8, basado en el sistema operativo Android 9.0.

- **Raspberry Pi 4**

La tarjeta de desarrollo Raspberry Pi 4, con 4 GB de RAM, procesador quadcore ARM v8 de 64GB y conectividad Bluetooth y Wi-Fi [18], es el equipo sobre el cual se ejecutó ROS, a través de la distribución Ubuntu Mate basada en Linux.

- **Unity 3D Engine**

Unity es un software de desarrollo de videojuegos cuya funcionalidad abarca el modelado 3D, animación 2D y 3D, creación de aplicaciones compatibles con diferentes sistemas operativos (Android, iOS, Windows, etc.), simulación 3D, y hasta desarrollo de entornos de realidad virtual, aumentada y mixta (combinación de las dos anteriores) [19]. La versión de Unity 3D empleada en el proyecto es la 2020.3.25f (LTS), además de los paquetes que permiten la integración AR en el entorno 3D.

- **Robotic Operating System (ROS)**

ROS es un sistema de acceso abierto que incluye una colección de librerías, paquetes y otras herramientas de software que facilitan la creación y manejo de aplica-

ciones robóticas. Tiene todas las funcionalidades de un sistema operativo, pero dedicado al desarrollo robótico, es decir contiene drivers, capacidad de manejo de paquetes, transmisión de mensajes a través de procesos, herramientas para desarrollo de código, etc. ROS no se puede ejecutar por sí solo, por lo que sus distribuciones necesitan ser instaladas sobre otro sistema operativo base como Linux, o en el caso del presente proyecto, una distribución de Linux, como lo es Ubuntu Mate. La versión compatible con el robot manipulador usado PincherX 150 es ROS Noetic 20.4 [20].

Por otro lado, la implementación del proyecto fue dividida en tres etapas fundamentales:

- A. Desarrollo del entorno AR en Unity 3D
- B. Preparación del entorno ROS
- C. Comunicación entre ROS y Unity
- D. Implementación y funcionamiento de la interfaz AR

A. Desarrollo del entorno AR en Unity 3D

Para diseñar la aplicación AR propuesta, primero fue necesario familiarizarse con las funciones que Unity 3D ofrece; al ser un entorno dedicado al desarrollo de juegos de video, Unity 3D brinda la posibilidad de simular la interacción de objetos 3D entre sí, siendo estos objetos diseñados por el propio desarrollador o importando modelos ya creados (assets) al entorno. De esta forma, se importaron a Unity los modelos 3D de cada articulación del robot PincherX 150 disponibles en el repositorio de Github Interbotix, como se muestra en la Figura 2; a este modelo se lo conoce como prefabricado. Para el diseño del prefabricado se tomó en consideración el movimiento jerárquico de las articulaciones móviles del robot, de modo que si la pieza con mayor jerarquía se mueve, todos los elementos debajo de ella también lo harán en la misma proporción que la articulación padre.



Figura 2: Diseño de prefabricado de robot manipulador en Unity.

Con el modelo 3D generado, se procedió a animar cada articulación con el teclado del computador, según los movimientos reales del brazo robótico. Se asignó una tecla por articulación de modo que exista independencia de movimiento entre cada

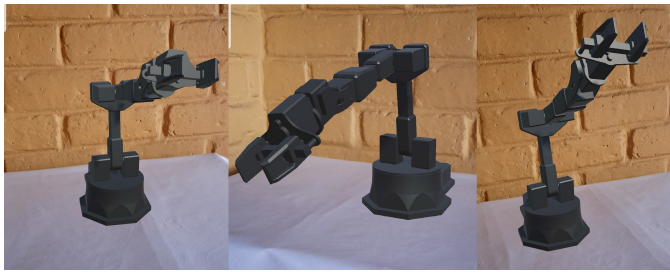
una, por ejemplo, la tecla «Q» controla únicamente el giro de la base (articulación «shoulder») del robot mientras que la tecla «S» controla la rotación del antebrazo (articulación «upperarm»). Para lograr esto, se desarrollaron scripts en C# usando los recursos que Unity ofrece en Visual Studio. Es importante recalcar que todas las articulaciones tienen como eje referencial de movimiento a su articulación «padre», de modo que facilite la obtención de coordenadas de cada pieza.

Para exportar el modelo prefabricado a la plataforma AR, se utilizó el paquete «AR Foundation», el cual permite crear aplicaciones AR a partir de un modelo o escena 3D (conjunto de objetos que interactúan en un entorno Unity 3D). Así se logró integrar el robot virtual con el mundo real, de modo que a través de la cámara de un dispositivo móvil el usuario pueda interactuar con los objetos dentro de la aplicación. Adicionalmente, para obtener una mejor inmersión del robot virtual en el mundo real, se tuvo en consideración aspectos como la textura, color, sensibilidad de movimiento, orientación, etc. De esta forma, la aplicación AR permitió seleccionar, distinguir y mover cada articulación del modelo prefabricado del robot, como se muestra en la Figura 3.



Figura 3: Interacción con el modelo en la aplicación AR.

A continuación, se procedió a desarrollar una primera versión de la aplicación. Para obtener un diseño amigable para el usuario, el equipo de investigación creyó necesario integrar elementos como un objetivo o «target» que realice la detección de planos a través de la cámara del dispositivo móvil y permita al usuario saber en dónde se colocará el modelo virtual del robot, y un controlador para abrir y cerrar el efector final del manipulador. En la Figura 4 se observan las primeras versiones de la aplicación; aquí se puede evidenciar cómo la tecnología AR provee una visualización completa del robot, pues al moverse alrededor del modelo colocado en pantalla es posible apreciarlo en 3 dimensiones, acercarse o alejarse del mismo e interactuar con el objeto como si realmente estuviera presente.



(a) Versión 1. Movimiento sin selección de articulaciones.



(b) Versión 2. Movimiento con selección de articulaciones.



(c) Versión 3. Se habilita efector final con botones virtuales.



(d) Versión 4. Definición de texturas y apariencia del robot.

Figura 4: Evolución del diseño de la aplicación AR.

Como se puede observar, la aplicación permite seleccionar cada pieza del modelo prefabricado, la cual se torna color verde una vez elegida, y permite rotar dicho elemento en la dirección en la que el robot manipulador debería moverse. Para la planificación de movimiento del efector final, fue necesario desarrollar scripts adicionales que indiquen la apertura o cierre de la pinza y asignarlos a botones virtuales (esferas de color rojo y verde) que permitan al usuario controlar el grado de apertura de la misma.

Al finalizar esta etapa, se logró diseñar y desplegar una aplicación AR que permite a un usuario interactuar con un modelo virtual de un robot manipulador de una forma sencilla e intuitiva. La aplicación funciona de modo que el usuario pueda colocar el modelo 3D del robot en cualquier superficie horizontal plana, seleccione la articulación que desea mover y

llevarla a la posición que desee, como si fuera un videojuego. Hasta este punto de la investigación, únicamente se trabajó en construir el entorno AR que servirá como interfaz para el controlador del robot.

B. Preparación del entorno ROS

Para la implementación en ROS, se utilizó un computador de placa única Raspberry Pi 4 con la distribución Ubuntu Mate para instalar y ejecutar ROS. Después de la instalación debida de todos los recursos necesarios para hacer funcionar al robot manipulador, se procedió a realizar pruebas de movimiento y retroalimentación a través de los scripts demostrativos de Python y el visualizador RViz, para evaluar la sensibilidad de movimiento del robot, su morfología y obtener más información sobre su comportamiento para realizar las respectivas mejoras a la interfaz AR. De esta manera, fue posible recopilar los datos necesarios para añadir elementos que optimicen el movimiento del robot y a su vez faciliten la interacción del usuario con la aplicación desarrollada.

C. Comunicación entre ROS y Unity

Una vez realizadas las mejoras necesarias a la aplicación AR y con el entorno ROS configurado, se procedió a interconectar las dos plataformas para transmitir los movimientos realizados desde la interfaz AR hacia el robot manipulador. En la Figura 5 se observa el sentido que deben seguir los datos para hacer que el sistema funcione correctamente.

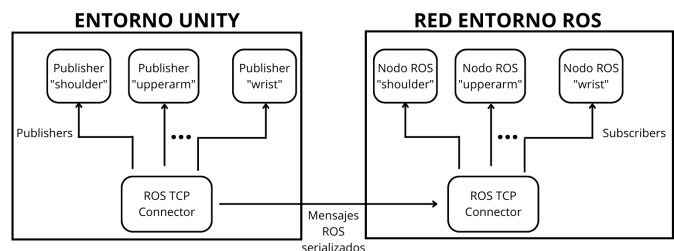


Figura 5: Envío de datos desde la intefaz AR.

Para lograr una comunicación ROS-Unity exitosa, se utilizaron las herramientas «ROS TCP Connector» en Unity, el cual permitirá a Unity enviar y recibir mensajes ROS, y «ROS TCP Endpoint» en ROS, que servirá para crear un punto final de conexión o «endpoint» para aceptar los mensajes enviados desde una escena de Unity.

Por otro lado, el diseño del sistema de nodos utilizado en el proyecto se basa en la simple comunicación uno a uno de dos nodos, un publicador y un suscriptor («publisher» y «subscriber») a través de un tópico. Este esquema funciona de modo que el punto del sistema que trabaja como publicador encuentre el tópico o medio para transmitir información hacia un suscriptor ligado al mismo tópico. De esta manera, los datos se intercambian entre dos puntos distintos de un mismo sistema: para el presente caso, la aplicación AR creada juega el papel de publicador, mientras que el entorno de ROS es el suscriptor. Es importante recalcar que tanto un publisher

como un subscriber pueden ser tratados como scripts ya sea en C# o en Python, dependiendo del entorno en el que se esté trabajando, y que cada script dentro del sistema es considerado como un nodo. En la Figura 6 se muestra el mapa detallado de nodos del sistema.

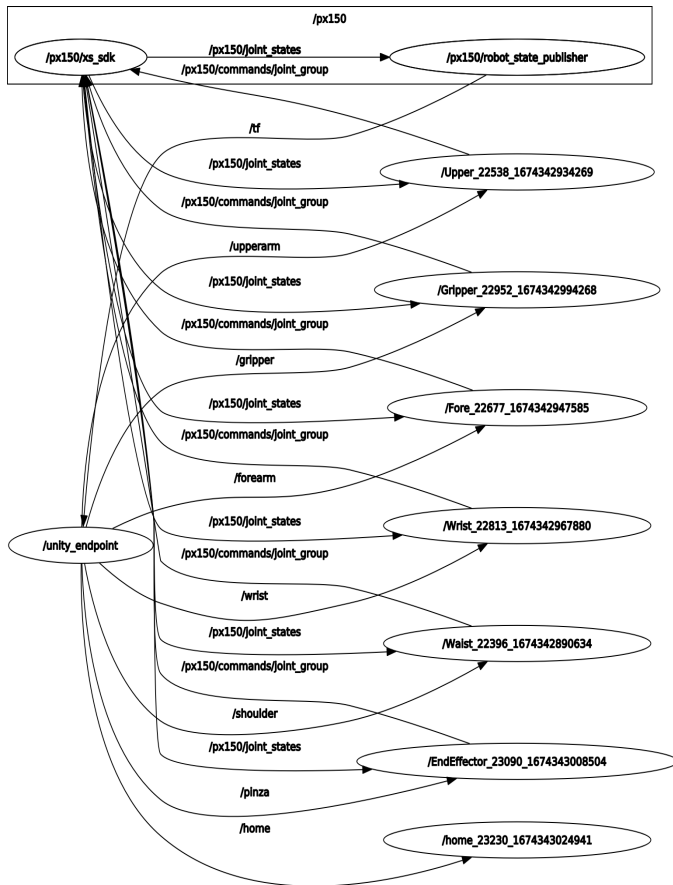


Figura 6: Sistema de nodos utilizado.

A continuación, se describe la configuración realizada en ambos entornos para lograr su interconexión:

Para la configuración del entorno en Unity, se asignó manualmente una dirección IP sobre la cual se efectuará la comunicación. Además, gracias a los paquetes instalados, se adecuaron los mensajes ROS al entorno Unity mediante la generación de scripts en C#, que traducen lo establecido en dicho mensaje [21]. Para el presente caso, el mensaje ROS utilizado contiene las coordenadas de traslación y rotación de un objeto colocado en la escena, siendo las coordenadas de rotación un cuaternión cuyas variables «x», «y», «z» y «w» serán transformadas a ángulos de Euler para facilitar la asignación de posiciones en el lado ROS.

Como cada articulación del modelo prefabricado del robot se mueve de manera individual, es posible tratar a cada pieza como un objeto cuyas coordenadas se pueden adquirir y enviar. Así, se consideró a cada articulación como un nodo que debe estar enlazado a un publisher, a un tópico y a un subscriber, cuyas identificaciones fueron asignadas según se establece en

la Figura 6. De esta forma, desde el entorno Unity, se realizó la asignación de un «publisher» y un tópico a cada articulación del modelo. Es importante recalcar que se asignó un tópico individual por articulación, debido a que la naturaleza del mensaje no permitía distinguir a qué pieza pertenecía cada coordenada si se enviaban todos los valores a través del mismo tópico.

De esta forma, cada articulación individual del robot funciona como un nodo publicador que recopila a tiempo real las coordenadas de posición y rotación y las envía a través de un tópico hacia el lado ROS. La información es obtenida gracias a los métodos de transformación de objetos en Unity y es guardada en las variables «pos_x», «pos_y», «pos_z», «rot_x», «rot_y», «rot_z» y «rot_w». Además, el script de «publisher» utilizado permite modificar la frecuencia de envío de datos desde Unity, lo cual sirve para habilitar el control a tiempo real del robot.

Por otro lado, para la configuración del entorno ROS, el paquete «ROS TCP Endpoint» permitirá aceptar los mensajes enviados desde una escena Unity, a través de una red Wi-Fi. El robot manipulador seleccionado para el proyecto tiene la facilidad de trabajar desde Python, de modo que mediante el uso de simples líneas de comando, es posible mover el robot tras ejecutar scripts desde un terminal del sistema.

Así como la aplicación AR (entorno Unity) funciona como «publisher» del sistema, el entorno ROS juega el papel de suscriptor, por lo que así como en Unity se asignaron «publishers» a cada articulación, en ROS se realizó el mismo proceso para los suscriptores, pero utilizando Python. De esta manera, se creó un suscriptor para cada articulación del robot que contenga el mensaje con todas las coordenadas individuales enviadas desde la aplicación de AR (escena de Unity). Todo «subscriber» se conecta directamente al mismo tópico configurado previamente en el publicador generado en Unity, siendo éste el canal de comunicación entre ambos nodos.

Cada suscriptor fue construido de modo que se determine la coordenada, ya sea de rotación o traslación, que permita efectuar el movimiento deseado de cada articulación. De esta manera, el suscriptor obtiene la coordenada que necesita del mensaje enviado desde Unity, lo muestra en el terminal en donde se está ejecutando el script, se suscribe al nodo de comunicación con el robot PincherX 150 y publica la coordenada a través de los métodos predeterminados de la API ROS-Python en la articulación correspondiente. Así, todo movimiento que el usuario efectúe sobre el robot virtual visualizado en la aplicación AR se replicará en el robot físico. Es importante considerar que previo a la visualización de coordenadas recibidas en el terminal del suscriptor, todas las coordenadas de rotación son previamente transformadas de cuaterniones a ángulos de Euler para facilitar la ejecución de los métodos de Python.

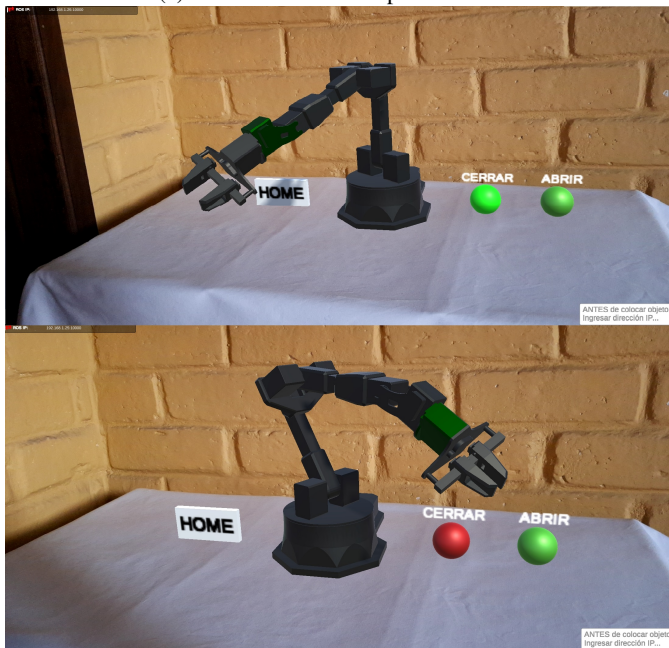
D. Implementación y funcionamiento del sistema

Una vez efectuada la conexión entre las plataformas de Unity y ROS, se procedió a implementar el sistema completo.

Es necesario considerar que desde la versión inicial de la aplicación de AR, se creyó necesario implementar nuevas mejoras para adaptar la interfaz al funcionamiento del robot. Entre estas actualizaciones estuvieron añadir un botón virtual para enviar al robot a su posición inicial, incluir etiquetas indicadoras sobre cada botón virtual dentro de la aplicación, insertar un campo en donde se ingrese la dirección IP a la que el sistema se conectará antes de colocar el robot y mejorar las texturas del modelo prefabricado del robot para una mejor visualización. Con estos ajustes adicionales, la Figura 7 muestra una serie de imágenes que indican tanto el diseño como el funcionamiento de la versión final de la aplicación AR.



(a) Diseño final de la aplicación AR.



(b) Funcionamiento de la aplicación.

Figura 7: Versión final de la interfaz AR.

De esta forma, el sistema completo funciona de la siguiente manera:

Para iniciar el sistema, antes de ejecutar la interfaz AR, es necesario asegurarse de que todos los suscriptores se encuentren activos, incluyendo los suscriptores de todas las articulaciones y botones virtuales asignados. Caso contrario,

el sistema no arrancará debidamente y la aplicación funcionará de manera independiente.

Al abrir la aplicación AR desde el celular Android en donde se haya instalado, el usuario primero deberá ingresar la dirección IP a la cual el entorno ROS se ha conectado en el recuadro inferior derecho de la pantalla, el cual ya tiene una dirección por defecto. Una vez ingresada la IP, aparecerá en pantalla un indicador («target») que realizará la detección de planos horizontales y le dirá al usuario en dónde puede colocar el modelo prefabricado del robot. Al determinar la superficie deseada, se debe hacer clic en la pantalla para colocar el robot virtual: esto interconectará a ROS y Unity e iniciará el envío de mensajes desde la aplicación AR hacia el entorno ROS. En la Figura 8, se observa la interfaz AR una vez establecida la conexión. A partir de este punto, todo movimiento que realice el usuario sobre el robot virtual será replicado por el robot físico.



Figura 8: App AR inicializada.

Para empezar a mover el robot desde la aplicación, se debe seleccionar la articulación que se desea manipular, colocar dos dedos sobre la pantalla y mantener uno de ellos en su posición inicial mientras que con el otro deslizar hacia arriba o hacia abajo. Una vez que la articulación haya alcanzado la posición deseada, se debe soltar los dedos de la pantalla y seleccionar en una zona en donde no se encuentren objetos de AR para deseleccionar la pieza. Después de unos segundos de terminar el movimiento, se observará que el robot físico irá a la posición indicada dentro de la aplicación. En el terminal de la Figura 9 se pueden observar las coordenadas recibidas desde la aplicación, las cuales a su vez se cargan al robot a través del nodo suscriptor de la articulación que corresponda. Además, mediante el visualizador RViz, el usuario puede obtener una retroalimentación adicional sobre el comportamiento real del robot, lo cual es efectivo para casos de teleoperación.

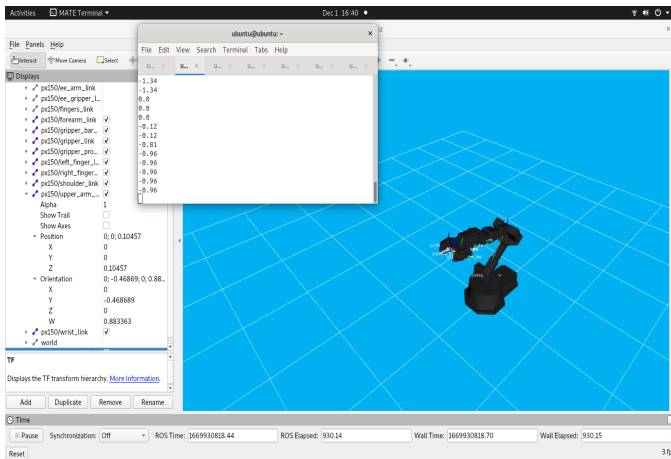


Figura 9: Mensajes recibidos y retroalimentación desde RViz en el entorno ROS.

IV. RESULTADOS

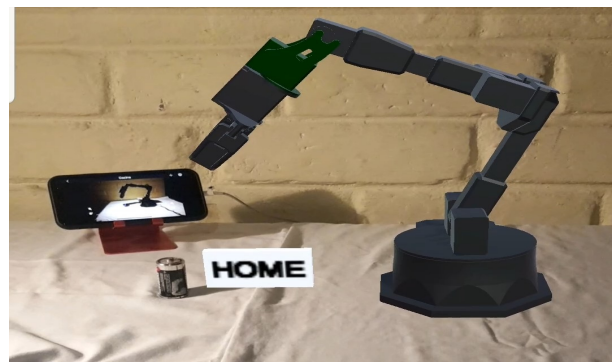
Una vez con la conexión entre las plataformas de Unity y ROS efectuada, se procedió a implementar el sistema completo para evaluar y determinar los parámetros más adecuados de funcionamiento. Entre estos parámetros están: el tiempo de respuesta entre el envío de coordenadas desde la interfaz AR y la ejecución del movimiento en el robot manipulador, el análisis de los parámetros cinemáticos del robot (margen de error de posiciones y trayectorias reales y teóricas), y la adaptabilidad del usuario a la interfaz creada para evaluar la efectividad de la interacción humano-robot. Para comprobar estos parámetros, se efectuaron tres pruebas que serán descritas a continuación.

Para el primer parámetro, se midió el tiempo que tarda el robot manipulador en ejecutar el movimiento asignado por la aplicación AR. Gracias al script utilizado para los publicadores, el sistema es capaz de controlar el intervalo de tiempo en el que la información se transmite desde la aplicación AR hacia ROS. Por ello, al aumentar o disminuir dicho valor, el tiempo que tarda el comando enviado en hacer efecto sobre el robot físico será mayor o menor, respectivamente.

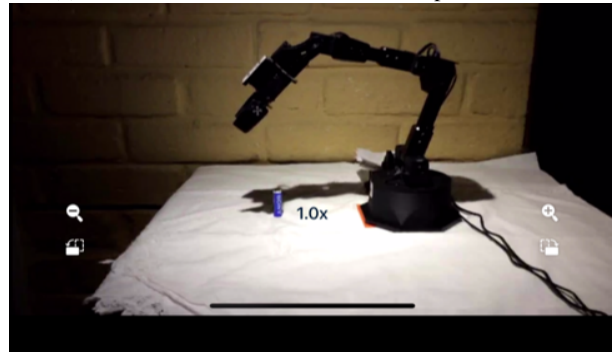
Después de realizar las debidas pruebas con el sistema, se pudo determinar que la frecuencia óptima de envío de mensajes rodeaba intervalos de entre 2 y 3 segundos; por debajo de este intervalo, el mensaje experimentaba un retardo significativo, mientras que por encima del mismo, la velocidad de actualización de los datos era elevada como para que se asignen las posiciones deseadas en el robot. Dentro de este tiempo, el robot obtenía una respuesta casi inmediata a las órdenes dadas por la interfaz AR. Sin embargo, como el sistema está previsto para ser probado en algunos usuarios con poca experiencia en robótica, se creyó conveniente asignar una frecuencia de 3 segundos en el envío de coordenadas, para precautelar accidentes o daños que puedan ocasionar movimientos bruscos o indebidos realizados por el usuario en la aplicación AR. Para evaluar los parámetros cinemáticos del robot, se analizó y comparó las posiciones y trayectorias

teóricas con respecto a las reales del robot manipulador. Los valores teóricos son aquellos que se envían desde la interfaz AR y aparecen en cada terminal de los nodos suscriptores en ROS, mientras que los reales son los que se registran del movimiento del robot físico y se reflejan en la herramienta de visualización RViz.

Para efectuar la comprobación de trayectorias, se llevó a cabo la tarea de «Pick & Place» con los miembros del equipo de investigación como usuarios de la interfaz AR. Se colocó como referencia un objeto real a una distancia aproximada tanto del robot físico como del modelo virtual, como se muestra en la Figura 10. Una vez en posición, se procedió a agarrar el objeto utilizando la aplicación AR; con esto se pudo apreciar que los movimientos realizados por el robot tienen un alto nivel de precisión con respecto a los movimientos realizados con la aplicación.



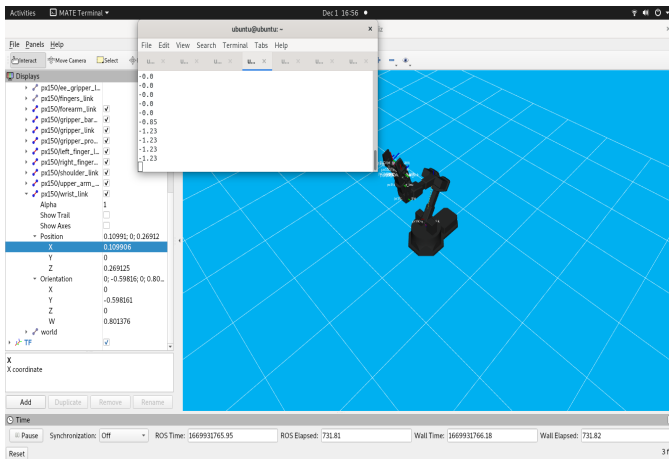
(a) Control del robot virtual desde la aplicación AR.



(b) Respuesta del robot físico a los comandos enviados.

Figura 10: Resultados comprobación de trayectorias.

Por otro lado, para determinar el margen de error entre las posiciones enviadas desde la aplicación y las asignadas en el robot real, se recolectaron 10 muestras por articulación, una por cada posición asignada. De esta manera, se pudo obtener información más detallada sobre el movimiento de las 5 articulaciones del robot; en la Figura 11 se observa un ejemplo de las acciones efectuadas sobre la articulación «wrist» durante la prueba. De esta manera, por cada cambio de posición se tomó la información tanto del terminal correspondiente a la articulación trabajada como del visualizador RViz, el cual otorga un seguimiento a tiempo real y preciso de los movimientos que ejecuta el robot manipulador.



(a) Movimiento articulación «wrist».

Figura 11: Pruebas de movimiento realizadas en todas las articulaciones del robot.

En las tablas **I**, **II**, **III**, **IV** y **V** se observa una comparación entre los valores enviados por la aplicación vs. los valores reales obtenidos desde RViz por cada articulación. Es importante recalcar que todos los ángulos fueron tomados en radianes y se realizó el redondeo a 2 decimales para obtener una comparación más precisa. Los ejes de rotación considerados para cada articulación son: para la articulación «shoulder», se tomó el eje «Z»; para «upperarm», «forearm» y «wrist», el eje «Y»; finalmente para la articulación «gripper», se consideró el eje «X».

Comparación posiciones «Shoulder»	
Posición teórica	Posición real
0	0
-0.9	-0.89
-1.28	-1.27
-0.31	-0.31
0.39	0.39
0.88	0.86
1.54	1.52
0.59	0.59
0.43	0.43
-0.47	-0.47

Tabla I: Posiciones teóricas vs. reales: «Shoulder».

Comparación posiciones «Upperarm»	
Posición teórica	Posición real
-1.14	-1.16
-1.34	-1.36
-0.96	-0.98
-0.62	-0.62
-0.4	-0.4
-0.16	-0.14
0.07	0.1
0.37	0.41
0.49	0.54
0.63	0.69

Tabla II: Posiciones teóricas vs. reales: «Upperarm».

Comparación posiciones «Forearm»	
Posición teórica	Posición real
0	0.06
-0.39	-0.28
-0.93	-0.84
-1.18	-1.11
-0.59	-0.54
0.27	0.32
0.56	0.6
0.73	0.77
0.14	0.22
-0.82	-0.72

Tabla III: Posiciones teóricas vs. reales: «Forearm».

Comparación posiciones «Wrist»	
Posición teórica	Posición real
0	-0.07
-1.23	-1.28
-0.91	-0.98
-0.68	-0.75
-0.45	-0.52
0.22	0.15
0.49	0.42
0.7	0.63
0.95	0.88
1.34	1.27

Tabla IV: Posiciones teóricas vs. reales: «Wrist».

Comparación posiciones «Gripper»	
Posición teórica	Posición real
0	0
-0.46	-0.45
-1.07	-1.06
-1.41	-1.41
-0.47	-0.48
0.57	0.57
0.85	0.84
1.32	1.31
1.36	1.35
0.43	0.44

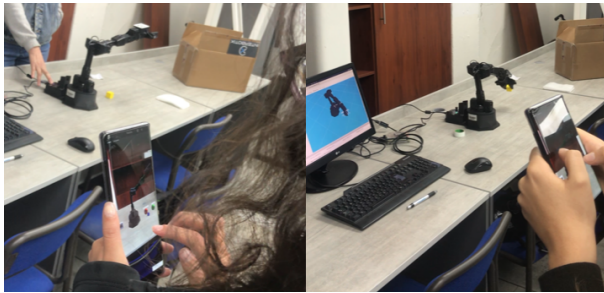
Tabla V: Posiciones teóricas vs. reales: «Gripper».

Finalmente, para comprobar el tercer y último parámetro de funcionamiento del proyecto, que corresponde a evaluar la interacción humano-robot a través del uso de la interfaz AR diseñada, se procedió a realizar pruebas en usuarios para evaluar su respuesta y adaptabilidad frente al sistema. A continuación, se detalla en qué consistió la prueba realizada y se describen las condiciones bajo las cuales se efectuó.

Como grupo objetivo para realizar la prueba de la aplicación AR se tomó a estudiantes de diferentes ciclos y a docentes, ambos de la carrera de la Escuela de Ingeniería Electrónica de la Universidad del Azuay, de entre 18 y 35 años. La mayoría de los usuarios que participaron en la prueba poseían conocimientos bajos o intermedios sobre robótica y no tenían mucha experiencia previa con el control de robots, a excepción de los docentes quienes fueron los más experimentados. La prueba como tal consistió en que los usuarios debían completar la tarea «Pick & Place» (tomar un objeto y colocarlo en un sitio distinto) controlando el robot manipulador PincherX 150 desde la aplicación AR desarrollada. Una vez completada la

tarea, los usuarios llenaron una encuesta de 10 preguntas en total para obtener resultados objetivos sobre su experiencia manejando el robot a través de la interfaz AR.

Para efectuar la prueba, los equipos de hardware se instalaron dentro de un laboratorio de la universidad, en una mesa se montó el robot manipulador y se dejó un espacio libre para que los usuarios puedan colocar el modelo virtual. El robot manipulador se colocó en una zona sin obstáculos alrededor por precaución para evitar golpes o daños resultantes de movimientos bruscos o indebidos por parte de los usuarios. Como objetos para realizar la tarea «Pick & Place», se utilizaron dos cubos de madera de 1g cada uno, sirviendo el un cubo como referencia para que el usuario sepa la ubicación aproximada del objeto dentro de la aplicación AR.



(a) Usuarios realizando la tarea «Pick & Place».



(b) Manejo de la aplicación AR.

Figura 12: Prueba de la interfaz AR en usuarios.

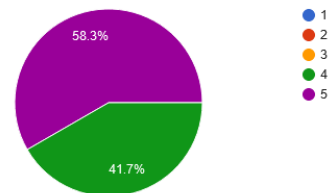
Al inicio, algunos usuarios experimentaron un poco de dificultad para realizar los gestos correctos sobre la pantalla del celular para que el robot efectúe los movimientos deseados. Sin embargo, después de una pequeña intervención del equipo de investigación, los usuarios no tuvieron problema en controlar el robot y completar la actividad. Es importante mencionar que como parte de las indicaciones iniciales, también se recomendó primero mover el robot ligeramente antes de acercarlo hacia el cubo por motivos de seguridad. A continuación, se exponen los resultados obtenidos de las encuestas realizadas. Para facilidad de visualización, las preguntas fueron agrupadas de acuerdo a la temática que trataban, de modo que se dividieron en 3 grupos: identificación de elementos en la interfaz, interacción con la interfaz y utilidad del proyecto.

Es importante recalcar que para las preguntas en las que se requirió de calificación, 5 corresponde a la calificación más alta siendo las designaciones de nivel de la siguiente manera: 1, totalmente difícil o muy bajo; 2, difícil o bajo; 3, medianamente sencillo o medio; 4, sencillo o alto; 5, totalmente sencillo o muy alto. Con estas asignaciones se debe realizar la valoración de resultados. Como se mencionó anteriormente, los usuarios que participaron en el proyecto tienen entre 18 y 35 años y su experiencia con robótica no es tan avanzada, puesto a que casi la mitad de los encuestados no están familiarizados con estos sistemas.

Con respecto a la distribución e identificación de los elementos visuales de la aplicación, como las articulaciones individuales del robot y los botones virtuales, ningún usuario presentó dificultades al diferenciar e interactuar con los objetos virtuales. Según los resultados expuestos en la Figura 13, a todos los usuarios les pareció sencillo tanto la selección de elementos como la apreciación del movimiento del robot virtual dentro de la interfaz, pues ninguna respuesta de las preguntas de este grupo tiene una calificación menor a 3.

En una escala del 1 al 5, siendo 1 el menor valor, ¿qué tan sencillo fue seleccionar las piezas del robot dentro de la aplicación?

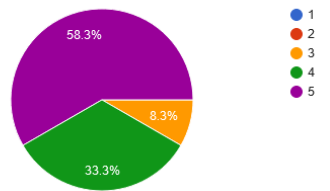
12 respuestas



(a) Selección de elementos en la interfaz AR.

En una escala del 1 al 5, siendo 1 el menor valor, ¿qué tan sencillo fue apreciar las posiciones de cada pieza del robot a través de la aplicación?

12 respuestas



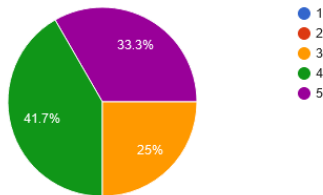
(b) Apreciación de movimientos del robot virtual.

Figura 13: Identificación de elementos en la interfaz AR.

Los resultados de las preguntas de la Figura 14 reflejan el nivel de interacción humano-robot que ofrece el sistema propuesto para el presente proyecto. En las preguntas de la Figura 14a se evidencia que los usuarios tuvieron una ligera dificultad para completar la actividad «Pick & Place» ya que el 25 % calificó como nivel 3 (medianamente sencillo); una situación similar ocurre en la Figura 14b, en donde el 8.3 % de los encuestados calificó con 3 al nivel del control que sintieron sobre el robot. Esto, según las observaciones realizadas, se

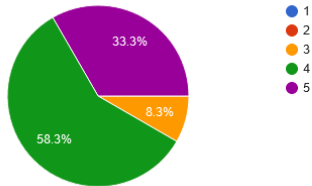
debió a que ciertas indicaciones sobre los controles de movimiento en la aplicación no estuvieron muy claras en un inicio. Sin embargo, luego de una pequeña demostración dada por el equipo investigador, los usuarios lograron completar la tarea sin problema.

En una escala del 1 al 5, siendo 1 el menor valor, ¿qué tan fácil le resultó completar la tarea Pick and Place (tomar un objeto de un sitio y posicionarlo en otro)?
12 respuestas



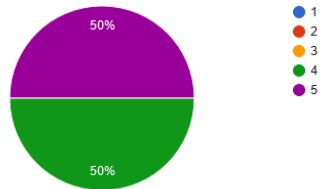
(a) Completar la prueba con éxito.

En una escala del 1 al 5, siendo 1 el menor valor, ¿qué nivel de control sintió usted que tenía sobre el robot mientras utilizaba la aplicación?
12 respuestas



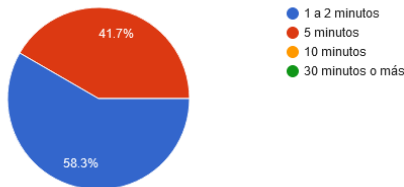
(b) Control sobre el robot manipulador.

En una escala del 1 al 5, siendo 1 el menor valor, ¿qué tan sencillo le pareció el uso de la aplicación para mover el robot?
12 respuestas



(c) Ejecución de movimientos sobre el robot.

¿Qué tiempo aproximadamente le tomó aprender a usar la aplicación para mover con éxito el robot?
12 respuestas



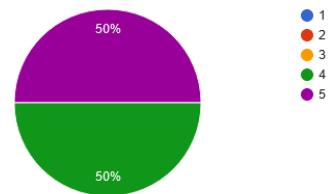
(d) Tiempo de aprendizaje para el uso de la aplicación.

Figura 14: Interacción con la interfaz AR.

Por otro lado, los resultados para las preguntas de las Figuras 14c y 14d, demostraron la efectividad del sistema diseñado para mejorar la interacción humano-robot. En la Figura 14c se observa que la calificación es mayor o igual a 4, lo cual indica que los usuarios tuvieron la facilidad para entender el comportamiento de un robot manipulador y fueron capaces de manejarlo sin tener mayor experiencia utilizando controladores dedicados a robótica. Además, es importante hacer énfasis en los resultados de la pregunta de la Figura 14d; aquí es posible demostrar que la aplicación ofrece una interfaz lo suficientemente intuitiva como para que un usuario cualquiera pueda familiarizarse con el manejo de robots, sin considerar sus conocimientos o experiencia en robótica.

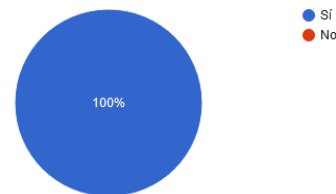
Finalmente, en la Figura 15 se pueden observar las dos preguntas finales que definen el nivel de satisfacción de los usuarios al utilizar la aplicación desarrollada. A partir de la Figura 15a se puede evidenciar que en general todos los usuarios tuvieron una experiencia favorable ante el uso de la aplicación, pues no tuvieron mayores inconvenientes para aprender a manejarla. Además, el 100% de los encuestados consideran que el presente proyecto es viable para su desarrollo en el campo ya sea laboral o investigativo, debido a la facilidad de uso y accesibilidad para el usuario.

Después de las indicaciones dadas para usar aplicación, siendo 1 el menor valor, ¿qué tan fácil le pareció el uso general de la aplicación para mover el robot?
12 respuestas



(a) Experiencia con el uso de la app.

¿Cree que este sistema sería eficiente para implementarlo en el ámbito laboral o investigativo?
12 respuestas



(b) Aplicación del proyecto en el campo.

Figura 15: Satisfacción de usuarios.

V. CONCLUSIONES

La AR es una tecnología en auge cuyas aplicaciones en los distintos campos como la industria, la salud o la educación, tiene mucho que ofrecer. Por otro lado, la evolución de la robótica en los últimos años se ha acelerado a nivel

exponencial, de modo que poco a poco juega un papel cada vez más importante en nuestra vida diaria. Por esta razón, es imperativo buscar herramientas que ayuden a mejorar la interacción humano-robot y permitan que cada vez más personas se familiaricen con los sistemas robóticos. Por ello, la presente investigación tuvo como objetivo el diseño e implementación de una interfaz AR capaz de controlar a distancia un robot manipulador a tiempo real.

Con la ayuda de la plataforma de desarrollo de videojuegos Unity y sus paquetes dedicados al desarrollo AR, se desarrolló una aplicación AR para el sistema operativo Android cuyo propósito fue generar un modelo virtual del robot manipulador PincherX 150 capaz de replicar todos los movimientos y acciones realizados por el robot real. El aplicativo funcionó como un controlador para el robot de modo que al modificar las posiciones del robot generado virtualmente, automáticamente el brazo robótico real reciba dicha información y reproduzca con gran exactitud todos los movimientos efectuados. El diseño de la aplicación fue exitoso, pues fue posible visualizar y manipular fácilmente el modelo virtual del robot, lo cual dio al usuario una sensación de presencia del robot manipulador real.

Con respecto a los equipos de hardware utilizados, la Raspberry Pi 4 funcionó adecuadamente para los propósitos del proyecto, pues no presentó problemas de eficiencia durante la ejecución de nodos ROS o del simulador RViz. Sin embargo, para aplicaciones de mayor alcance, se recomienda utilizar equipos con procesadores más robustos para dar más fiabilidad al sistema. Tras las pruebas de análisis de tiempo de respuesta y parámetros cinemáticos del robot, el equipo pudo concluir que el proyecto cumple con los parámetros de funcionamiento de un sistema tele-operado a tiempo real; con respecto al tiempo de respuesta del robot manipulador, el envío de coordenadas desde la interfaz AR hacia el robot fue constante y se comprobó que todas las posiciones publicadas por la aplicación fueron registradas y ejecutadas por los suscriptores del sistema. Por otro lado, con respecto a los aspectos cinemáticos del robot, se pudo observar que tanto las trayectorias como las posiciones teóricas vs. las reales tuvieron un margen de error mínimo, lo cual demuestra la correcta funcionalidad del proyecto.

Finalmente, con respecto a las pruebas realizadas en usuarios, fue posible verificar que el sistema propuesto mejora las capacidades de dominio de un usuario sobre un sistema robótico. De acuerdo a los resultados obtenidos de las encuestas, la interfaz AR es amigable e intuitiva con el usuario, lo que hace que se sienta más seguro de controlar y manejar robots manipuladores; todos los usuarios que participaron de la prueba dieron calificaciones muy favorables al sistema, lo cual sugiere que la interfaz AR es eficiente para que las personas puedan familiarizarse más con el control de robots manipuladores. Las preguntas de la Figura 4 que tratan sobre la interacción del usuario con el sistema, brindan información muy importante sobre la capacidad del sistema de mejorar la interacción humano-robot, ya que al estar todas sobre el nivel 3 de calificación, se demuestra la facilidad de un usuario, sin

considerar si es experimentado o no, para aprender rápidamente a usar la interfaz, a controlar un robot manipulador e incluso completar tareas con el mismo. Además, teniendo en cuenta que el tiempo máximo que tomó a una persona en asimilar el funcionamiento de la interfaz AR y del robot como tal fue de aproximadamente 5 minutos, se reitera el carácter amigable del sistema con los usuarios.

En lo que concierne al desempeño general del sistema propuesto, la interfaz AR cumplió con los objetivos establecidos por el equipo de investigación. Con respecto a la respuesta del robot frente a la aplicación, el robot fue capaz de captar todas las posiciones y movimientos que el usuario deseaba realizar con alta precisión. Se considera que el margen de error obtenido es aceptable para el sistema dentro de los parámetros establecidos por el equipo de investigación. Por otro lado, la forma en la que está diseñada la interfaz AR admite el uso del sistema con diferentes modelos de robots manipuladores, no únicamente con el utilizado en la presente investigación, siempre y cuando se posea el modelo 3D del robot a utilizar y se realicen modificaciones mínimas sobre el modelo en el entorno Unity. Sin embargo, es necesario analizar que el robot manipulador no contenga defectos físicos o mecánicos que afecten al posicionamiento del mismo y verificar que la conexión inalámbrica se dé de forma ininterrumpida antes de poner el sistema en funcionamiento. Además, se considera re-evaluar el diseño de la interfaz de usuario de la aplicación para añadir o modificar elementos que puedan dar al usuario una mejor experiencia al controlar un robot manipulador con el sistema propuesto.

La integración de tecnologías AR en la robótica tiene la capacidad de mejorar el aprendizaje y entendimiento del funcionamiento de sistemas robóticos, de modo que personas con mucha o poca experiencia puedan manejar estos sistemas sin dificultad, como se pudo evidenciar durante las pruebas realizadas para este proyecto. A partir de los resultados obtenidos, podemos concluir que el enfoque de AR hacia la robótica efectivamente ayuda a disminuir la brecha que existe entre humanos y máquinas, puesto a que esta tecnología es capaz de brindar al usuario una mejor retroalimentación de los movimientos de un robot manipulador que cualquier otro sistema de control actual.

REFERENCIAS

- [1] A. O. Baturone, *Robótica: manipuladores y robots móviles*. Marcombo, 2005.
- [2] G. Michalos, P. Karagiannis, S. Makris, O. Tokcalar, and G. Chryssolouris, "Augmented reality (AR) applications for supporting human-robot interactive cooperation", *Procedia CIRP*, vol. 41, pp. 370–375, 2016.
- [3] E. Oliver, «Con esta tecnología podrás meterte en la cabeza de un robot». [En línea] Disponible en: <https://es.digitaltrends.com/realidad-virtual/tecnologia-mit-robots/> (accedido: enero, 2023).
- [4] Open Robotics, «Robot operating system». [En línea] Disponible en: <https://www.ros.org/> (accedido: enero, 2023).
- [5] C. Nuzzi, S. Ghidini, R. Pagani, S. Pasinetti, G. Coffetti, and G. Sansoni, «Hands-free: a robot augmented reality teleoperation system», in *2020 17th International Conference on Ubiquitous Robots (UR)*. IEEE, 2020, pp. 617–624.
- [6] D. Sánchez Martínez, «Teleoperación del robot Pepper con métodos de realidad aumentada», 2020.

- [7] J. Pareja Muñoz, «Rediseño, sensorización y teleoperación mediante realidad aumentada de kyma, un robot manipulador blando», 2018.
- [8] M. E. Walker, H. Hedayati, and D. Szafir, «Robot teleoperation with augmented reality virtual surrogates», in *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2019, pp. 202–210.
- [9] H. Hedayati, M. Walker, and D. Szafir, «Improving collocated robot teleoperation with augmented reality» in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 2018, pp. 78–86.
- [10] C. Li, A. Fahmy, and J. Sienz, «An augmented reality based human-robot interaction interface using kalman filter sensor fusion», *Sensors*, vol. 19, no. 20, p. 4586, 2019.
- [11] J. Iranzo Rodrigo, «Control de sistema de realidad aumentada accionado mediante un guante», Ph.D. dissertation, Universitat Politècnica de València, 2017.
- [12] J. M. Gregory, C. Reardon, K. Lee, G. White, K. Ng, and C. Sims, «Enabling intuitive human-robot teaming using augmented reality and gesture control» *arXiv preprint arXiv:1909.06415* 2019.
- [13] M. Hernández-Ordoñez, M. A. Nuño-Maganda, C. A. Calles-Arriaga, O. Montaña-Rivas, and K. E. Bautista Hernández, «An education application for teaching robot arm manipulator concepts using augmented reality», *Mobile Information Systems*, vol. 2018, 2018.
- [14] L. Manring, J. Pederson, D. Potts, B. Boardman, D. Mascarenas, T. Harden, and A. Cattaneo, «Augmented reality for interactive robot control», in *Special Topics in Structural Dynamics & Experimental Techniques, Volume 5*. Springer, 2020, pp. 11–18.
- [15] D. Bambašek, Z. Materna, M. Kapinus, V. Beran, and P. Smrž, «Combining interactive spatial augmented reality with head-mounted display for end-user collaborative robot programming» in *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2019, pp. 1–8.
- [16] Trossen Robotics, «PincherX 150 robot arm». [En línea] Disponible en: <https://www.trossenrobotics.com/pincherx-150-robot-arm.aspx> (accedido: marzo, 2022).
- [17] Google, «Arcore supported devices». [En línea] Disponible en: <https://developers.google.com/ar/devices> (accedido: enero, 2022).
- [18] Raspberry Pi, «Raspberry Pi 4 Computer Model B Product Brief». (enero, 2021). [En línea] Disponible en: <https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-product-brief.pdf> (accedido: enero, 2022).
- [19] Unity Technologies, «Unity User Manual 2020.3». (2022). [En línea] Disponible en: <https://docs.unity3d.com/2020.3/Documentation/Manual> (accedido: julio, 2022).
- [20] A. Dattalo, «Wiki: Ros/introduction». (agosto, 2018). [En línea] Disponible en: <http://wiki.ros.org/ROS/Introduction> (accedido: diciembre, 2022).
- [21] Unity Community, «Unity robotics hub». (febrero, 2022). [En línea] Disponible en: <https://github.com/Unity-Technologies/Unity-Robotics-Hub><https://github.com/Unity-Technologies/Unity-Robotics-Hub> (accedido: septiembre, 2022).