



**UNIVERSIDAD TÉCNICA DE AMBATO**

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E  
INDUSTRIAL**

**CARRERA DE INGENIERÍA INDUSTRIAL EN PROCESOS DE  
AUTOMATIZACIÓN**

**Tema:**

---

**“DISEÑO DE SISTEMAS DE CONTROL INDUSTRIAL DE ROBOTS  
BASADOS EN INDUSTRIA 4.0”**

---

Trabajo de Titulación Modalidad: Proyecto de Investigación, presentado previo a la obtención del título de Ingeniero Industrial en Procesos de Automatización.

**ÁREA:** Industrial y Manufactura

**LÍNEA DE INVESTIGACIÓN:** Tecnología de la Información y Sistemas de Control

**AUTOR:** Pablo Josué Bonilla Vásquez

**TUTOR:** Dr. Marcelo Vladimir García Sánchez, Mg.

Ambato – Ecuador

Abril 2020

## **APROBACIÓN DEL TUTOR**

En calidad de tutor del Trabajo de Titulación con el tema: “DISEÑO DE SISTEMAS DE CONTROL INDUSTRIAL DE ROBOTS BASADOS EN INDUSTRIA 4.0”, desarrollado bajo la modalidad de proyecto de investigación por el señor Pablo Josué Bonilla Vásquez, estudiante de la Carrera de Ingeniería Industrial en Procesos de Automatización, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 15 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y el numeral 7.4 del respectivo instructivo.

Ambato, agosto 2020.



Firmado electrónicamente por:  
**MARCELO  
VLADIMIR GARCIA  
SANCHEZ**

-----  
Dr. Marcelo Vladimir García Sánchez, Mg.

**TUTOR**

## AUTORÍA

El presente Proyecto de Investigación titulado: “DISEÑO DE SISTEMAS DE CONTROL INDUSTRIAL DE ROBOTS BASADOS EN INDUSTRIA 4.0” es absolutamente original, auténtico y personal. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, agosto 2020.



---

Pablo Josué Bonilla Vásquez

1804568622

AUTOR

## **APROBACIÓN TRIBUNAL DE GRADO**

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor Pablo Josué Bonilla Vásquez, estudiante de la Carrera de Ingeniería Industrial en Procesos de Automatización, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad de proyecto de investigación, titulado : “DISEÑO DE SISTEMAS DE CONTROL INDUSTRIAL DE ROBOTS BASADOS EN INDUSTRIA 4.0”, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 17 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y al numeral 7.6 del respectivo instructivo. Para cuya constancia suscribimos, conjuntamente con la señora presidenta del Tribunal.

Ambato, agosto 2020.



Firmado electrónicamente por:  
**ELSA PILAR  
URRUTIA**

-----  
Ing. Pilar Urrutia, Mg.  
**PRESIDENTA DEL TRIBUNAL**



Firmado electrónicamente por:  
**SANTIAGO MAURICIO  
ALTAMIRANO  
MELENDEZ**

-----  
Ing. Santiago Altamirano, Mg.  
**PROFESOR CALIFICADOR**

Digitally signed by FRANKLIN  
WILFRIDO SALAZAR LOGRONO  
Date: 2020.07.29 11:47:03 COT

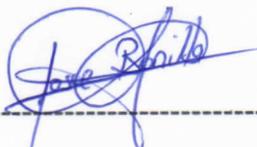
-----  
Ing. Franklin Salazar, Mg.  
**PROFESOR CALIFICADOR**

## **DERECHOS DE AUTOR**

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación en favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, agosto 2020.



---

Pablo Josué Bonilla Vásquez

1804568622

AUTOR

## **DEDICATORIA:**

*Con el más grande respeto y admiración que les tengo quiero dedicar este trabajo a mi Padre JORGE y a mi madre GLADYS ya que gracias al esfuerzo, sacrificio, dedicación y sudor me han dado todo lo necesario para alcanzar mis metas, han forjado en mi un espíritu de lucha, de valores y principios para ser una persona de bien, enseñándome el valor del trabajo y el esfuerzo para la consecución de logros*

*Pablo Josué Bonilla Vásquez*

## **AGRADECIMIENTO:**

*Principalmente a **Dios**, por todas las bendiciones que he recibido a lo largo de esta carrera, porque me ha dado la sabiduría, fuerza y valentía para afrontar todas las adversidades que en el camino hasta mi objetivo se presentaron.*

*A mis **Padres**, por su apoyo incondicional, por depositar en mí su confianza y esfuerzo y nunca claudicar ni en los peores momentos, por guiarme en el camino corriendo mis errores.*

*Al **Dr. Marcelo García**, por el apoyo durante la investigación, por incentivar me hacia el camino de la investigación y plantearme retos que en un principio parecían imposibles de lograr, pero con la guía brindada se los pudo alcanzar.*

*Pablo Josué Bonilla Vásquez*

## Índice de contenidos

A.	PÁGINAS PRELIMINARES .....	ii
B.	DESARROLLO .....	1
	CAPÍTULO I. MARCO TEÓRICO .....	1
	1.1. Antecedentes Investigativos .....	1
	1.1.1. Contextualización del Problema .....	1
	1.1.2. Estudio del Arte.....	2
	1.1.3. Fundamentación Teórica .....	5
	1.2. Objetivos .....	19
	CAPÍTULO II. METODOLOGÍA.....	20
	2.1. Materiales .....	20
	2.2. Métodos .....	31
	2.2.1. Modalidad de Investigación.....	31
	2.2.2. Recolección de Información .....	32
	2.2.3. Procesamiento y Análisis de Datos.....	32
	2.2.4. Desarrollo del Proyecto .....	33
	CAPÍTULO III. RESULTADOS Y DISCUSIÓN .....	33
	3.1. Análisis y discusión de los resultados .....	33
	3.1.1. Desarrollo de la propuesta .....	33
	3.1.2. Análisis y Discusión de los Resultados .....	54
	CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES .....	56
	4.1. Conclusiones .....	56
	4.2. Recomendaciones .....	58
C.	MATERIALES DE REFERENCIA .....	59
	Referencias Bibliográficas.....	59
	Anexos.....	63

## Índice de Tablas

Tabla 1: Conceptos Ligados a los robots .....	6
Tabla 2: Comparación entre tarjetas controladoras.....	11
Tabla 3: Comparación entre softwares de desarrollo virtual .....	13
Tabla 4: Comparación entre protocolos de comunicación.....	16
Tabla 5:Tabla Comparativa Gafas Inteligentes.....	17
Tabla 6: Especificaciones técnicas Scrobot ER 4U .....	21
Tabla 7: Estructura Brazo manipulador Scrobot ER 4U.....	22
Tabla 8:Valoración Entre Tarjetas SBC.....	24
Tabla 9:Valoración software de desarrollo virtual.....	26
Tabla 10: Valoración protocolos de comunicación.....	29
Tabla 11: Escala de Valoración.....	30
Tabla 12: Matriz de selección programa CAD .....	30
Tabla 13: Parámetros Denavit - Hartenberg.....	36
Tabla 14: Descripción de las clases movimiento Scrobot Er 4U .....	45
Tabla 15: Descripción de las clases comunicación .....	47
Tabla 16: Descripción de las clases principal .....	49
Tabla 17: Descripción de las clases Control GPIOs.....	50

## Índice de figuras

Figura N° 1: Arquitectura robot articulado .....	7
Figura N° 2: Cinemática inversa y directa de un manipulador .....	8
Figura N° 3: Tarjeta Raspberry Pi 3.....	9
Figura N° 4: Tarjeta Odriod .....	10
Figura N° 5: Tarjeta Jaguar One .....	10
Figura N° 6: Tarjeta LattePanda .....	11
Figura N° 7: Estructura MQTT .....	14
Figura N° 8: Protocolo de Comunicación COAP .....	15
Figura N° 9: Estructura AMQP.....	16
Figura N° 10: Manipulador SCORBOT-ER 4u .....	22
Figura N° 11: Estructura Manipulador SCORBOT-ER 4u.....	23
Figura N° 12: Conector DB50 de Scorbob ER 4U .....	23
Figura N° 13: Kit Meta Development 2.....	28
Figura N° 14: Relación de la Cinemática Inversa y Directa. ....	34
Figura N° 15: Punto de Referencia .....	35
Figura N° 16: Geometría Scorbob ER 4U .....	35
Figura N° 17: Referencias para cinemática inversa.....	38
Figura N° 18: Esquema general de articulaciones .....	39
Figura N° 19: Análisis trigonométrico.....	39
Figura N° 20: Arquitectura de software y hardware .....	41
Figura N° 21: Diagrama de Componentes físicos sistema de control Scorbob ER 4U .....	41
Figura N° 22: Arquitectura Hardware Raspberry pi y Scorbob.....	42
Figura N° 23: Diagrama de componentes interfaz virtual .....	44
Figura N° 24: Diagrama de Clases Movimiento Scorbob .....	45
Figura N° 25: Diagrama de clases comunicación .....	47
Figura N° 26: Diagrama de clases escena principal.....	48
Figura N° 27: Diagrama de Clases “Control GPIOs” .....	50
Figura N° 28: Diagrama de Clase Conexión MQTT .....	51
Figura N° 29: Entorno Virtual .....	52
Figura N° 30: Puntero de referencia .....	53
Figura N° 31: Tomar y soltar el puntero de referencia .....	54

Figura N° 32: Latencia de red ..... 54  
Figura N° 33: Grafica Del Throughput ..... 55

## Resumen Ejecutivo

El crecimiento tecnológico, y los últimos avances de la ciencia enfocadas a la industria ha ido creciendo hasta la aparición del término “Industria 4.0” el mismo que se ha convertido en sinónimo de innovación a nivel de producción y control industrial, en los últimos años la automatización industrial ha tenido un crecimiento paulatino, de allí se genera la necesidad de crear y desarrollar sistemas de control capaces de realizar actividades complejas de una manera eficiente logrando así cumplir con las exigencias industriales.

La incorporación de sistemas robóticos a los procesos de producción ha dado lugar a la reducción de costos en producción y mejora de la calidad en los productos sin embargo conforme avanzan las tecnologías se requiere de sistemas de control mucho más complejos y de costos altos, de allí nace la necesidad de crear sistemas ciber físicos de bajo costo y con sistemas de control eficientes, pero de una complejidad baja a la hora de operarlos.

Por estas razones, la investigación realizada presenta un sistema de control para el brazo manipulador Scorbot ER 4U haciendo uso de tarjetas empotradas de bajo costo y utilizando dispositivos y tecnologías innovadoras como es la realidad aumentada que brinda a los usuarios una experiencia real con complementos virtuales, facilitando así el entrenamiento y el manejo de equipos robóticos.

El sistema mencionado presenta un ensamble físico de un controlador para el brazo Scorbot ER 4U haciendo uso de un microcontrolador de bajo costo y de software libre basado en debían, y un diseño virtual para el control del brazo, diseñado en un motor gratuito como es Unity 3D y haciendo uso de un dispositivo de realidad aumentada permitiendo así que la aplicación sea mucho mas interactiva y de gran facilidad para el usuario.

Para la validación del sistema se realizaron pruebas de funcionamiento, dando como resultado movimientos con gran precisión y de una manera muy sencilla sin tener mayor complejidad a la hora de indicar posiciones a las cuales debe llegar el manipulador.

**Palabras Clave:** Industria 4.0, Sistemas CPS, Robótica, Sistemas Empotrados, Realidad Aumentada, Unity 3D

## **Abstract**

Technological growth, and the latest advances in science focused on industry has been growing until the appearance of the term “Industry 4.0”, which has become synonymous with innovation at the level of production and industrial control, in recent years the Industrial automation has had a gradual growth, hence the need to create and develop control systems capable of carrying out complex activities in an efficient manner, thus complying with industrial requirements.

The incorporation of robotic systems into production processes has resulted in reduced production costs and improved product quality, however, as technologies advance, much more complex and high cost control systems are required, hence The need arises to create physical cyber systems of low cost and with efficient control systems, but of a low complexity when operating them.

For these reasons, the research carried out presents a control system for the Scorbot ER 4U manipulator arm using low-cost embedded cards and using innovative devices and technologies such as augmented reality that gives users a real experience with virtual add-ons, thus facilitating the training and handling of robotic equipment.

The mentioned system presents a physical assembly of a controller for the Scorbot ER 4U arm using a low-cost microcontroller and free software based on debian, and a virtual design for arm control, designed on a free motor such as Unity 3D and making use of an augmented reality device thus allowing the application to be much more interactive and very easy for the user.

For the validation of the system, performance tests were carried out, resulting in movements with great precision and in a very simple way without having greater complexity in indicating positions to which the manipulator should reach.

**Keywords:** Industry 4.0, CPS Systems, Robotics, Embedded Systems, Augmented Reality, Unity 3D

## **B. DESARROLLO**

### **CAPÍTULO I. MARCO TEÓRICO**

#### **1.1. Antecedentes Investigativos**

##### **1.1.1. Contextualización del Problema**

En la actualidad la evolución de los sistemas productivos son la base de la competitividad, mejoramiento y desarrollo económico y social de las organizaciones industriales a nivel mundial. La fusión de la automatización industrial y la informática han dado surgimiento a la innovación técnica conocida como Industria 4.0[1]. La industria 4.0 se caracteriza por una dependencia creciente sobre la automatización e interconexión de sistemas debido a la necesidad de aumentar la eficiencia, autonomía y personalización de procesos a un bajo costo, y así, los sistemas robóticos basados en sistemas 4.0 se convierten en una importante herramienta que tiene un incremento en la demanda año tras año dentro de la industria mundial [2].

Durante los últimos años, la robótica ha tenido como objetivo encontrar soluciones a las necesidades de todos los campos en los que se desenvuelve el humano. La evolución de los campos de aplicación y su sofisticación han influenciado los temas de investigación en la comunidad de robótica. Esta evolución ha estado dominada por las necesidades humanas principalmente en trabajos que son de alto riesgo para el hombre, de allí surge la idea de reemplazarlos con robots industriales [3].

En el contexto actual del mundo tecnológico, los sistemas robóticos han tenido un avance importante en cuanto a costo e implementación, con la aparición de las tarjetas SBC de bajo costo se puede desarrollar sistemas de control eficientes para automatismos robóticos a precios realmente bajos en comparación a los sistemas que se manejan en el mercado de la robótica y automatización. Sin embargo, y a pesar de ser un gran avance estas nuevas tecnologías han tenido lugar solo en las grandes potencias mundiales, existe un limitado conocimiento de las mismas en países menos industrializados.

Actualmente en América latina la utilización de sistemas robóticos y en particular manipuladores es una necesidad que va adquiriendo fuerza dentro de la industria, ya que, los mismos permiten realizar tareas con mayor fuerza, precisión y rapidez además se adaptan fácilmente a cualquier actividad siendo los más aptos para tareas peligrosas

en las cuales se pone en riesgo la integridad de las personas, todo esto hace que se deba tener un control preciso para su correcto funcionamiento, a pesar de ser una gran necesidad para el desarrollo de la industria, los altos precios en el mercado de la robótica hace difícil que la mayoría de empresas puedan implementar uno de estos sistemas de automatización [4].

Dentro de la industria ecuatoriana existen un limitado número de empresas que han integrado a sus instalaciones manipuladores y robots para realizar sus tareas diarias, esto debido a dos factores importantes, en primer lugar, el precio para adquirir sistemas robóticos es elevado y segundo aún existe un escaso conocimiento hacia los sistemas que controlan estos robots, ya que, los sistemas de control que actualmente se presentan son complejos y utilizan hardware que tienen precios elevados [5].

La ingeniería industrial al ser la encargada de formar profesionales capaces de buscar soluciones a los problemas que afronta la industria en la actualidad, debe incentivar a los estudiantes a proponer ideas nuevas que aporten al sector industrial, en base a este argumento y a la problemática descrita surge la necesidad de diseñar sistemas de control más simples que faciliten el entendimiento y manipulación de los mismos, utilizando componentes de bajo costo, que es en si lo que ofrece la industria 4.0.

### **1.1.2. Estudio del Arte**

En los últimos años la industria se ha planteado nuevos desafíos para realizar procesos de producción procurando que estos sean más eficientes, autónomos y personalizables, este hecho condujo a la aparición de un nuevo concepto llamado Industria 4.0 y gracias a esta la industria en general se está adaptando a los nuevos sistemas tecnológicos para la creación de las llamadas "Fábricas inteligentes" [6].

Actualmente se habla de un futuro donde un usuario controla un robot virtual desde su teléfono o tableta que involucra también un movimiento en un robot físico. Esto no solo en la industria sino también en la vida cotidiana de las personas por ejemplo un usuario puede verificar el contenido de su refrigerador en el mundo virtual, y automáticamente ordenar comestibles arrastrando y soltando los productos deseados

en la nevera virtual, el efecto al final es llenar tanto la nevera virtual como la nevera física.

De igual forma un robot físico puede ser controlado directamente programando el movimiento al presionar teclas, con el antecedente que para controlar un robot virtual se lo hace de manera similar. Sin embargo, los robots físicos y virtuales pueden no reflejarse con precisión el uno al otro sin un usuario humano que ejecute las interacciones [7].

A continuación, se detalla investigaciones en las cuales se ha puesto en evidencia la importancia de utilizar sistemas en base a industria 4.0, para el control de articulaciones robóticas desde entornos virtuales utilizando tarjetas de bajo costo, así es lo que detalla la investigación de McLin en [7], donde se propone un enfoque para establecer interacciones bidireccionales entre robots en lo físico y mundos virtuales, permitiendo cambios que ocurren en un robot físico para ser reflejado automáticamente por su contraparte virtual, y viceversa, basada en esta investigación se puede establecer cuáles son los requisitos para crear un virtual que refleje el mundo físico.

Del mismo modo en la investigación desarrollada por Yoshiaki en [8], en donde explica acerca de una nueva arquitectura de software para un simulador de interacción humano-robot (HRI) en entornos virtuales, una plataforma de realidad virtual basada en la nube, llamada "SIGVerse", que reduce costos para desarrollar robots reales y experimentos de interacción en el mundo real con la ayuda de lo virtual. Esta información sustenta el entendimiento de cómo se generan las plataformas virtuales para generar movimientos en aparatos físicos logrando una sincronización entre ambos.

Investigaciones similares demuestran que se puede diseñar sistemas de control eficientes para manipuladores como es el caso del SCORBOT, así pues, lo expone Mota Muñoz en [9], el cual explica sobre la implementación de un controlador de punto de ajuste PD + G para la segunda y tercera articulación del robot SCORBOT ER-4u. Para lo cual se calculó los parámetros dinámicos del robot a través de mediciones de laboratorio y cálculos en un modelo CAD. El modelo dinámico fue obtenido por Euler método de Lagrange, en base a esto se puede realizar el estudio del

comportamiento del robot y como establecer los parámetros que se debe considerar para crear un controlador del manipulador Scorbobot.

En la investigación desarrollada por Michail G en [10], describe un método de control moderno para el robot SCORBOT ER-III en el artículo se menciona el vínculo entre la tecnología moderna, actualizada y equipos robóticos de edades más avanzadas, propone herramientas de programación más eficientes para la creación de interfaces más fáciles de usar que las que se presenta en entornos más antiguos. Esta investigación da las pautas de como integrar los últimos avances en programación para el diseño de interfaces intuitivas que permitan el control y manipulación del brazo Scorbobot.

Según los recientes avances en tecnología de hardware describe que se puede crear nuevas interfaces con Realidad Aumentada (AR) para el control y manipulación sistemas robóticos, así lo expone Courtney Hutton en [11] que se refiere a las nuevas tecnologías que están cada vez más presentes en actividades para robots orientados a tareas definidas, en la investigación explica la forma de comenzar a desarrollar un sistema AR en Unity para controlar robots, bajo el sistema simple de "apuntar el lugar al cual desplazarse".

Por otra parte, también se puede utilizar la realidad aumentada para la visualización de datos así lo explica Florian Leutert en [12] quien expone sobre la Realidad Aumentada (AR), como una forma de Realidad Mixta que es una técnica que mejora el entorno natural del usuario con información virtual útil adicional para ayudarlo en su tarea; se detalla la forma correcta de cómo debe integrarse y orientarse dentro del entorno la información, la cual debe ser dinámica, interactiva y en tiempo real en el espacio de trabajo del usuario. Bajo esta consideración se puede diseñar entornos de realidad eficientes, en donde existe un entorno virtual con soporte de realidad aumentada para la interacción con el usuario.

Finalmente, Tom Williams en [13] expone sobre la interacción humano robot basado en sistemas de realidad aumentada, avances recientes en realidad mixta, aumentada y virtual han permitido sugerir un método alternativo para mediar las interacciones humano-robot, permitiendo la comunicación de estas intenciones aprovechando el espacio físico humano como un lienzo compartido para él envió señales visuales. La

técnica se basa en la utilización la mano que se presenta como interactiva de dentro del entorno virtual creando así un Puntero (IHP) para controlar un robot en un espacio de trabajo físico real.

Con el sustento de las investigaciones mencionadas se hace factible el desarrollo del tema propuesto, ya que, apoyados en los temas desarrollados por los diferentes autores se puede desarrollar un método de control para el manipulador SCORBOT-ER 4U, desde un entorno virtual con soporte de realidad aumentada el cual sea interactivo e intuitivo y utilizando dispositivos de bajo costo como medios hardware para la comunicación física.

### **1.1.3. Fundamentación Teórica**

La implementación de un sistema de control para el manipulador SCORBOT-ER 4U, aborda la temática que se detalla a continuación:

## **INTRODUCCIÓN A LA ROBÓTICA**

La robótica tuvo sus inicios a comienzos del siglo XXI, fue en 1917 cuando Joseph Capek describió la palabra autómatas y en 1921 su hermano Karel Capek escribió la obra “Robots Universales de Rossum (RUR)” desde entonces el concepto de robótica entró a lenguaje mundial. El término robot se deriva de la palabra checa robota, que significa siervo o trabajador [14].

La robótica es la ciencia de la tecnología que estudia la construcción y el diseño de máquinas las cuales pueden realizar tareas con cierta lógica, siendo capaces de reemplazar la mano de obra del hombre. Para realizar estas tareas estas máquinas estas programadas con base en ciencias exactas y lógicas como la algebra, la mecánica, los autómatas programables entre otras.

En los últimos años la robótica ha ido creciendo en cuanto aplicaciones en los diferentes campos de la industrial como la medicina, la exploración planetaria y submarina, automatización de procesos industriales, seguridad, entretenimiento, entre otros. Uno de los aportes de mayor impacto se ha dado en la educación, donde los robots al integrarse al grupo de estudiantes y tutores, propician el aprendizaje y el fortalecimiento de habilidades cognitivas. [15]

## Robot

Un robot es un dispositivo electromecánico con múltiples grados de libertad, programable para realizar una variedad de tareas. En su gran mayoría estos robots adoptan la forma de un brazo humano con un artefacto o herramienta en su parte final, algunos también tienen una herramienta final en forma de muñeca. Normalmente su trabajo es realizar llevar a cabo actividades repetitivas, realizar grandes cargas, soldadura punto a punto entre otras [16]

Los grados de libertad o movilidad son la cantidad de movimientos independientes que puede hacer un dispositivo. Los conceptos ligados a un robot se describen en la tabla No.1

Tabla 1: Conceptos Ligados a los robots

<b>Manipulador</b>	<b>Dispositivo electromecánico capaz de interactuar con su entorno.</b>
<b>Efactor Final</b>	La herramienta, la pinza u otro dispositivo montado al final de un manipulador, para realizar tareas útiles.
<b>Espacio de Trabajo</b>	El espacio que puede alcanzar el efector final de un robot, tanto en posición como en orientación.
<b>Posición</b>	La ubicación traslacional (línea recta) de un objeto.
<b>Orientación</b>	La ubicación rotacional (angular) de un objeto.
<b>Articulación</b>	El dispositivo que permite el movimiento relativo entre dos enlaces en un robot.
<b>Cinemática</b>	El estudio del movimiento sin tener en cuenta las fuerzas que lo provocan.
<b>Dinámica</b>	El estudio del movimiento con respecto a las fuerzas que lo producen.

<b>Actuador</b>	Proporciona fuerza para el movimiento del robot.
<b>Sensor</b>	Lee variables reales en el movimiento del robot para usar en el control.

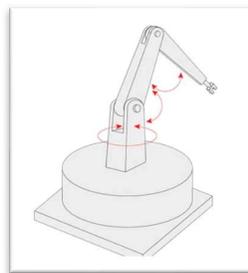
## Tipos de Robots

Existen un sin número de robots como son:

- Robots Cilíndricos
- Robots Transaccionales
- Robots Esféricos
- Robots Móviles
- Robots Humanoides
- Robots Insectos
- Robots Articulado [17].

De entre esta lista se enfatiza el estudio de los robots articulados ya que son los más utilizados en la industria.

**Robot articulado:** Los robots articulados se asemejan al brazo humano en su movimiento 3D (son antropomórficos). Ellos tienen tres articulaciones R, con tres ángulos variables teta 1, teta 2 y teta 3, que representan la cintura del cuerpo humano, articulaciones de hombro y codo. Son robots versátiles, pero tienen una cinemática y dinámica más compleja que los anteriores. Estos robots pueden usarse con una variedad de muñecas o herramienta final en el extremo de su último eslabón. Los robots articulados tienen la arquitectura mostrada en la figura No.1



**Figura N° 1:** Arquitectura robot articulado

## Cinemática de un Robot Articulado

La cinemática es el estudio analítico de la geometría de movimiento de un mecanismo con respecto a un sistema de coordenadas fijo, sin tener en cuenta las fuerzas o momentos que causan el movimiento. Para controlar y programar un robot debemos tener conocimiento de su disposición espacial y un medio de referencia al medio ambiente. La mecánica de un manipulador puede ser representado como una cadena cinemática de cuerpos rígidos conectados por articulaciones revolutas o prismáticas. Un extremo de la cadena está restringido a una base, mientras que un actuador final es montado en el otro extremo de la cadena [18]. La cinemática del robot se puede dividir en cinemática directa e inversa.

### Cinemática Directa

Un manipulador está compuesto por enlaces que se fijan entre sí mediante articulaciones ya sea revolutas o prismáticas desde la base hasta extremo final del robot. El cálculo de la posición y orientación del extremo final en términos de las variables conjuntas se denomina cinemática directa.

### Cinemática Inversa

El cálculo de la orientación que debe tomar cada articulación que compone el manipulador a partir de conocer la posición y orientación del extremo final del robot se denomina cinemática inversa. [19]

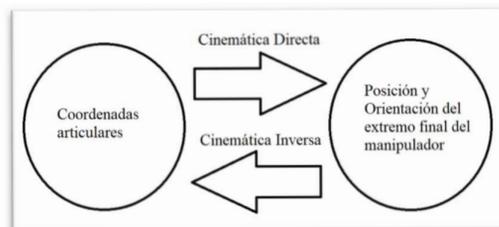


Figura Nº 2: Cinemática inversa y directa de un manipulador

## TARJETAS MICROCONTROLADORAS DE BAJO COSTO

El desarrollo de las nuevas tecnologías además de los constantes requerimientos de nuevas prestaciones, servicios, grandes limitantes dadas por los fabricantes de software y hardware y los altos precios para la automatización, ha hecho que nuevos

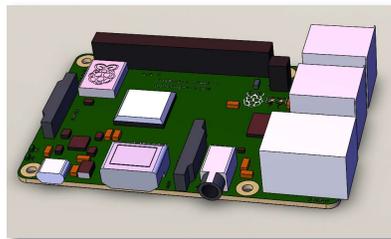
estándares sean desarrollados dando lugar a la aparición de nuevos hardware microcontroladores, como son las tarjetas empotradas de bajo costo, entre las cuales destacan Raspberry Pi, Odroid-c2, LattePanda que son las más usadas hoy en día.

### **Raspberry Pi**

Es un ordenador de placa única o reducida (SBC) de bajo costo, con el objetivo de desarrollar sistemas de enseñanza en informática, fue lanzado por la Fundación Raspberry Pi el modelo A del microcontrolador, expresamente no se define como hardware libre sin embargo al ser de código abierto (Open Source) cualquiera puede aportar haciendo modificaciones en la programación del software original, la tarjeta permite el uso de varios sistemas operativos GNU/Linux como Ubuntu, Raspbian e incluso Windows 10 [20]

En cualquier versión de la tarjeta se puede encontrar un procesador Broadcom, HDMI, memoria RAM, puertos USB, Ethernet, una GPU, 40 pines GPIO y un conector para cámara, ninguna edición posee memoria por lo cual se debe colocar una MicroSD de alta velocidad para no afectar el rendimiento de la tarjeta [21].

Se considera como una potente alternativa ante los ordenadores actuales pues al poseer arquitectura armhf es de bajo costo, además al tener software basado en Linux, sus aplicaciones son de tipo Open Source, la tarjeta puede servir como un microcontrolador donde a partir de repositorios el usuario podrá tratar con una amplia multitud de programas [22].

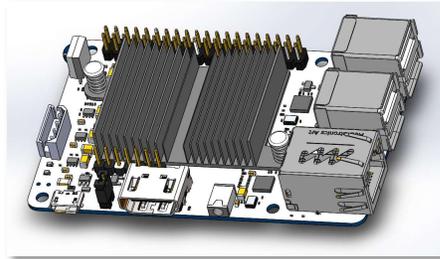


**Figura N° 3:** Tarjeta Raspberry Pi 3

### **Odroid-c2**

Es una placa desarrollada por la compañía coreana Hardkernel, tiene 2 GB de memoria RAM pero no cuenta con conectividad Wifi, aunque esto puede resolverse con un

adaptador, posee puerto de infrarrojos y que su salida HDMI permite una resolución de video de 4K a 60 Hz, algunos sistemas operativos como Ubuntu, Android y ARCH Linux se pueden ejecutar en el ODROID-C2 [23].



**Figura N° 4:** Tarjeta Odriod

### **Jaguar One**

Jaguar Electronic HK Co., Ltd lanzó oficialmente la primera computadora de placa única basada en X86 denominada Jaguarboard la cual se basó en el procesador Intel Atom de cuatro núcleos, este proporciona 1 GB de memoria RAM DDR3L y 16 GB de almacenamiento flash eMMC integrado, simplifica significativamente los kits de herramientas de desarrollo, además permite que aquellos que aún no están familiarizados con la arquitectura ARM, no necesiten invertir el tiempo y el esfuerzo en aprender algo nuevo antes de poner el Jaguarboard a trabajar en un proyecto. Admite ampliamente distribuciones de Linux convencionales como Ubuntu, Debian y Red Hat, así como versiones estándar de Windows 8.1 / 10 y Android [24].



**Figura N° 5:** Tarjeta Jaguar One

### **LattePanda**

Es una placa que integra los sistemas de Raspberry Pi y Arduino, surge esta nueva, desarrollada por la compañía China DFRobot, esta placa se puede ejecutar en Windows 10 integrando su versión completa, viene equipada con un procesador y un

coprocesador donde se integran las funciones de Arduino en esta misma placa, además ofrece dos alternativas según sus especificaciones (2/32 GB o 4/64 GB) también se incluye conectividad Wifi y Bluetooth [25].



**Figura N° 6:** Tarjeta LattePanda

**Tabla comparativa de tarjetas microcontroladoras**

A continuación, se presenta una tabla comparativa de las especificaciones técnicas más sobresalientes de cada una de las tarjetas microcontroladoras:

**Tabla 2:** Comparación entre tarjetas controladoras

	Raspberry Pi 3	Odroid C2	Jaguar One	LattlePanda
Procesador	1.4 GHz 64-bit Quad-Core ARMv8	1.5 GHz Quad-Core ARM Cortex A53	1.3 GHz Quad-Core Intel Atom Z3735G/F	1.8 GHz 64-bit Quad-Core Intel Z8300
Memoria de almacenamiento	Tarjeta microSD	Tarjeta microSD	16gb	16gb
Memoria RAM	1gb	2gb	1gb	2 y 4 gb
Conectividad	4 USB 2.0 HDMI 1.4 Wifi Bluetooth 4.1 Bluetooth (BLE) Ethernet	USB 2.0 1 USD OTG HDMI 2.0 Ethernet	3 USB 2.0 HDMI 1.4 Ethernet	1 USB 3.0 2 USB 2.0 Wifi Bluetooth 4.0 Ethernet
Sistema Operativo	Linux Windows	Linux Android	Linux Android Windows	Windows
Valor \$	60	65	80	160

## **SOFTWARES PARA DESARROLLO DE ENTORNOS VIRTUALES**

Desde hace años, los desarrolladores de juegos de computadora enfrentaron el mismo problema que enfrentan los investigadores de hoy: cómo simular movimientos reales de personajes y objetos en un ambiente divertido sin la necesidad de tener que programar directamente en compiladores rústicos en su lenguaje nativo

La solución fue desarrollar motores de juego que incluyen editores gráficos. Varios motores de juego, como Unreal Engine, CryEngine y Unity 3D están disponibles y debido a la alta demanda hoy por hoy los motores de juego se convirtieron en multiplataforma y son usados no solo para el desarrollo de videojuegos sino también han entrado con gran fuerza en el mundo de la investigación para desarrollar entornos virtuales e interfaz de usuario para el desarrollo de múltiples aplicaciones.

### **Unity**

Unity 3D creado por Unity Technologies, es un potente motor 3D multiplataforma y un entorno de desarrollo fácil de usar. Es un software para el desarrollo de aplicaciones virtuales principalmente para videojuegos, cuenta con herramientas para realizar aplicaciones desde las más básicas desde juegos en 2D hasta aplicaciones con una alta complejidad en diseño en 2 y 3D; Unity debería interesar a cualquiera que quiera crear fácilmente juegos y aplicaciones 3D para dispositivos móviles, computadoras de escritorio, la web y consolas [26].

### **Unreal Engine**

Unreal Engine es un motor de desarrollo de juegos. Fue desarrollado por Epic Games en el año 1988. Inicialmente, se desarrolló como un juego de disparos en primera persona. Actualmente se está utilizando para crear juegos de lucha, juegos de rol, sigilo y otros. Este motor es popular entre los desarrolladores de juegos y es utilizado por ellos como una herramienta para el desarrollo y creación de escenarios virtuales. Aunque la mayoría de los desarrolladores usan Unity para la creación de juegos, ofrece técnicas básicas de desarrollo que ayudan mucho a los principiantes [27].

### **CryEngine**

CryEngine es un motor de juego diseñado por el desarrollador de juegos alemán Crytek. Se ha utilizado en todos sus títulos con la versión inicial utilizada en Far Cry,

## **B. DESARROLLO**

### **CAPÍTULO I. MARCO TEÓRICO**

#### **1.1. Antecedentes Investigativos**

##### **1.1.1. Contextualización del Problema**

En la actualidad la evolución de los sistemas productivos son la base de la competitividad, mejoramiento y desarrollo económico y social de las organizaciones industriales a nivel mundial. La fusión de la automatización industrial y la informática han dado surgimiento a la innovación técnica conocida como Industria 4.0[1]. La industria 4.0 se caracteriza por una dependencia creciente sobre la automatización e interconexión de sistemas debido a la necesidad de aumentar la eficiencia, autonomía y personalización de procesos a un bajo costo, y así, los sistemas robóticos basados en sistemas 4.0 se convierten en una importante herramienta que tiene un incremento en la demanda año tras año dentro de la industria mundial [2].

Durante los últimos años, la robótica ha tenido como objetivo encontrar soluciones a las necesidades de todos los campos en los que se desenvuelve el humano. La evolución de los campos de aplicación y su sofisticación han influenciado los temas de investigación en la comunidad de robótica. Esta evolución ha estado dominada por las necesidades humanas principalmente en trabajos que son de alto riesgo para el hombre, de allí surge la idea de reemplazarlos con robots industriales [3].

En el contexto actual del mundo tecnológico, los sistemas robóticos han tenido un avance importante en cuanto a costo e implementación, con la aparición de las tarjetas SBC de bajo costo se puede desarrollar sistemas de control eficientes para automatismos robóticos a precios realmente bajos en comparación a los sistemas que se manejan en el mercado de la robótica y automatización. Sin embargo, y a pesar de ser un gran avance estas nuevas tecnologías han tenido lugar solo en las grandes potencias mundiales, existe un limitado conocimiento de las mismas en países menos industrializados.

Actualmente en América latina la utilización de sistemas robóticos y en particular manipuladores es una necesidad que va adquiriendo fuerza dentro de la industria, ya que, los mismos permiten realizar tareas con mayor fuerza, precisión y rapidez además se adaptan fácilmente a cualquier actividad siendo los más aptos para tareas peligrosas

en las cuales se pone en riesgo la integridad de las personas, todo esto hace que se deba tener un control preciso para su correcto funcionamiento, a pesar de ser una gran necesidad para el desarrollo de la industria, los altos precios en el mercado de la robótica hace difícil que la mayoría de empresas puedan implementar uno de estos sistemas de automatización [4].

Dentro de la industria ecuatoriana existen un limitado número de empresas que han integrado a sus instalaciones manipuladores y robots para realizar sus tareas diarias, esto debido a dos factores importantes, en primer lugar, el precio para adquirir sistemas robóticos es elevado y segundo aún existe un escaso conocimiento hacia los sistemas que controlan estos robots, ya que, los sistemas de control que actualmente se presentan son complejos y utilizan hardware que tienen precios elevados [5].

La ingeniería industrial al ser la encargada de formar profesionales capaces de buscar soluciones a los problemas que afronta la industria en la actualidad, debe incentivar a los estudiantes a proponer ideas nuevas que aporten al sector industrial, en base a este argumento y a la problemática descrita surge la necesidad de diseñar sistemas de control más simples que faciliten el entendimiento y manipulación de los mismos, utilizando componentes de bajo costo, que es en si lo que ofrece la industria 4.0.

### **1.1.2. Estudio del Arte**

En los últimos años la industria se ha planteado nuevos desafíos para realizar procesos de producción procurando que estos sean más eficientes, autónomos y personalizables, este hecho condujo a la aparición de un nuevo concepto llamado Industria 4.0 y gracias a esta la industria en general se está adaptando a los nuevos sistemas tecnológicos para la creación de las llamadas "Fábricas inteligentes" [6].

Actualmente se habla de un futuro donde un usuario controla un robot virtual desde su teléfono o tableta que involucra también un movimiento en un robot físico. Esto no solo en la industria sino también en la vida cotidiana de las personas por ejemplo un usuario puede verificar el contenido de su refrigerador en el mundo virtual, y automáticamente ordenar comestibles arrastrando y soltando los productos deseados en la nevera virtual, el efecto al final es llenar tanto la nevera virtual como la nevera física.

De igual forma un robot físico puede ser controlado directamente programando el movimiento al presionar teclas, con el antecedente que para controlar un robot virtual se lo hace de manera similar. Sin embargo, los robots físicos y virtuales pueden no reflejarse con precisión el uno al otro sin un usuario humano que ejecute las interacciones [7].

A continuación, se detalla investigaciones en las cuales se ha puesto en evidencia la importancia de utilizar sistemas en base a industria 4.0, para el control de articulaciones robóticas desde entornos virtuales utilizando tarjetas de bajo costo, así es lo que detalla la investigación de McLin en [7], donde se propone un enfoque para establecer interacciones bidireccionales entre robots en lo físico y mundos virtuales, permitiendo cambios que ocurren en un robot físico para ser reflejado automáticamente por su contraparte virtual, y viceversa, basada en esta investigación se puede establecer cuáles son los requisitos para crear un virtual que refleje el mundo físico.

Del mismo modo en la investigación desarrollada por Yoshiaki en [8], en donde explica acerca de una nueva arquitectura de software para un simulador de interacción humano-robot (HRI) en entornos virtuales, una plataforma de realidad virtual basada en la nube, llamada "SIGVerse", que reduce costos para desarrollar robots reales y experimentos de interacción en el mundo real con la ayuda de lo virtual. Esta información sustenta el entendimiento de cómo se generan las plataformas virtuales para generar movimientos en aparatos físicos logrando una sincronización entre ambos.

Investigaciones similares demuestran que se puede diseñar sistemas de control eficientes para manipuladores como es el caso del SCORBOT, así pues, lo expone Mota Muñoz en [9], el cual explica sobre la implementación de un controlador de punto de ajuste PD + G para la segunda y tercera articulación del robot SCORBOT ER-4u. Para lo cual se calculó los parámetros dinámicos del robot a través de mediciones de laboratorio y cálculos en un modelo CAD. El modelo dinámico fue obtenido por Euler método de Lagrange, en base a esto se puede realizar el estudio del comportamiento del robot y como establecer los parámetros que se debe considerar para crear un controlador del manipulador Scrobot.

En la investigación desarrollada por Michail G en [10], describe un método de control moderno para el robot SCORBOT ER-III en el artículo se menciona el vínculo entre la tecnología moderna, actualizada y equipos robóticos de edades más avanzadas, propone herramientas de programación más eficientes para la creación de interfases más fáciles de usar que las que se presenta en entornos más antiguos. Esta investigación da las pautas de como integrar los últimos avances en programación para el diseño de interfases intuitivas que permitan el control y manipulación del brazo Scorbot.

Según los recientes avances en tecnología de hardware describe que se puede crear nuevas interfases con Realidad Aumentada (AR) para el control y manipulación sistemas robóticos, así lo expone Courtney Hutton en [11] que se refiere a las nuevas tecnologías que están cada vez más presentes en actividades para robots orientados a tareas definas, en la investigación explica la forma de comenzar a desarrollar un sistema AR en Unity para controlar robots, bajo el sistema simple de "apuntar el lugar al cual desplazarse".

Por otra parte, también se puede utilizar la realidad aumentada para la visualización de datos así lo explica Florian Leutert en [12] quien expone sobre la Realidad Aumentada (AR), como una forma de Realidad Mixta que es una técnica que mejora el entorno natural del usuario con información virtual útil adicional para ayudarlo en su tarea; se detalla la forma correcta de cómo debe integrarse y orientarse dentro del entorno la información, la cual debe ser dinámica, interactiva y en tiempo real en el espacio de trabajo del usuario. Bajo esta consideración se puede diseñar entornos de realidad eficientes, en donde existe un entorno virtual con soporte de realidad aumentada para la interacción con el usuario.

Finalmente, Tom Williams en [13] expone sobre la interacción humano robot basado en sistemas de realidad aumentada, avances recientes en realidad mixta, aumentada y virtual han permitido sugerir un método alternativo para mediar las interacciones humano-robot, permitiendo la comunicación de estas intenciones aprovechando el espacio físico humano como un lienzo compartido para él envió señales visuales. La técnica se basa en la utilización la mano que se presenta como interactiva de dentro del entorno virtual creando así un Puntero (IHP) para controlar un robot en un espacio de trabajo físico real.

Con el sustento de las investigaciones mencionadas se hace factible el desarrollo del tema propuesto, ya que, apoyados en los temas desarrollados por los diferentes autores se puede desarrollar un método de control para el manipulador SCORBOT-ER 4U, desde un entorno virtual con soporte de realidad aumentada el cual sea interactivo e intuitivo y utilizando dispositivos de bajo costo como medios hardware para la comunicación física.

### **1.1.3. Fundamentación Teórica**

La implementación de un sistema de control para el manipulador SCORBOT-ER 4U, aborda la temática que se detalla a continuación:

## **INTRODUCCIÓN A LA ROBÓTICA**

La robótica tuvo sus inicios a comienzos del siglo XXI, fue en 1917 cuando Joseph Capek describió la palabra autómatas y en 1921 su hermano Karel Capek escribió la obra “Robots Universales de Rossum (RUR)” desde entonces el concepto de robótica entró a lenguaje mundial. El término robot se deriva de la palabra checa robota, que significa siervo o trabajador [14].

La robótica es la ciencia de la tecnología que estudia la construcción y el diseño de máquinas las cuales pueden realizar tareas con cierta lógica, siendo capaces de reemplazar la mano de obra del hombre. Para realizar estas tareas estas máquinas estas programadas con base en ciencias exactas y lógicas como la algebra, la mecánica, los autómatas programables entre otras.

En los últimos años la robótica ha ido creciendo en cuanto aplicaciones en los diferentes campos de la industrial como la medicina, la exploración planetaria y submarina, automatización de procesos industriales, seguridad, entretenimiento, entre otros. Uno de los aportes de mayor impacto se ha dado en la educación, donde los robots al integrarse al grupo de estudiantes y tutores, propician el aprendizaje y el fortalecimiento de habilidades cognitivas. [15]

### **Robot**

Un robot es un dispositivo electromecánico con múltiples grados de libertad, programable para realizar una variedad de tareas. En su gran mayoría estos robots adoptan la forma de un brazo humano con un artefacto o herramienta en su parte final,

algunos también tienen una herramienta final en forma de muñeca. Normalmente su trabajo es realizar llevar a cabo actividades repetitivas, realizar grandes cargas, soldadura punto a punto entre otras [16]

Los grados de libertad o movilidad son la cantidad de movimientos independientes que puede hacer un dispositivo. Los conceptos ligados a un robot se describen en la tabla No.1

**Tabla 1: Conceptos Ligados a los robots**

<b>Manipulador</b>	<b>Dispositivo electromecánico capaz de interactuar con su entorno.</b>
<b>Efector Final</b>	La herramienta, la pinza u otro dispositivo montado al final de un manipulador, para realizar tareas útiles.
<b>Espacio de Trabajo</b>	El espacio que puede alcanzar el efector final de un robot, tanto en posición como en orientación.
<b>Posición</b>	La ubicación traslacional (línea recta) de un objeto.
<b>Orientación</b>	La ubicación rotacional (angular) de un objeto.
<b>Articulación</b>	El dispositivo que permite el movimiento relativo entre dos enlaces en un robot.
<b>Cinemática</b>	El estudio del movimiento sin tener en cuenta las fuerzas que lo provocan.
<b>Dinámica</b>	El estudio del movimiento con respecto a las fuerzas que lo producen.
<b>Actuador</b>	Proporciona fuerza para el movimiento del robot.
<b>Sensor</b>	Lee variables reales en el movimiento del robot para usar en el control.

## Tipos de Robots

Existen un sin número de robots como son:

- Robots Cilíndricos
- Robots Transaccionales
- Robots Esféricos
- Robots Móviles
- Robots Humanoides
- Robots Insectos
- Robots Articulado [17].

De entre esta lista se enfatiza el estudio de los robots articulados ya que son los más utilizados en la industria.

**Robot articulado:** Los robots articulados se asemejan al brazo humano en su movimiento 3D (son antropomórficos). Ellos tienen tres articulaciones R, con tres ángulos variables teta 1, teta 2 y teta 3, que representan la cintura del cuerpo humano, articulaciones de hombro y codo. Son robots versátiles, pero tienen una cinemática y dinámica más compleja que los anteriores. Estos robots pueden usarse con una variedad de muñecas o herramienta final en el extremo de su último eslabón. Los robots articulados tienen la arquitectura mostrada en la figura No.1



**Figura N° 1:** Arquitectura robot articulado

### Cinemática de un Robot Articulado

La cinemática es el estudio analítico de la geometría de movimiento de un mecanismo con respecto a un sistema de coordenadas fijo, sin tener en cuenta las fuerzas o momentos que causan el movimiento. Para controlar y programar un robot debemos tener conocimiento de su disposición espacial y un medio de referencia al medio

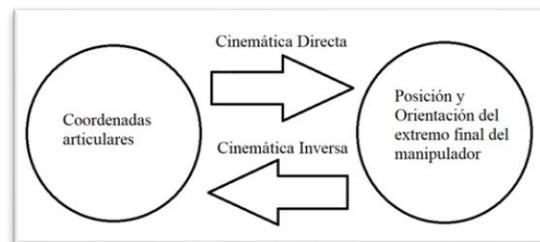
ambiente. La mecánica de un manipulador puede ser representado como una cadena cinemática de cuerpos rígidos conectados por articulaciones revolutas o prismáticas. Un extremo de la cadena está restringido a una base, mientras que un actuador final es montado en el otro extremo de la cadena [18]. La cinemática del robot se puede dividir en cinemática directa e inversa.

### **Cinemática Directa**

Un manipulador está compuesto por enlaces que se fijan entre sí mediante articulaciones ya sea revolutas o prismáticas desde la base hasta extremo final del robot. El cálculo de la posición y orientación del extremo final en términos de las variables conjuntas se denomina cinemática directa.

### **Cinemática Inversa**

El cálculo de la orientación que debe tomar cada articulación que compone el manipulador a partir de conocer la posición y orientación del extremo final del robot se denomina cinemática inversa. [19]



**Figura N° 2:** Cinemática inversa y directa de un manipulador

## **TARJETAS MICROCONTROLADORAS DE BAJO COSTO**

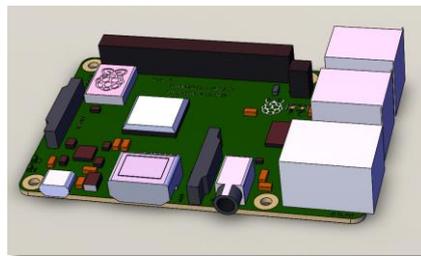
El desarrollo de las nuevas tecnologías además de los constantes requerimientos de nuevas prestaciones, servicios, grandes limitantes dadas por los fabricantes de software y hardware y los altos precios para la automatización, ha hecho que nuevos estándares sean desarrollados dando lugar a la aparición de nuevos hardware microcontroladores, como son las tarjetas empotradas de bajo costo, entre las cuales destacan Raspberry Pi, Odroid-c2, LattePanda que son las más usadas hoy en día.

## Raspberry Pi

Es un ordenador de placa única o reducida (SBC) de bajo costo, con el objetivo de desarrollar sistemas de enseñanza en informática, fue lanzado por la Fundación Raspberry Pi el modelo A del microcontrolador, expresamente no se define como hardware libre sin embargo al ser de código abierto (Open Source) cualquiera puede aportar haciendo modificaciones en la programación del software original, la tarjeta permite el uso de varios sistemas operativos GNU/Linux como Ubuntu, Raspbian e incluso Windows 10 [20]

En cualquier versión de la tarjeta se puede encontrar un procesador Broadcom, HDMI, memoria RAM, puertos USB, Ethernet, una GPU, 40 pines GPIO y un conector para cámara, ninguna edición posee memoria por lo cual se debe colocar una MicroSD de alta velocidad para no afectar el rendimiento de la tarjeta [21].

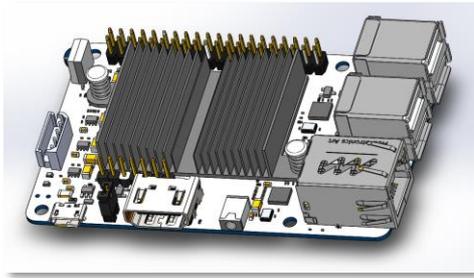
Se considera como una potente alternativa ante los ordenadores actuales pues al poseer arquitectura armhf es de bajo costo, además al tener software basado en Linux, sus aplicaciones son de tipo Open Source, la tarjeta puede servir como un microcontrolador donde a partir de repositorios el usuario podrá tratar con una amplia multitud de programas [22].



**Figura N° 3:** Tarjeta Raspberry Pi 3

## Odroid-c2

Es una placa desarrollada por la compañía coreana Hardkernel, tiene 2 GB de memoria RAM pero no cuenta con conectividad Wifi, aunque esto puede resolverse con un adaptador, posee puerto de infrarrojos y que su salida HDMI permite una resolución de video de 4K a 60 Hz, algunos sistemas operativos como Ubuntu, Android y ARCH Linux se pueden ejecutar en el ODROID-C2 [23].



**Figura N° 4:** Tarjeta Odriod

### **Jaguar One**

Jaguar Electronic HK Co., Ltd lanzó oficialmente la primera computadora de placa única basada en X86 denominada Jaguarboard la cual se basó en el procesador Intel Atom de cuatro núcleos, este proporciona 1 GB de memoria RAM DDR3L y 16 GB de almacenamiento flash eMMC integrado, simplifica significativamente los kits de herramientas de desarrollo, además permite que aquellos que aún no están familiarizados con la arquitectura ARM, no necesiten invertir el tiempo y el esfuerzo en aprender algo nuevo antes de poner el Jaguarboard a trabajar en un proyecto. Admite ampliamente distribuciones de Linux convencionales como Ubuntu, Debian y Red Hat, así como versiones estándar de Windows 8.1 / 10 y Android [24].



**Figura N° 5:** Tarjeta Jaguar One

### **LattePanda**

Es una placa que integra los sistemas de Raspberry Pi y Arduino, surge esta nueva, desarrollada por la compañía China DFRobot, esta placa se puede ejecutar en Windows 10 integrando su versión completa, viene equipada con un procesador y un coprocesador donde se integran las funciones de Arduino en esta misma placa, además ofrece dos alternativas según sus especificaciones (2/32 GB o 4/64 GB) también se incluye conectividad Wifi y Bluetooth [25].



**Figura N° 6:** Tarjeta LattePanda

### Tabla comparativa de tarjetas microcontroladoras

A continuación, se presenta una tabla comparativa de las especificaciones técnicas más sobresalientes de cada una de las tarjetas microcontroladoras:

**Tabla 2:** Comparación entre tarjetas controladoras

	Raspberry Pi 3	Odroid C2	Jaguar One	LattlePanda
Procesador	1.4 GHz 64-bit Quad-Core ARMv8	1.5 GHz Quad-Core ARM Cortex A53	1.3 GHz Quad-Core Intel Atom Z3735G/F	1.8 GHz 64-bit Quad-Core Intel Z8300
Memoria de almacenamiento	Tarjeta microSD	Tarjeta microSD	16gb	16gb
Memoria RAM	1gb	2gb	1gb	2 y 4 gb
Conectividad	4 USB 2.0 HDMI 1.4 Wifi Bluetooth 4.1 Bluetooth (BLE) Ethernet	USB 2.0 1 USD OTG HDMI 2.0 Ethernet	3 USB 2.0 HDMI 1.4 Ethernet	1 USB 3.0 2 USB 2.0 Wifi Bluetooth 4.0 Ethernet
Sistema Operativo	Linux Windows	Linux Android	Linux Android Windows	Windows
Valor \$	60	65	80	160

## **SOFTWARES PARA DESARROLLO DE ENTORNOS VIRTUALES**

Desde hace años, los desarrolladores de juegos de computadora enfrentaron el mismo problema que enfrentan los investigadores de hoy: cómo simular movimientos reales de personajes y objetos en un ambiente divertido sin la necesidad de tener que programar directamente en compiladores rústicos en su lenguaje nativo

La solución fue desarrollar motores de juego que incluyen editores gráficos. Varios motores de juego, como Unreal Engine, CryEngine y Unity 3D están disponibles y debido a la alta demanda hoy por hoy los motores de juego se convirtieron en multiplataforma y son usados no solo para el desarrollo de videojuegos sino también han entrado con gran fuerza en el mundo de la investigación para desarrollar entornos virtuales e interfaz de usuario para el desarrollo de múltiples aplicaciones.

### **Unity**

Unity 3D creado por Unity Technologies, es un potente motor 3D multiplataforma y un entorno de desarrollo fácil de usar. Es un software para el desarrollo de aplicaciones virtuales principalmente para videojuegos, cuenta con herramientas para realizar aplicaciones desde las más básicas desde juegos en 2D hasta aplicaciones con una alta complejidad en diseño en 2 y 3D; Unity debería interesar a cualquiera que quiera crear fácilmente juegos y aplicaciones 3D para dispositivos móviles, computadoras de escritorio, la web y consolas [26].

### **Unreal Engine**

Unreal Engine es un motor de desarrollo de juegos. Fue desarrollado por Epic Games en el año 1988. Inicialmente, se desarrolló como un juego de disparos en primera persona. Actualmente se está utilizando para crear juegos de lucha, juegos de rol, sigilo y otros. Este motor es popular entre los desarrolladores de juegos y es utilizado por ellos como una herramienta para el desarrollo y creación de escenarios virtuales. Aunque la mayoría de los desarrolladores usan Unity para la creación de juegos, ofrece técnicas básicas de desarrollo que ayudan mucho a los principiantes [27].

### **CryEngine**

CryEngine es un motor de juego diseñado por el desarrollador de juegos alemán Crytek. Se ha utilizado en todos sus títulos con la versión inicial utilizada en Far Cry,

y continúa actualizándose para admitir nuevas consolas y hardware para sus juegos. También se ha utilizado para muchos juegos de terceros bajo el esquema de licencias de Crytek, incluidos Sniper: Ghost Warrior 2 y SNOW [28].

### Tabla comparativa de los softwares para desarrollo de entornos virtuales

En la siguiente tabla se presenta las características más sobresalientes de los motores de juegos descritos anteriormente

**Tabla 3:** Comparación entre softwares de desarrollo virtual

	Unity 3D	Unreal Engine	CryEngine
Sistema Operativo soportado	Windows Mac OS Linux	Windows Mac OS Linux	Windows Mac OS
Memoria RAM requerida	4gb	4gb	4gb
Memoria de video requerida	512mb	1gb	1gb
Documentación	Cuenta con gran cantidad de documentación en su página oficial, además de un sin número de tutoriales.	Gran cantidad de documentos disponibles y manuales de uso	Existe mucha información como tutoriales disponibles en la red.
Calidad Gráfica	Media – alta	Alta	Media
Herramientas y Librerías	Ofrece una gran cantidad de complementos, modelos prefabricados y librerías para adaptar todo tipo de	Ofrece complementos y librerías para generar aplicaciones de todo tipo.	Sus herramientas son limitadas y requiere en su mayoría de librerías exclusivas de la marca.

	dispositivos externos.		
Lenguaje del Código Base	C SHARP (C#), simple de aprender	C++, complejo de aprender	C++, complejo de aprender
Precio \$	0	0	0

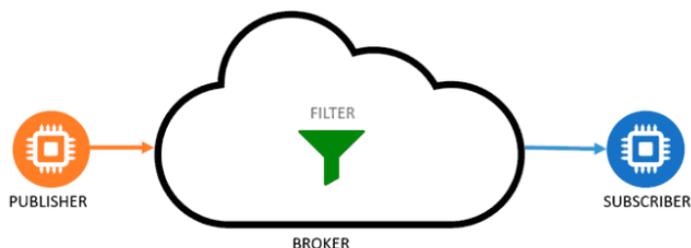
## PROTOCOLOS IOT DE COMUNICACIÓN

Internet de las cosas (en inglés, Internet of Things, abreviado: IoT) es un concepto que se refiere a la interconexión digital de cosas u objetos en Internet. Con la proliferación de la comunicación de máquina a máquina, existen muchos protocolos de comunicación estandarizados para aplicaciones de IoT. El rendimiento de estos protocolos puede diferir significativamente entre sí, incluso en las mismas condiciones de funcionamiento. Existen varios protocolos de comunicación IoT conocidos, a saber, CoAP (Protocolo de aplicación restringida), MQTT (Transporte de telemetría de mensajes en cola) y AMQP (Protocolo avanzado de colas de mensajes) [29].

### MQTT

MQTT es un protocolo de capa de aplicación que funciona por encima del protocolo de capa de transporte TCP / IP para transferir mensajes. Es un protocolo adecuado para dispositivos con recursos limitados y fue inventado por el Dr. Andy Stanford-Clark y Arlen Nipper en 1999. Es un protocolo ligero, abierto, simple y fácil de usar, lo que lo hace ideal para la comunicación en entornos con recursos limitados como IoT [30].

El protocolo funciona bajo la modalidad publicador – suscriptor a través de la creación de un bróker, para filtrar los mensajes, estos se disponen en tópicos organizados jerárquicamente. En la figura 7 se ilustra el método de trabajo de mqtt.



**Figura N° 7:** Estructura MQTT

Los parámetros de comunicación son los siguientes:

- Dirección del Broker
- Definir el nombre del tópico
- Definir el puerto de comunicación

## CoAP

Constrained Application Protocol (CoAP) es un protocolo de transferencia web especializado para usarse con nodos y redes. Está diseñado principalmente para el transporte de datos pequeños entre nodos con recursos limitados. Este protocolo trabaja como cliente – servidor sin embargo aun no está estandarizado al 100%, su forma de operar es mediante un nodo cliente en el cual puede comandar a otro enviando un paquete de datos, el servidor lo interpretará, extraerá la información y decidirá qué acción ejecutar. El servidor no necesariamente realizará la conformación de la petición [31].

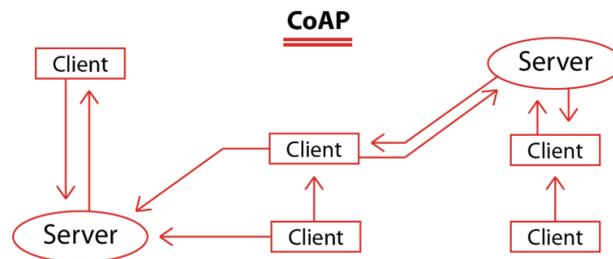
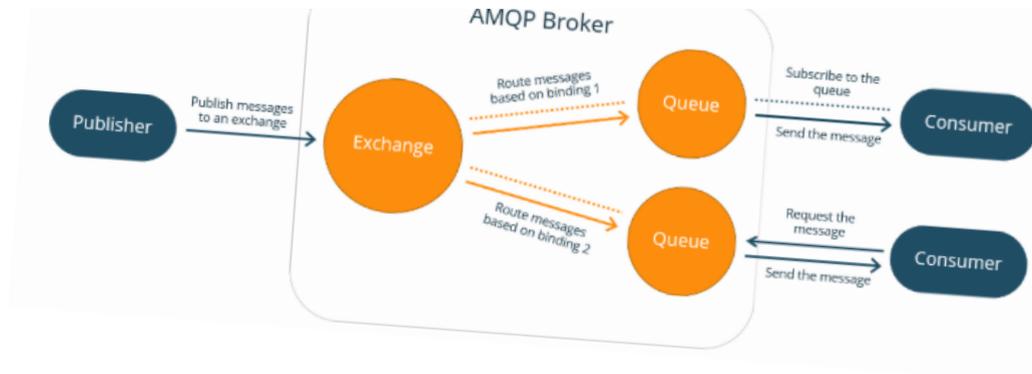


Figura N° 8: Protocolo de Comunicación COAP

## AMQP

El Advanced Message Queuing Protocol (AMQP) es un protocolo de capa de aplicación estándar abierto para middleware orientado a mensajes. Sus características principales son orientación de mensajes, colas, enrutamiento, seguridad, confiabilidad e interoperabilidad. Estas características lo hacen perfecto para pasar mensajes comerciales entre organizaciones y aplicaciones. AMQP también se puede usar como un protocolo de Internet de las cosas (IoT) [32].



**Figura N° 9:** Estructura AMQP

### Tabla comparativa de Protocolos de Comunicación

En la siguiente tabla se presenta las características más sobresalientes de los protocolos de comunicación IOT.

**Tabla 4:** Comparación entre protocolos de comunicación

	MQTT	COAP	AMQP
Protocolo Base	TCP/IP	UDP/IP	TCP/IP
Modelo usado para comunicación	Publicación / Suscripción	Petición / Respuesta	Intercambio de Mensajes punto a punto
Fiabilidad	<ul style="list-style-type: none"> <li>• QoS 0: Entrega no garantizada</li> <li>• QoS 1: Confirmación de entrega</li> <li>• QoS 2: Doble confirmación de entrega</li> </ul>	<ul style="list-style-type: none"> <li>• Mensajes no confirmados</li> <li>• Mensajes no confirmados</li> <li>• ACKs</li> <li>• Retransmisiones</li> </ul>	No presenta confirmación de entrega de datos.
Consumo de Energía	Alto	Medio	Bajo
Seguridad	TLS	DTLS	TLS

Tolerancia a fallos	En el nodo central Broker	Descentralizado	En la implementación
Implementación	Fácil de implementar, compleja para añadir extensiones	Pocas librerías existentes y soporte casi nulo	Fácil de Implementar

### REALIDAD AUMENTADA

Realidad Aumentada es una tecnología que permite añadir capas de información visual sobre el mundo real que nos rodea, utilizando dispositivos como pueden ser teléfonos móviles, gafas tecnológicas, tabletas, computadoras, cámara, entre otros. Esto nos ayuda a generar experiencias que aportan un conocimiento relevante sobre nuestro entorno, y además recibimos esa información en tiempo real. Entre sus principales características figuran la facilidad para interactuar en el mundo real con elementos del mundo virtual, mezclando así lo mejor de ambos y la inclusión de información en tiempo real.

La realidad aumentada ha traído consigo un gran aporte a diversos campos como son: la medicina, educación, entretenimiento y en los últimos años se han desarrollado diversas aplicaciones para el campo industrial, aporta grandes ventajas a los diversos campos entre las cuales destacan la optimización de tiempos en tareas diarias de muchos trabajadores, implementar nuevos canales para mostrar información a las personas y facilidad para entrenamiento de personal[33].

#### Tabla comparativa gafas inteligentes para realidad aumentada

En la siguiente tabla se presenta las características de 3 tipo de gafas inteligentes utilizadas como dispositivos para realidad aumentada

**Tabla 5:**Tabla Comparativa Gafas Inteligentes

Gafas Inteligentes			
Propiedad	Meta II	Oculus Rift S	Hololens II

<b>Entornos que Ofrece</b>	Realidad Aumentada	Realidad Virtual	Realidad Aumentada
<b>Resolución</b>	2560 x 1440	2160 x 1200	1268 x 720
<b>Características que ofrece</b>	<ul style="list-style-type: none"> <li>• Interacción con entornos virtuales</li> <li>• Imágenes en 3D con alta resolución</li> <li>• Localización y mapeo en tiempo real</li> <li>• Sensores visuales</li> <li>• Micrófono</li> <li>• Diseño ergonómico</li> </ul>	<ul style="list-style-type: none"> <li>• Controlador Táctil</li> <li>• Sensores de movimiento</li> </ul>	<ul style="list-style-type: none"> <li>• Conectividad Inalámbrica WIFI y Bluethoot</li> <li>• Control de movimiento por voz</li> <li>• Batería recargable</li> </ul>
<b>Tipo de Conexión</b>	HDMI, USB 3.0	HDMI USB 2.0 y 3.0	Inalámbrica
<b>Campo de Visión</b>	90°	110°	90°
<b>Requisitos CPU</b>	<ul style="list-style-type: none"> <li>• Tarjeta gráfica GPU NVIDIA GeForce GTX 1050, AMD Radeon R 480</li> <li>• CPU: Intel Core i7-6700 (HQ)</li> <li>• 16 GB de RAM</li> </ul>	<ul style="list-style-type: none"> <li>• Tarjeta gráfica NVIDIA GTX 1060,</li> <li>• 8GB de memoria RAM,</li> <li>• Puertos USB 3.0 [35]</li> </ul>	<ul style="list-style-type: none"> <li>• CPU con 4 procesadores como mínimo</li> <li>• 8 GB de memoria RAM [36].</li> </ul>

	<ul style="list-style-type: none"> <li>• 10 GB de disco duro [34].</li> </ul>		
<b>Costo</b>	1299 dólares	599 dólares	3000 dólares

## 1.2. Objetivos

### Objetivo General

Desarrollar un método de control desde un entorno virtual para el brazo robótico manipulador SCORBOT-ER 4U utilizando tarjetas empotradas de bajo costo.

El objetivo principal de esta investigación es el desarrollo de un entorno virtual para el control y manipulación de un manipulador robótico como es el Scrobot ER 4U, utilizando microcontroladores de bajo costo o las llamadas tarjetas empotradas de bajo costo, para ello en primera instancia se va a identificar los parámetros Cinemáticos del brazo robótico y así establecer el método de control del manipulador, que será diseñado en un entorno virtual mediante el uso del motor de diseño Unity, el mismo que integrara gafas de realidad aumentada para la interacción con el usuario, una vez diseñado el método de control y la interfaz se podrá enviar mensajes a la tarjeta empotrada mediante un protocolo de comunicación, y a su vez la tarjeta controladora se encargara de mover al brazo robótico, finalmente estos mensajes serán validados comprobando que el movimiento del robot es el deseado.

### Objetivos Específicos

- Identificar los parámetros Cinemáticos del brazo robótico para su control mediante mensajes enviados hacia la tarjeta empotrada.
- Diseñar el entorno virtual para el control del manipulador SCORBOT-ER 4U utilizando el software Unity 3D y las gafas de realidad aumentada Meta-Development II
- Validar los parámetros establecidos de comunicación y control entre el entorno virtual y el manipulador Scrobot ER 4U para la manipulación real del mismo.

Las actividades para el cumplimiento de los objetivos se detallan a continuación

- Identificar los parámetros cinemáticos del brazo manipulador SCORBOT-ER 4U.

- Recolectar información acerca del Motor para desarrollo de entornos virtuales Unity 3d basado en realidad virtual.
- Establecer un método para el control del manipulador
- Diseñar el entorno virtual para el control del manipulador SCORBOT-ER 4U utilizando el software Unity 3D y las gafas de realidad aumentada Meta-Development II
- Definir la tarjeta empotrada de bajo costo a utilizar
- Definir el protocolo de comunicación entre el entorno virtual y la tarjeta empotrada
- Realizar pruebas de comunicación entre el entorno virtual y la tarjeta empotrada.
- Realizar pruebas de conexión entre la tarjeta empotrada y el manipulador.
- Analizar los errores encontrados y buscar soluciones para eliminar los mismos.
- Realizar pruebas de funcionamiento finales integrando todas las etapas enviando mensajes desde la interfaz de usuario hasta el robot manipulador verificando que el movimiento del mismo es el deseado.

## **CAPÍTULO II. METODOLOGÍA**

### **2.1. Materiales**

#### **SCORBOT ER 4U**

El robot SCORBOT-ER 4u es un robot industrial, una máquina programable de propósito general que posee ciertas características antropomórficas. Este brazo junto con la capacidad del robot para ser programado lo hace ideal para una variedad de tareas de producción. El robot SCORBOT-ER 4u es un sistema versátil y confiable para uso educativo. El brazo del robot se puede montar sobre una mesa, un pedestal o una base deslizante lineal. La velocidad y la repetibilidad del robot lo hacen muy adecuado tanto para operaciones independientes como para uso integrado en aplicaciones de celdas de trabajo automatizadas, como soldadura robótica, visión artificial, tendido de máquinas CNC y otras operaciones [37].

Las especificaciones se detallan en la siguiente tabla:

**Tabla 6:** Especificaciones técnicas Scorbot ER 4U

<b>SCORBOT-ER 4U ESPECIFICACIONES</b>	
<b>Estructura mecánica</b>	Articulado Vertical
<b>Número de ejes</b>	5 ejes
<b>Eje Movimientos:</b>	
<b>Eje 1: Giro básico</b>	310
<b>Eje 2: Rotación del hombro</b>	158
<b>Eje 3: Rotación del codo</b>	260
<b>Eje 4: Paso de la muñeca</b>	260
<b>Eje 5: Rollo de muñeca</b>	(mecánicamente) + - 570 (eléctricamente)
<b>Operación del radio máximo</b>	610 mm
<b>Efactor final</b>	Pinza servo DC, con codificador óptico movimiento de los dedos en paralelo; Medición del tamaño de objeto por medio de sensor de sujeción y componentes de software
<b>La apertura máxima de pinza</b>	75 mm (3”) sin almohadilla de goma 65 mm (2,6”) con almohadillas de goma
<b>Buscador de Inicio de carrera</b>	Posición fija en cada eje, Encontrado por medio de micro interruptores
<b>Realimentación</b>	codificador óptico en cada eje
<b>Actuadores</b>	motor 12 VDC servo
<b>Capacidad del motor (axis1-6)</b>	15oz.en par máximo 70 W de potencia para el pico de par
<b>Transmisión</b>	Engranaje, correa de distribución, tornillo de avance
<b>La carga máxima</b>	1 kg (2.21b) incluyendo pinza
<b>Peso</b>	10.8kg (23,6 1b)
<b>La velocidad máxima camino</b>	600 mm / seg (23.6”)

<b>Temperatura ambiente de funcionamiento</b>	2-40 c (36-104f)
---	------------------



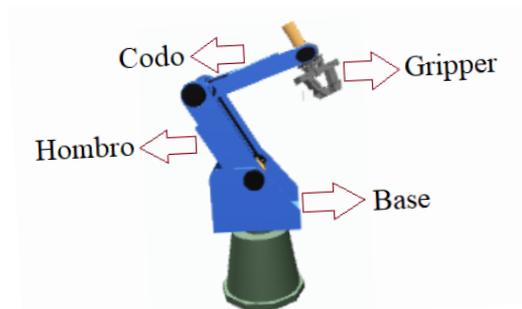
**Figura N° 10:** Manipulador SCORBOT-ER 4u

El SCORBOT-ER4 tiene un diseño y desarrollo para emular un robot industrial. La estructura abierta del brazo robótico permite a los estudiantes observar y aprender sobre su interior y los mecanismos que lo componen. El SCORBOT-ER 4 es un robot articulado vertical, con cinco articulaciones giratorias y una pinza adjunta, el robot tiene cinco grados de libertad. Este diseño permite al elemento final ser posicionado y orientado arbitrariamente dentro de un gran espacio de trabajo. Los movimientos de las articulaciones se describen en la siguiente tabla:

**Tabla 7:** Estructura Brazo manipulador Scorbtor ER 4U

Eje No.	Nombre de la junta	Movimiento	Motor No.
1	Base	Rotación del cuerpo	1
2	Hombro	Sube y baja la parte superior del brazo.	2
3	Codo	Sube y baja el antebrazo.	3
4	Elevación de Muñeca	Eleva y baja el efector final (pinza).	4+5

5	Giro de Muñeca	Gira el efector final (pinza).	4+5
---	----------------	--------------------------------	-----

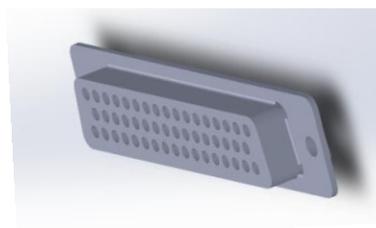


**Figura N° 11:** Estructura Manipulador SCORBOT-ER 4u

El cable principal del robot contiene 50 cables divididos en seis grupos (uno para cada motor). Cada grupo contiene ocho pistas:

- 2 cables de tensión de alimentación al motor.
- 2 cables reciben pulsos del codificador óptico (canal 0 y canal 1).
- 1 cable transporta la señal del micro interruptor.
- 1 cable suministra voltaje al codificador (VLED).
- 1 cable proporciona el terreno para el micro interruptor.
- 1 cable proporciona el terreno para el codificador.

Todos los comandos, tanto operativos como de control, se transmiten a través de este cable.



**Figura N° 12:** Conector DB50 de Scrobot ER 4U

### **HARDWARE CONTROLADOR DE BAJO COSTO**

En base a la tabla No. 2 de entre las tarjetas mencionadas se va hacer la selección de la tarjeta SCB para el control del Brazo Scrobot. La selección se hace en base a 3 criterios específicos:

- Velocidad de Procesamiento (Bueno = 3; Medio = 2, Malo = 1)
- Costo (Alto = 1; Moderado = 2; bajo = 3)
- Periféricos de Comunicación (Bueno = 3; Medio = 2, Malo = 1)

**Tabla 8:** Valoración Entre Tarjetas SBC

<b>Criterios</b>	<b>Raspberry PI</b>	<b>Odriod C2</b>	<b>Jaguar One</b>	<b>LattePanda</b>
<b>Velocidad de Procesamiento</b>	2	2	2	3
<b>Costo</b>	3	3	1	1
<b>Periféricos de Comunicación</b>	3	1	2	3
<b>Total</b>	8	5	5	7

Debido a su precio moderado en relación a los demás dispositivos, además de los diversos periféricos de comunicación con los que cuenta, y la velocidad de procesamiento, hace que Raspberry Pi sea una tarjeta apta para desarrollar la aplicación planteada.

### **TARJETA SBC RASPBERRY PI 3B+**

#### **Principales Características**

De entre sus principales características podemos citar la velocidad de su procesador que es de 1.4 GHz lo cual hace que su capacidad de procesamiento de datos sea muy alta, indispensable para controlar un proceso o sistema en tiempo real. Al poseer puertos de comunicación físicos e inalámbricos USB, Ethernet, Bluetooth y wifi permiten la posibilidad de desarrollar un sin número de aplicaciones bajo la mayoría de protocolos de comunicación (MQTT, TCP/IP, AMQP, Serial, entre otras). Trabaja con sistemas operativos como Windows y sistemas derivados de Linux (Raspbian), este último al ser un software libre, permite el desarrollo de código abierto y multiplataforma sin ningún tipo de restricción [22].

## **Raspbian**

Raspbian es un sistema operativo gratuito basado en Debian optimizado para el hardware Raspberry Pi. Raspbian proporciona más que un sistema operativo puro: viene con más de 35,000 paquetes, software pre compilado incluido en un formato agradable para una fácil instalación en su Raspberry Pi. La compilación inicial de más de 35,000 paquetes Raspbian, optimizada para el mejor rendimiento en Raspberry Pi, se completó en junio de 2012. Sin embargo, Raspbian todavía está en desarrollo activo con énfasis en mejorar la estabilidad y el rendimiento de tantos paquetes Debian como sea posible. Raspbian no está afiliado a la Fundación Raspberry Pi fue creado por un pequeño y dedicado equipo de desarrolladores que son fanáticos del hardware Raspberry Pi, los objetivos educativos de la Fundación Raspberry Pi y, por supuesto, el Proyecto Debian [38] .

## **Funciones**

La tarjeta raspberry pi satisface las necesidades para la creación del sistema de control del manipulador Scorbot, posee pines digitales de entrada / salida (GPIOs), necesarios para accionar los servomotores y leer los dispositivos de retroalimentación (Encoders) del brazo Scorbot, y así dar movimiento a las extremidades del robot, pues en si Raspberry Pi será el controlador del brazo robótico en el proyecto planteado.

Otro punto importante a considerar es que raspberry pi en conjunto con su sistema operativo nativo raspbian, permite la integración de librerías necesarias para generar protocolos de comunicación bajo IOT (Internet de las cosas), factor necesario para el desarrollo de la investigación, pues la manera de operar de la tarjeta SCB será controlar cada servomotor del brazo Scorbot en función del mensaje que se envíe desde un entorno virtual.

## **SOFTWARE PARA DESARROLLO DE ENTORNO VIRTUAL**

En base a la tabla No.3 se hace la selección del motor para desarrollar el entorno virtual. La selección se hace en base a 4 criterios específicos:

- Requerimientos de CPU (Altos = 1; Medios = 2; Bajos=3)
- Lenguaje de codificación (Complejo = 1; Moderado =2; Fácil=3)
- Documentación y Herramientas (Limitada =1; Moderada=2; Amplia=3)

- Calidad grafica (Baja = 1; Media=2; Alta=3)

**Tabla 9:** Valoración software de desarrollo virtual

Criterios	Unity 3D	Unreal Engine	CryEngine
Requerimientos de CPU	3	3	3
Documentación y Herramientas	3	3	1
Lenguaje de codificación	3	1	1
Calidad grafica	2	3	2
<b>Total</b>	<b>11</b>	10	7

Por su consumo bajo de recursos CPU además de la gran cantidad de documentación y herramientas que ofrece lo hacen un software apto para el desarrollo de aplicaciones y entornos virtual es ideal para desarrollar la aplicación que se ha puesto en este trabajo de investigación.

## **UNITY 3D**

### **Características Principales**

Es un motor de aplicaciones multiplataforma, ofrece soporte para programación en C # y la integración de Visual Studio. Unity también ofrece JavaScript como lenguaje de ensamble y MonoDevelop como IDE para aquellos que desean una alternativa a Visual Studio. Viene con potentes herramientas de animación que simplifican la creación de escenas en 3D o crear animaciones en 2D desde cero. Casi cualquier cosa puede ser animada, con la característica de ofrecer soporte para la casi la totalidad de dispositivos como, controles, volantes, gafas de realidad aumentada y virtual, dispositivos de sonido entre otros.

Con las herramientas de edición de Unity, puede manejar simultáneamente entradas para mouse, teclados y controladores de juegos. También hay un soporte bastante sólido para soluciones basadas en la nube para juegos multijugador con alojamiento

de servidor y emparejamiento escalable, lo que lo convierte en una solución todo en uno para experiencias multiplicadoras. Además, posee una tienda de activos que contiene miles de modelos, guiones, escenas, materiales y todo lo que pueda desear. Incluso puede vender sus propios activos en la tienda Unity [39].

Para juegos en 3D, Unity permite la especificación de compresión de texturas, mapas MIP y configuraciones de resolución para cada plataforma que admite el motor del juego, y proporciona soporte para mapeo de relieve, mapeo de reflexión, mapeo de paralaje, oclusión ambiental del espacio de pantalla (SSAO), dinámico sombras usando mapas de sombras, renderizado a textura y efectos de procesamiento posterior de pantalla completa. A partir de 2018, Unity se ha utilizado para crear aproximadamente la mitad de los nuevos juegos móviles en el mercado y el 60 por ciento de la realidad aumentada y el contenido de realidad virtual [40].

### **IDE para desarrollo de código**

Unity 3D cuenta con 2 softwares auxiliares para el desarrollo del código los cuales son Mono DEVELOP que viene instalado junto con el software Unity y Visual Studio que es un software externo. Ambos programas ofrecen codificación en C#.

La selección del programa ensamblador de código se la hace básicamente en base al sistema operativo (SO) sobre el cual se va a trabajar, visual studio es la mejor opción para desarrollo de código pues este software al ser nativo de la compañía Microsoft, ofrece un gran rendimiento en cualquier versión Windows.

### **Visual studio**

Visual Studio es un entorno de desarrollo integrado (IDE) desarrollado por Microsoft para desarrollar GUI (interfaz gráfica de usuario), consola, aplicaciones web, aplicaciones web, aplicaciones móviles, nube y servicios web. Con la ayuda de este IDE, puede crear código administrado, así como código nativo, no es un IDE específico del idioma, ya que puede usarlo para escribir código en C #, C ++, VB (Visual Basic), Python, JavaScript y muchos más idiomas. Proporciona soporte para 36 lenguajes de programación diferentes [41].

## HARDWARE DE REALIDAD AUMENTADA

Para poder realizar el control y manipulación del brazo Scrobot se ha planteado el uso de Realidad Aumentada, lo cual hace imprescindible la selección del hardware que será utilizado como dispositivo para la interacción del mundo virtual con el usuario.

La selección del hardware de realidad aumentada se la hace en base a 2 puntos específicos:

- Facilidad de uso
- Integración Unity

Para los 2 aspectos a evaluar se da una valoración de:

- Alto = 3
- Medio = 2
- Bajo = 1

	Meta II	Oculus Rift S	Holelens II
Facilidad de uso	3	3	3
Integración a Unity	3	1	2
Total	6	4	5

En base a la facilidad de uso que presenta y la fácil integración a Unity se ha seleccionado las gafas de realidad aumentada Meta II

### META 2 DEVELOPMENT KIT

Meta 2 ofrece numerosas capacidades para AR (realidad aumentada), incluido el campo de visión más amplio de cualquier producto de realidad aumentada en el mercado, el acceso más intuitivo a la información digital y soporte para la interacción directa de la mano con contenido digital y hologramas.



**Figura N° 13:** Kit Meta Development 2

Entre las características técnicas de Meta 2 Development están que incluye un auricular AR transparente que muestra contenido digital u hologramas con un campo de visión de 90 grados y una pantalla de alta resolución de 2550x1440. Diseñados para la comodidad, los auriculares permiten que el usuario vea sin obstrucciones todo lo que se encuentra debajo de las cejas para que puedan hacer contacto visual fácilmente con otras personas o usar lentes cómodamente.

El kit también incluye una cámara frontal de 720p, un sistema de audio de cuatro altavoces cerca del oído, control de brillo y volumen y una matriz de sensores para las interacciones con las manos y el seguimiento posicional. También hay un cable de 9 pies para video (HDMI 1.4b), datos y alimentación[42].

### **PROTOCOLO DE COMUNICACIÓN ENTRE ENTORNO VIRTUAL Y RASPBERRY PI**

En base a la tabla No.9 se hace la selección del protocolo de comunicación entre el entorno virtual y la tarjeta SCB. Para la selección se ha considerado 3 aspectos fundamentales:

- Facilidad de Implementación (Fácil =3; Moderado =2; Difícil = 1)
- Seguridad (Buena=3; Media=2; Mala=1)
- Consumo de Energía (Alto =1; Medio =2; Bajo=3)

**Tabla 10:** Valoración protocolos de comunicación

	MQTT	COAP	AMQP
Facilidad de Implementación	3	1	3

<b>Seguridad</b>	3	3	1
<b>Consumo de energía</b>	2	3	3
<b>Total</b>	8	7	7

El protocolo seleccionado es MQTT debido a su ligereza ya que se puede ejecutar en dispositivos de baja potencia lo que lo hace perfectamente adaptable a desarrollar aplicaciones para tabletas, telefonos móviles y laptops. Además, este protocolo de comunicación dispone de características importantes de seguridad y básicamente la calidad de servicio es excelente, no hay pérdida de datos ya que dispone del servicio QoS 2 assured que garantiza que cada mensaje se entregue al suscriptor.

### **SOFTWARE PARA MODELO 3D**

El desarrollo de la investigación se debe modelar el brazo robótico Scorbot ER 4U en un software de diseño asistido por computadora. Para ello se hace la selección de un software en base a los siguientes parámetros:

- Requerimientos de CPU
- Interfaz de usuario
- Eficiencia de diseño en 3D
- Ayuda y soporte
- Facilidad para exportar a Unity

Los softwares considerados son AutoCAD, SolidWorks y Blender

**Tabla 11: Escala de Valoración**

Bueno	3
Regular	2
Malo	1

En base a estos parámetros se escoge el programa apto para el modelamiento del robot Scorbot ER 4U, los programas a considerar son, SolidWorks, AutoCAD y Blender.

**Tabla 12: Matriz de selección programa CAD**

Criterios	Puntuación		
	AutoCAD	SolidWorks	Blender
<b>Requerimientos de CPU</b>	2	1	3
<b>Interfaz de usuario</b>	3	3	3
<b>Eficiencia de diseño en 3D</b>	2	3	3
<b>Ayuda y soporte</b>	3	3	2
<b>Facilidad para exportar a Unity</b>	1	1	3
<b>Total</b>	11	11	14

De entre los softwares CAD se escoge Blender debido a su facilidad para exportar a Unity además de los ventajas que ofrece, como es la ayuda para el diseño su interfaz es amigable e intuitiva, además que los recursos CPU son bajos en comparación con los otros softwares considerados. Con el software listo se procede a crear cada componente del manipulador SCORBOT ER 4U.

## **BLENDER**

Blender es un programa informático multiplataforma, es la suite de creación 3D gratuita y de código abierto. Admite la totalidad de creaciones en 3D: modelado, aparejo, animación, simulación, renderizado, composición y seguimiento de movimiento, edición de video y canalización de animación 2D [43].

## **2.2. Métodos**

### **Metodología**

La presente investigación tendrá un enfoque cuali-cuantitativo, cualitativo porque busca los beneficios de implementar nuevos sistemas de control y comunicación bajo industria 4.0 para la interacción entre sistemas físicos y virtuales enfocados a la manipulación de robots manipuladores, así también cuantitativo porque se obtendrá

valores numéricos que demuestren las mejoras que se puede tener al implementar estos sistemas.

### **2.2.1. Modalidad de Investigación**

#### **Investigación bibliográfica – documental**

Se realizará una investigación bibliográfica-documental en libros, tesis de tercer nivel, artículos científicos entre otros a manera de obtener información relevante acerca del tema de estudio, antecedentes investigativos, así como trabajos ya realizados con anterioridad que servirán de sustento científico del proyecto, se analizará resultados encontrados por otros autores, así como posibles soluciones para la problemática.

#### **Investigación de campo**

Debido a que la presente investigación se llevará a cabo en el manipulador SCORBOT-ER 4u que se encuentra en los laboratorios de la Facultad de Ingeniería En Sistemas Electrónica e Industrial de la Universidad Técnica de Ambato.

#### **Modalidad Exploratoria**

Se aplicará esta modalidad debido a que es necesario conocer, analizar, verificar y explorar las causas que generan problemas en los protocolos de comunicación entre medios físicos y virtuales para la manipulación de sistemas robóticos y de esta manera optar por el mejor protocolo de comunicación.

### **2.2.2. Recolección de Información**

#### **Métodos**

- **Documentos:** La información se la obtendrá utilizando el método de búsqueda en documentos y revistas científicas, lo cual garantiza información veraz y fiable.

#### **Técnicas**

- **Registros de pruebas:** Se tomará datos acerca de las pruebas de funcionamiento, determinando tiempos de retardo en la comunicación además de los principales problemas tanto en el entorno físico como virtual y de las posibles fallas en la interacción entre ellos como puede ser pérdida de datos.

### **2.2.3. Procesamiento y Análisis de Datos**

Toda la información recolectada de las pruebas de funcionamiento en cada etapa del sistema de control será registrada en el software de escritura Word para un análisis a fondo de las posibles fallas y problemas del sistema implementado con el afán de eliminar tiempos de retardos y pérdidas de datos en la comunicación entre la simulación y entorno físico.

### **2.2.4. Desarrollo del Proyecto**

Para el control y manipulación del robot SCORBOT-ER 4U se lo realizará mediante mensajes enviados desde el software de simulación Unity 3D hacia la tarjeta Raspberry Pi 3B+ utilizando un protocolo de comunicación basado en IOT (Internet de las cosas), la tarjeta Raspberry estará conectada directamente al robot SCORBOT y de esta manera podrá accionar cada una de sus articulaciones dando movimiento al robot.

A continuación, se describen las actividades para el desarrollo del proyecto.

- Recolectar información acerca de los sistemas de control para automatismos robóticos basados en industria 4.0 y tarjetas empotradas de bajo costo.
- Recolectar información acerca del software de simulación unity 3d.
- Identificar los parámetros cinemáticos del brazo manipulador SCORBOT-ER 4U.
- Definir las características y detalles del entorno virtual.
- Diseñar el entorno virtual con las características establecidas.
- Definir el protocolo de comunicación a utilizar para interactuar entre el medio virtual y el medio físico.
- Implementar el protocolo de comunicación definido y realizar pruebas de funcionamiento.
- Tomar datos acerca de los errores en la comunicación.
- Analizar los errores encontrados y buscar soluciones para eliminar los mismos.
- Realizar pruebas de funcionamiento finales integrando todas las etapas del sistema de control.
- Elaborar un informe final.

## CAPÍTULO III. RESULTADOS Y DISCUSIÓN

### 3.1. Análisis y discusión de los resultados

#### 3.1.1. Desarrollo de la propuesta

Para la presente investigación se ha propuesto el desarrollo de un sistema de control para el brazo robótico Scorbot ER 4U, desde un entorno virtual creado en Unity 3D y con la utilización de las gafas de realidad aumentada meta II, el controlador del brazo es la tarjeta SBC raspberry pi la cual interactúa con el manipulador robótico a través de los pines de entrada/salida GPIO. Para la comunicación entre el entorno virtual y el controlador se utiliza el protocolo de comunicación MQTT. El trabajo de investigación aborda las siguientes etapas:

**Etapla 1: Desarrollo de la Cinemática del Brazo Scorbot ER 4U.** Según el cumplimiento de los objetivos la primera etapa consiste en el desarrollo de la cinemática directa e indirecta del brazo manipulador Scorbot.

La cinemática del Robot Scorbot ER 4u se enfoca en el estudio del movimiento del brazo robótico respecto a un sistema de referencia fijo, el modelo cinemático busca presentar una descripción analítica del cambio de posición de sus articulaciones en los tres ejes coordenados (x,y,z) en función del tiempo y así conocer la posición y la orientación final del efector (pinza), o viceversa a partir de la orientación y posición del efector o pinza encontrar el cambio de posición de las articulaciones, en base a lo descrito se estudia la cinemática directa o la cinemática inversa.

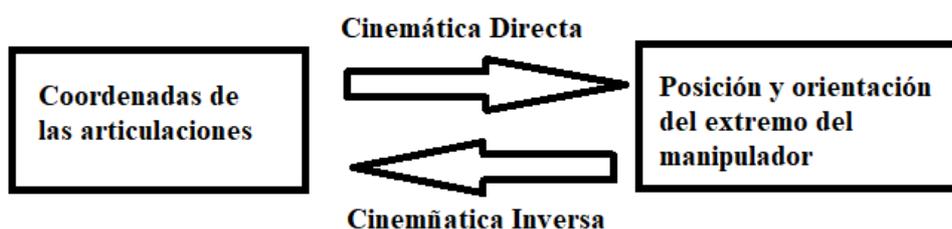


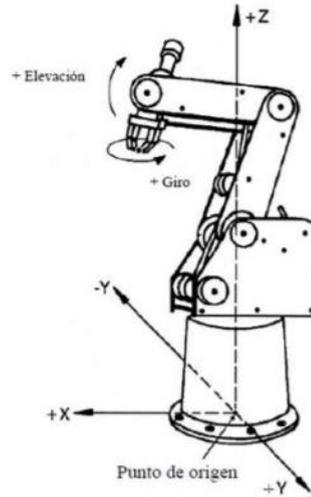
Figura N° 14: Relación de la Cinemática Inversa y Directa.

#### Cinemática directa

La cinemática directa del manipulador Scorbot ER 4U consiste en determinar cuál es la posición y orientación del extremo final de brazo robótico (pinza) respecto a un sistema de coordenadas global fijo. Para lograr determinar estos parámetros es

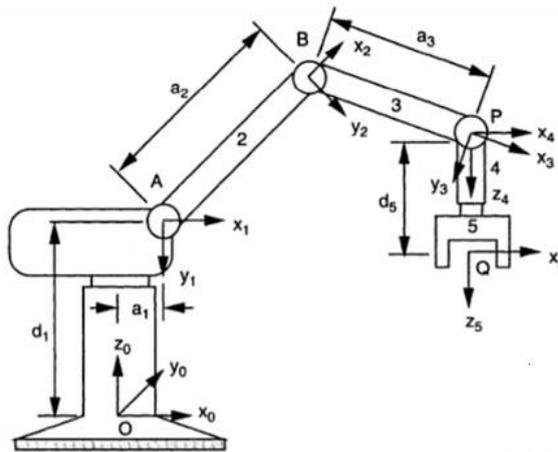
necesario conocer el valor de las articulaciones y los parámetros geométricos del manipulador es decir las longitudes de la base, hombro, codo y muñeca.

Para resolver el modelo cinemático directo se tiene como base el algoritmo de Denavit Hartenberg aplicado al brazo robótico Scorbot ER 4U. Para lo cual se toma como origen del sistema de referencia al punto central de la base del manipulador.



**Figura N° 15:** Punto de Referencia

En la figura 16 se muestra un análisis de la geometría del Scorbot ER 4U con el fin de resolver la cinemática directa.



**Figura N° 16:** Geometría Scorbot ER 4U

Los parámetros necesarios para resolver el algoritmo propuesto se toman de las especificaciones del robot [38]:

$$d_1 = 340mm$$

$$a_1 = 15.7mm$$

$$a_2 = 220mm$$

$$a_3 = 220mm$$

$$d_5 = 1350mm$$

Los parámetros D-H se ilustran en la tabla 8:

**Tabla 13:** Parámetros Denavit - Hartenberg

Articulación (i)	$d_i$	$\theta_i$	$A_i$	$\alpha_i$
1	$d_1$	$\theta_1$	$a_1$	$-\frac{\pi}{2}$
2	0	$\theta_2$	$a_2$	0
3	0	$\theta_3$	$a_3$	0
4	0	$\theta_4$	0	$-\frac{\pi}{2}$
5	$d_5$	$\theta_5$	0	0

Representación de cada parámetro:

- $A_i$  = Distancia entre el origen de dos juntas 01,02
- $d_i$  = Coordenada del origen 01 a lo largo de  $Z_i$
- $\alpha_i$  = El ángulo entre los ejes  $Z_{i-1}$  y  $Z_i$  en referencia al eje  $X_i$ , se toma positivo cuando la rotación es en sentido horario.
- $\theta_i$  = El ángulo entre los ejes  $X_{i-1}$  y  $X_i$ , en referencia al eje  $Z_{i-1}$ , se toma positivo cuando la rotación es en sentido horario.

Con referencia a la matriz de transformación homogénea y los parámetros D-H se obtiene las matrices entre cada sistema de referencia siendo  $S\theta = \sin\theta$  y  $C\theta = \cos\theta$  [44].

$${}^0 A_1 = \begin{bmatrix} C\theta_1 & 0 & -S\theta_1 & a_1 C\theta_1 \\ S\theta_1 & 0 & C\theta_1 & a_1 S\theta_1 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1 A_2 = \begin{bmatrix} C\theta_2 & -S\theta_2 & 0 & a_2 C\theta_2 \\ S\theta_2 & C\theta_2 & 0 & a_2 S\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2 A_3 = \begin{bmatrix} C\theta_3 & -S\theta_3 & 0 & a_3 C\theta_3 \\ S\theta_3 & C\theta_3 & 0 & a_3 S\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^3 A_4 = \begin{bmatrix} C\theta_4 & 0 & -S\theta_4 & 0 \\ S\theta_4 & 0 & C\theta_4 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^4 A_5 = \begin{bmatrix} C\theta_5 & -S\theta_5 & 0 & 0 \\ S\theta_5 & C\theta_5 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Si siguiendo la metodología de la matriz de transformación homogénea T para encontrar la matriz que relaciona el extremo final con el sistema de referencia base se multiplica las matrices de transformación entre cada sistema de referencia interno [44].

$$T = {}^0 A_5 = {}^0 A_1 {}^1 A_2 {}^2 A_3 {}^3 A_4 {}^4 A_5$$

$$T = \begin{bmatrix} n_x & o_x & r_x & p_x \\ n_y & o_y & r_y & p_y \\ n_z & o_z & r_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Los elementos resultantes son:

$$n_x = C\theta_1 C(\theta_2 + \theta_3 + \theta_4) C\theta_1 + S\theta_1 S\theta_5$$

$$n_y = S\theta_1 C(\theta_2 + \theta_3 + \theta_4) C\theta_5 - C\theta_1 S\theta_5$$

$$n_z = -S(\theta_2 + \theta_3 + \theta_4) C\theta_5$$

$$o_x = -C\theta_1 C(\theta_2 + \theta_3 + \theta_4) S\theta_5 + C\theta_5 S\theta_1$$

$$o_y = -S\theta_1 C(\theta_2 + \theta_3 + \theta_4) S\theta_5 + C\theta_1 S\theta_5$$

$$o_z = -S(\theta_2 + \theta_3 + \theta_4) S\theta_5$$

$$r_x = -C_1 S(\theta_2 + \theta_3 + \theta_4)$$

$$r_y = -S_1 S(\theta_2 + \theta_3 + \theta_4)$$

$$r_z = -C(\theta_2 + \theta_3 + \theta_4)$$

$$p_x = -C\theta_1[S(\theta_2 + \theta_3 + \theta_4)d_5 - a_3C(\theta_2 + \theta_3) - a_2C\theta_2 - a_1]$$

$$p_y = -S\theta_1[S(\theta_2 + \theta_3 + \theta_4)d_5 - a_3C(\theta_2 + \theta_3) - a_2C\theta_2 - a_1]$$

$$p_z = -C(\theta_2 + \theta_3 + \theta_4)d_5 - a_3S(\theta_2 + \theta_3) - a_2S\theta_2 - d_1]$$

Finalmente, la orientación y la posición del extremo final están determinadas por las siguientes matrices:

$$R_0^5 = \begin{bmatrix} n_x & o_x & r_x \\ n_y & o_y & r_y \\ n_z & o_z & r_z \end{bmatrix} \text{ Matriz de rotación}$$

$$P = \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} \text{ Matriz posición extremo final}$$

### Cinemática Inversa

Para el análisis de la cinemática inversa se ocupa trigonometría es decir aplicando un análisis trigonométrico se calcula los ángulos que las articulaciones adoptan tomando como referencia el punto del extremo del robot y las longitudes de los eslabones que componen el manipulador Scorbobot ER 4U.

En la figura 17 se presenta los puntos de referencia para el análisis de la cinemática inversa.

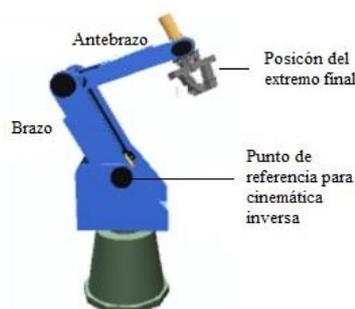
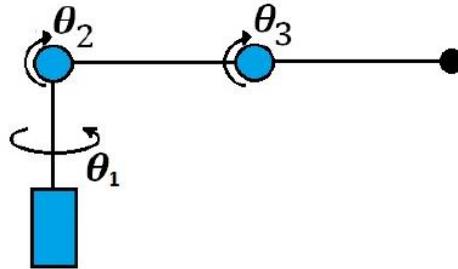


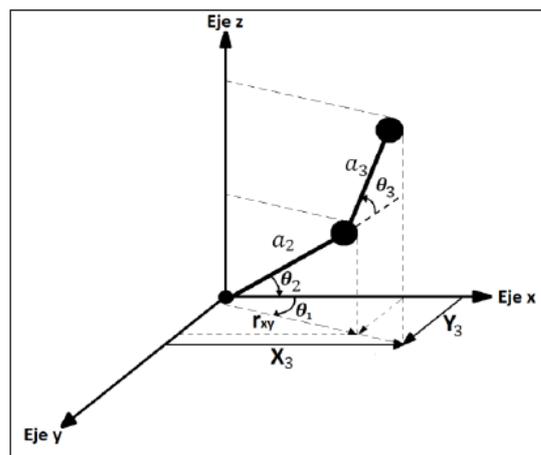
Figura N° 17: Referencias para cinemática inversa

En base a las referencias planteadas se presenta un esquema general para el cálculo de la cinemática inversa:



**Figura N° 18:** Esquema general de articulaciones

Realizando el análisis trigonométrico podemos determinar las ecuaciones para determinar el valor de los ángulos  $\theta_1$ ,  $\theta_2$ ,  $\theta_3$ . En la figura No. 19 se ilustra el desarrollo de los triangulo para el análisis trigonométrico.



**Figura N° 19:** Análisis trigonométrico

Se calcula  $\theta_1$

$$\theta_1 = \arctan \frac{y_3}{y_2}$$

Para  $\theta_2$

$$R_{xy} = \frac{x_3}{\cos \theta_1}$$

$$R_{xyz} = \sqrt{R_{xy}^2 + z_3^2}$$

$$\beta = \arctan \frac{z_3}{R_{xy}}$$

$$\alpha = \arccos \left[ \frac{a_2^2 + R_{xy}^2 * a_2^2}{2 * a_2 * R_{xyz}} \right]$$

$$\theta_2 = \beta - \alpha$$

Finalmente se calcula  $\theta_3$

$$A_{2y} = A_2 * \text{sen}\theta_2$$

$$Z_2 = A_{2y}$$

$$A_{2x} = A_2 * \text{cos}\theta_2$$

$$XY_2 = A_{2x}$$

$$A_{3xy} = r_{xy} - A_{2x}$$

$$A_{3z} = Z_3 - Z_2$$

$$\tan\gamma = \frac{A_{3xy}}{A_{3z}}$$

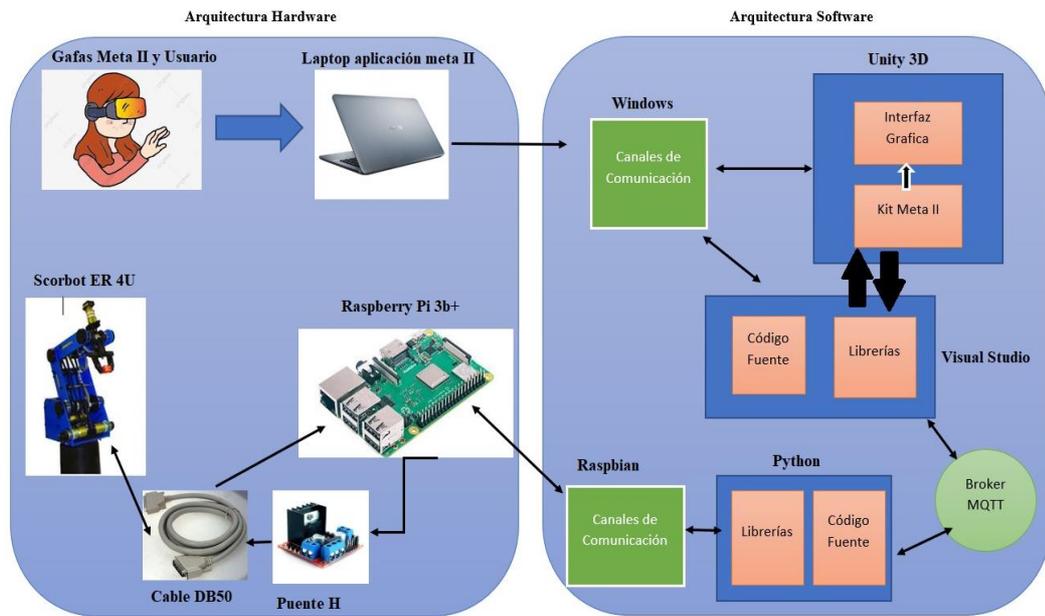
$$\gamma = \arctan \frac{A_{3xy}}{A_{3z}}$$

$$\theta_3 = \gamma - \theta_2$$

**Etapa 2: Diseño del Sistema.** Esta etapa contempla el diseño de la arquitectura del sistema tanto en nivel hardware como en software, se debe describir todos los componentes que estarán conformando el sistema de control, para la descripción se hace uso de los diagramas UML, tanto diagrama de componentes como diagrama de clase.

### **Arquitectura global del sistema de control**

Con base en el cumplimiento del objetivo planteado que es el desarrollo de un método de control para el robot Scorbobot ER 4U utilizando tarjetas empotradas de bajo costo, en la figura No.20 se muestra la estructura general del sistema de control.

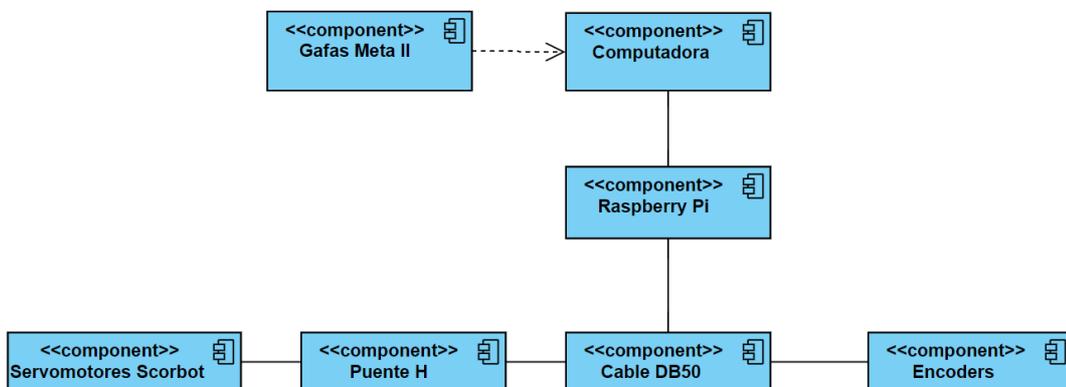


**Figura N° 20:** Arquitectura de software y hardware

En la figura se puede observar la representación del sistema global en la que hay una relación entre componentes de arquitectura hardware y arquitectura software, e interacción humana con estos componentes.

### Arquitectura Hardware

La arquitectura hardware consiste en el diseño de una red de comunicación física que permite la interacción de todos los componentes hardware del sistema apoyados en el software. En la figura No.21 se tiene un diagrama de componentes de los medios físicos que componen al sistema de control propuesto.



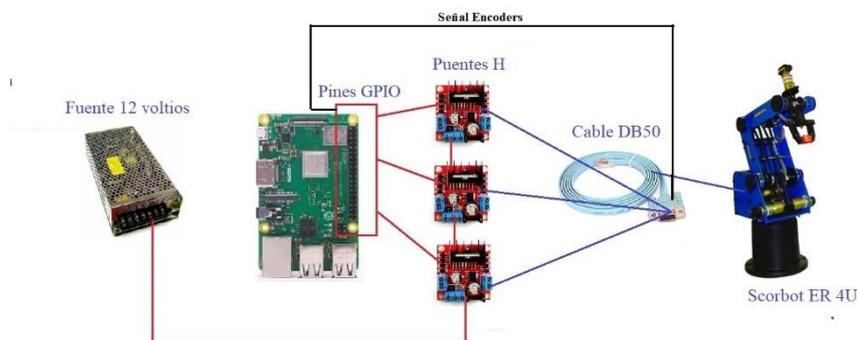
**Figura N° 21:** Diagrama de Componentes físicos sistema de control Scrobot ER 4U

Para la interacción del usuario con el sistema se utiliza el visor de realidad aumentada Meta II, estas gafas inteligentes ofrecen la capacidad de realizar seguimiento en tiempo real, estable y preciso, brindan un mapeo del entorno y siguen la ubicación real del usuario. Las gafas a su vez interactúan con el mundo virtual a través de la computadora mediante el uso de canales de comunicación como son el puerto HDMI y el puerto USB3.0. De esta manera el usuario al colocarse las gafas de realidad virtual podrá visualizar al robot Scorbot en 3D, teniendo la posibilidad de cambiar la posición del extremo final del robot simplemente indicando hacia donde debe dirigirse mediante el movimiento de su mano.

Raspberry pi a su vez interactúa con el brazo Scorbot para dar accionamiento a los servomotores y recibir la señal que provienen de los encoders, para dicha interacción la tarjeta SBC utiliza los pines de entrada/salida GPIO. La tarjeta empotrada también está conectada con el mundo virtual por medio de la computadora, a través de los puertos de comunicación ethernet o wifi y apoyado en el protocolo de transmisión y recepción de datos MQTT recibe todas las señales que provienen desde el pc.

Entre raspberry pi, los actuadores y sensores del brazo Scorbot existen dos periféricos adicionales para la comunicación, puentes H necesarios para elevar la corriente y voltaje de la señal que envía la tarjeta empotrada, y el cable DB50 que es el que conecta directamente con los elementos del robot, además cabe mencionar que para tener el abasto suficiente de corriente y voltaje es imprescindible el uso de una fuente auxiliar.

En la figura No.22 se tiene la representación de la arquitectura de hardware entre raspberry pi y el brazo manipulador Scorbot



**Figura N° 22:** Arquitectura Hardware Raspberry pi y Scorbot

## **Arquitectura de Software**

La arquitectura de software ofrece un panorama referencial para la construcción del sistema de control del manipulador Scorbot cumpliendo así los objetivos propuestos.

### **Nivel de Interfaz**

Como se puede observar en la figura No.20 el diseño del sistema está compuesto por un nivel de interfaz con el usuario, el cual se desarrolla en Unity 3D, en este nivel se agregan todos los componentes gráficos que el usuario podrá visualizar (diseños 3D, botones y etiquetas), también se agregan todos los paquetes prefabricados para uso de las gafas de realidad aumentada de meta II.

### **Niveles de Desarrollo**

Consta de dos niveles de ensamble, el primero, se desarrolla en Visual Studio y permite la codificación de las instrucciones en lenguaje C# para el control y movimiento del modelo 3D de Scorbot el cual fue previamente diseñado, instrucciones para entablar comunicación bajo el protocolo MQTT y código para interactuar con las gafas meta II, además, en este nivel también se agregan todas las librerías necesarias para la interfaz de usuario, control de objetos en el espacio de trabajo y comunicaciones.

El segundo nivel de ensamble se desarrolla en el IDE de raspbian Thonny que es un gestor de código para lenguaje Python, en este ensamblador se codifican las instrucciones para envío y recepción de señales a través de los pines GPIO, logrando así interactuar con el robot Scorbot, además, código para establecer comunicación a través de MQTT, y finalmente se agregan todas las librerías necesarias para el uso de los pines GPIO de raspberry y librerías de MQTT.

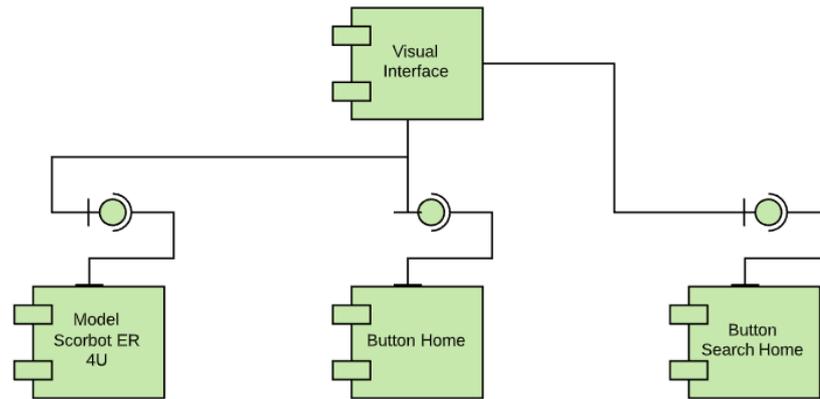
### **Niveles base**

Los niveles base de software como son Windows y Raspbian necesarios para comunicación de los medios hardware con el sistema de control.

### **Diagrama de componentes**

#### **Entorno virtual**

El diagrama de componentes del entorno virtual muestra los distintos objetos con los que cuenta la aplicación.



**Figura N° 23:** Diagrama de componentes interfaz virtual

**Modelo Scorbob ER 4U:** Dentro de la interfaz gráfica se puede visualizar el modelo 3D de Scorbob, que por medio del dispositivo de realidad aumentada se puede interactuar con el mismo.

**Botón Home:** Se puede visualizar un botón con la etiqueta “Posición Home”, que al presionar envía al robot a su posición home.

**Botón Buscar Home:** Dentro de la interfaz se encuentra un botón con la etiqueta “Buscar Home” que al presionar envía un mensaje al microcontrolador para que el robot físico inicie su búsqueda de la posición home.

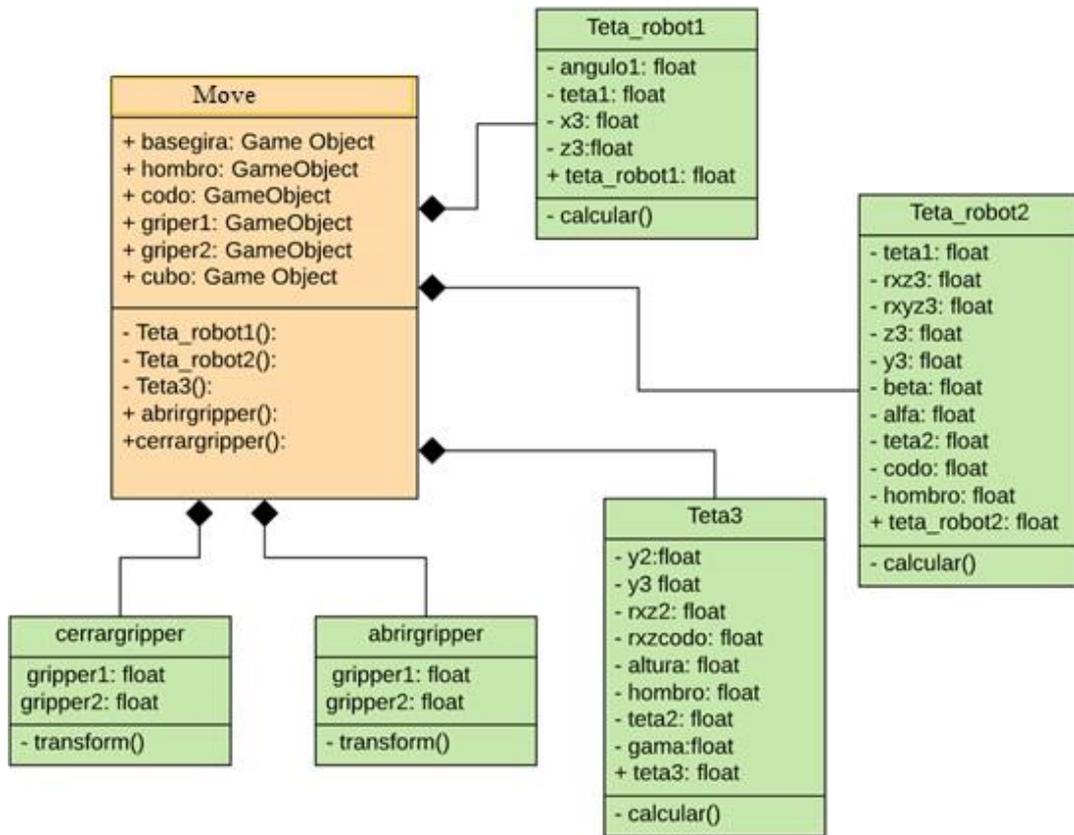
### **Diagramas de Clase**

Los diagramas de clase se realizan basados en el método de funcionamiento de cada uno de los ensambladores de código.

- **Unity 3D**

Basa su funcionamiento en la manipulación de objetos en 3D, los cuales se encuentran dentro de una escena, para poder manipular los objetos se requiere de codificación de scripts en lenguaje C Sharp. Dichos scripts implementan una clase con atributos y métodos. Para el desarrollo del sistema de control propuesto se han desarrollado scripts tanto para el control y manipulación del robot como para la comunicación con la tarjeta empujada. A continuación, se presentan los diagramas de clase:

➤ **Movimiento Scorbot ER 4U**



**Figura N° 24:** Diagrama de Clases Movimiento Scorbot

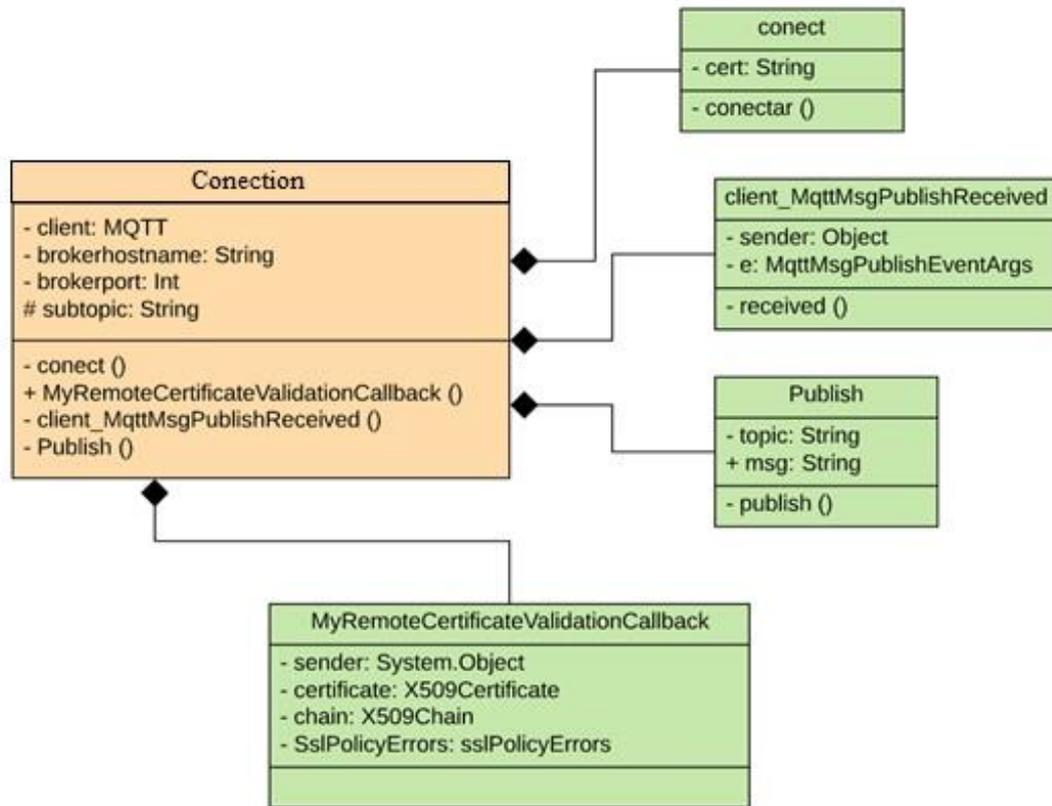
La figura No 24 muestra las clases que se utilizadas para el control del brazo Scorbot, cada clase se indica con sus respectivos atributos y métodos, además existe una relación de composición entre las clases como se puede observar la clase “movimiento” predomina sobre las demás es decir sin la clase “movimiento” las demás no pueden seguir funcionando de manera independiente. La descripción de cada clase se presenta en la siguiente tabla:

**Tabla 14:** Descripción de las clases movimiento Scorbot Er 4U

Clase	Descripción
<b>Movimiento</b>	Es la clase principal para el control del brazo Scorbot, en ella se encuentra la codificación para poder acceder a cada elemento de tipo Game Object que se encuentra en la escena de la aplicación.

	<p>Cada elemento Game Object proporciona atributos a las subclases, estos atributos son necesarios para realizar el cálculo de los valores que deben tomar las articulaciones del brazo en base a la cinemática ya calculada en la etapa 1.</p>
<b>Teta_robot1</b>	<p>Clase que contiene la codificación de los cálculos necesarios para encontrar el valor que debe tomar la articulación 1, todos sus atributos son de tipo float, los mismos que se obtienen a partir de la clase principal.</p>
<b>Teta_robot3</b>	<p>Clase para determinar el valor que debe tomar la articulación 2, todos sus atributos son de tipo float, y al igual que la clase anterior sus atributos son tomados a partir de clase principal.</p>
<b>Teta3</b>	<p>Clase que determinar el valor que debe tomar la articulación 3, a partir de los atributos de tipo float que proporciona la clase principal.</p>
<b>Abrirgripper</b>	<p>Clase que contiene la codificación para abrir el gripper. Sus atributos son de tipo float que proporciona la clase principal.</p>
<b>Cerrargripper</b>	<p>Clase que contiene la codificación para cerrar el gripper. Sus atributos son de tipo float que proporciona la clase principal.</p>

➤ **Comunicación**



**Figura N° 25:** Diagrama de clases comunicación

La figura No. 25 muestra las clases que se utilizan para la comunicación MQTT, cada clase cuenta con sus respectivos atributos y métodos, es importante destacar que existe una relación de composición entre la clase “comunicación” y el resto de clases, es decir que las clases no pueden funcionar independientemente sin la clase “comunicación”.

A continuación, se describe cada clase en la siguiente tabla:

**Tabla 15:** Descripción de las clases comunicación

Clase	Descripción
<b>Comunicación</b>	Es la clase principal para establecer comunicación MQTT, realiza la conexión a un Broker y se suscribe a un tópico.
<b>Conect</b>	Clase la cual establece los parámetros necesarios para entablar

	comunicación, requiere de atributos de tipo Sting, mqtt e int, los cuales proporciona la clase principal, realiza una sola acción que es inicializar la comunicación.
<b>client_MqttMsgPublishReceived</b>	Clase para recibir mensajes que proviene del tópicos al cual se ha conectado, requiere de atributos de tipo object y mqtt.
<b>Publish</b>	Clase que permite enviar mensajes a un tópicos, requiere de atributos de tipo Sting.
<b>MyRemoteCertificateValidationCallback</b>	Clase que proporciona los parámetros de seguridad y cifrado para los mensajes que se intercambian dentro de un tópicos en comunicaciones mqtt, requiere de atributos privados de la clase MQTT.

➤ **Principal**

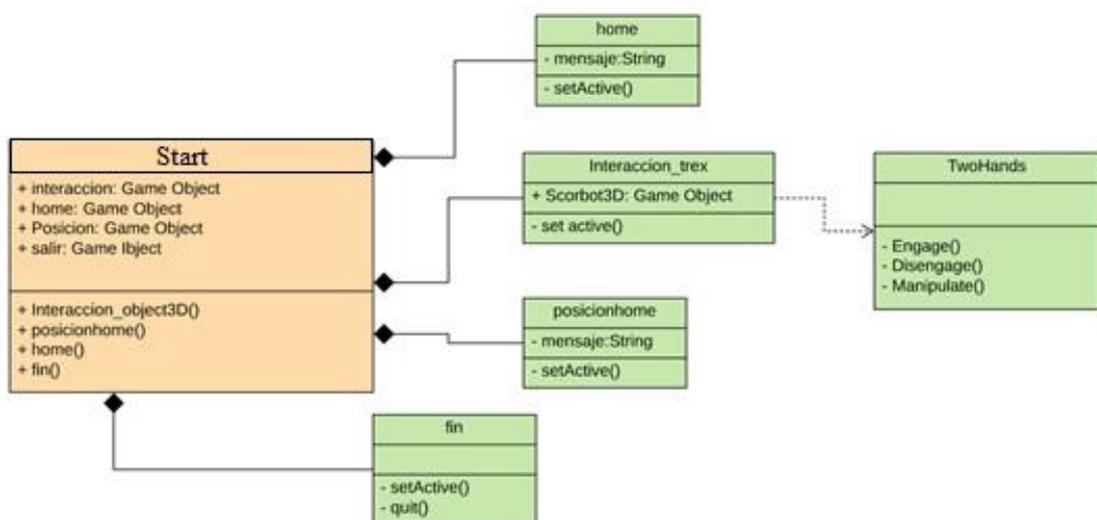


Figura N° 26: Diagrama de clases escena principal

La figura No. 26 muestra las clases que se utilizan para la escena principal, cada clase cuenta con sus respectivos atributos y métodos, es importante destacar que existe una relación de composición entre la clase “principal” y el resto de clases, es decir que las clases no pueden funcionar independientemente sin la clase “principal”.

A continuación, se describe cada clase en la siguiente tabla:

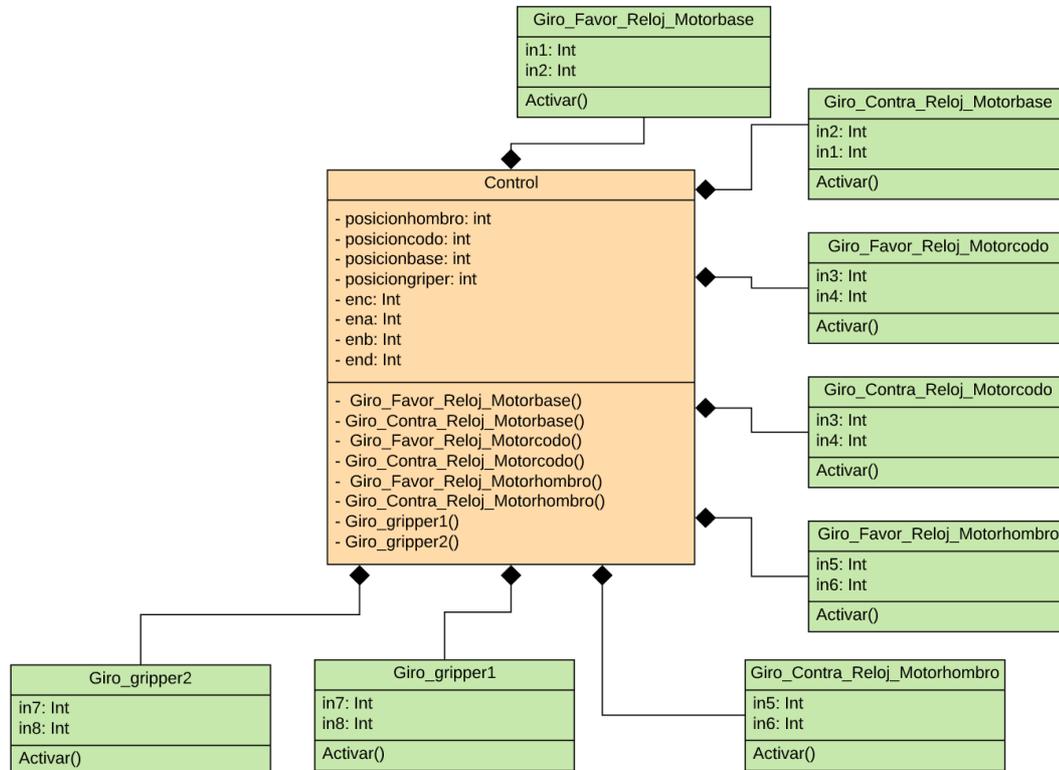
**Tabla 16:** Descripción de las clases principal

Clase	Descripción
<b>Principal</b>	Clase principal del módulo, pues es la que me permite acceder a cada elemento “GameObject” que representan una escena dentro de la aplicación.
<b>Interaccion_Trex</b>	Clase que permite manipular el brazo Scorbot 3D
<b>Fin</b>	Clase que permite cerrar la aplicación
<b>Home</b>	Clase que permite enviar el mensaje para posición home del robot.
<b>Posición home</b>	Clase que permite enviar el mensaje para que el robot busque su posición home.

- **Raspberry PI**

Basa su funcionamiento en la activación de los pines GPIO para envío y recepción de señales digitales. Para poder controlar cada GPIO se utiliza codificación de clases en lenguaje Python. A continuación, se presentan los diagramas de clases de la codificación en raspberry:

## Control GPIOs



**Figura N° 27:** Diagrama de Clases “Control GPIOs”

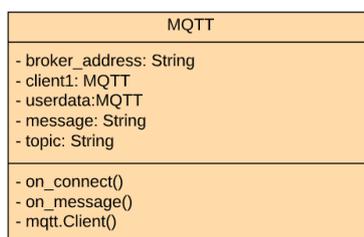
La figura No. 27 muestra las clases que se utilizan para el control de los pines GPIO en raspberry, cada clase cuenta con sus respectivos atributos y métodos, es importante destacar que existe una relación de composición entre la clase “control” y el resto de clases. A continuación, se describe cada clase en la siguiente tabla:

**Tabla 17:** Descripción de las clases Control GPIOs

Clase	Descripción
<b>Control</b>	Clase principal que proporciona a las subclases todos los atributos necesarios para el control de los pines GPIO.
<b>Giro_Favor_Relej_Motorbase</b>	Clase que acciona al motor de la base del manipulador Scorbot en un sentido.

<b>Giro_Contra_Reloj_Motorbasea</b>	Clase que acciona al motor de la base del robot Scrobot en sentido contrario a la clase anterior.
<b>Giro_Favor_Reloj_codo</b>	Clase que acciona al motor del codo del brazo Scrobot en un sentido.
<b>Giro_Contra_Reloj_codo</b>	Clase que acciona al motor del codo del manipulador Scrobot en sentido contrario al de la clase anterior.
<b>Giro_Favor_Reloj_hombro</b>	Clase que acciona al motor del hombro del brazo Scrobot en un sentido.
<b>Giro_Contra_Reloj_hombro</b>	Clase que acciona al motor del hombro del manipulador Scrobot en sentido contrario al de la clase anterior.
<b>Giro_gripper1</b>	Clase que acciona al motor del gripper del manipulador Scrobot en un sentido.
<b>Giro_gripper2</b>	Clase que acciona al motor del gripper del manipulador Scrobot en sentido contrario al anterior.

## Conexión MQTT



**Figura N° 28:** Diagrama de Clase Conexión MQTT

En la figura 28 se representa el diagrama de clase de la comunicación mqtt en raspberry, cuenta con atributos de tipo MQTT como son cliente, y el userdata, también

atributos de tipo String que son lo que definen la dirección IP del Broker, el nombre del tópico, y el mensaje que se recibe. Los métodos que se utilizan son para crear conexión (on\_conecct), para recibir el mensaje (on\_message), y para inicializar el cliente (mqtt.client). Estos son los parámetros necesarios para entablar la comunicación bajo el protocolo MQTT con el entorno de Unity.

### **Etapa 3: Desarrollo de la interfaz en Unity**

En esta sección se lleva a cabo el desarrollo de la aplicación que corresponde a la formación de los objetos virtuales como tal, para ello primeramente se debe diseñar el modelo 3D de Scorbot. Además, de los botos de acción para las posiciones de home del robot y etiquetas. El paquete SDK del visor de realidad aumentada proporciona Prefabs para definir tipos válidos de selección y manipulación permitidos para la escena.

Dentro del entorno virtual se debe posicionar el modelo 3D de Scorbot el cual es construido en el software Blender. Con el modelo ya en el espacio de trabajo se procede a darle características físicas y cinemáticas para el movimiento de cada una de las partes del robot, además se le agrega texturas y así darle el color característico, se agrega un pequeño cubo de color blanco que es un puntero de referencia que indica hacia donde debe dirigirse el extremo final del robot. Finalmente se crean botones y etiquetas que estarán dentro de la escena, también se ha agregado una imagen de fondo para darle a la escena un aspecto agradable acorde a la situación.

En la figura No. 29 se presenta el diseño del entorno virtual.



**Figura N° 29:** Entorno Virtual

#### **Etapa 4: Evaluación y validación**

Con la escena ya finalizada en la aplicación, en esta etapa se realiza el proceso de evaluación del sistema.

En la aplicación con realidad aumentada como herramienta de manipulación del brazo robótico deben considerarse todos los aspectos que puedan afectar su funcionamiento, debido a estas razones es necesario detallar el modo de funcionamiento de la aplicación, Para dar accionamiento al brazo se debe manipular el puntero de referencia con la ayuda de las gafas de realidad aumentada, las mismas que actúan como medio para detectar las manos del usuario y así interactuar con el medio virtual, de esta manera la persona mediante el movimiento de su mano puede indicar el punto hacia donde quiere que se desplace el robot, para lo cual se ha establecido dos señales que el interlocutor debe efectuar para poder desplazar el puntero hacia cualquier posición dentro del espacio virtual. El puntero de referencia es un cuadrado de color blanco que se encuentra en la misma posición que el extremo final del robot. En la figura 30 se muestra donde se encuentra el puntero de referencia.



**Figura N° 30:** Puntero de referencia

En la figura No. 31 se indicará la forma como se debe colocar las manos para tomar y sostener el puntero de referencia.



Forma de tomar el puntero      Forma de soltar el puntero

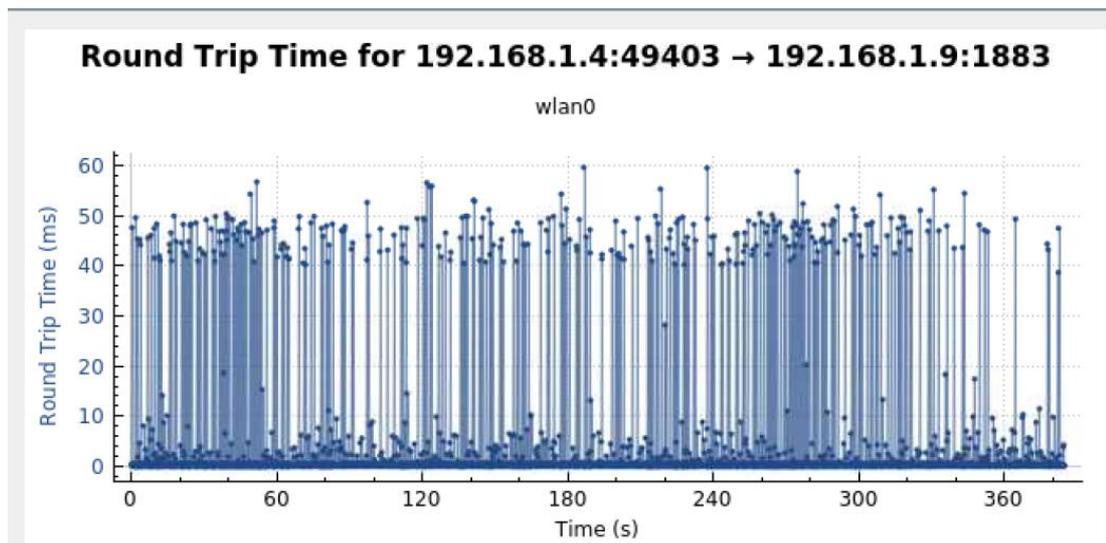
**Figura N° 31:** Tomar y soltar el puntero de referencia

De esta manera se comprueba que cuando el puntero de referencia cambia de posición el robot empieza su movimiento de manera virtual y física a la posición indicada.

### 3.1.2. Análisis y Discusión de los Resultados

Para un tiempo de 10 minutos se obtuvieron los siguientes resultados:

#### Latencia de Red



**Figura N° 32:** Latencia de red

Se puede observar que la latencia de red se mantiene constante con una media de 45.42ms en la mayor parte de la conexión para el tiempo evaluado, existiendo picos altos y bajos constantes teniendo un valor límite superior de 60ms y un valor límite inferior de 1ms

Los resultados obtenidos indican una latencia en red baja, lo cual denota que el protocolo de comunicación establecido es lo suficientemente rápido para realizar procesos en tiempo real, al ser la gráfica constante en la mayor parte del proceso asegura que la respuesta del medio físico responde acorde al medio virtual teniendo una sincronización entre el robot virtual y el robot físico.

### Throughput

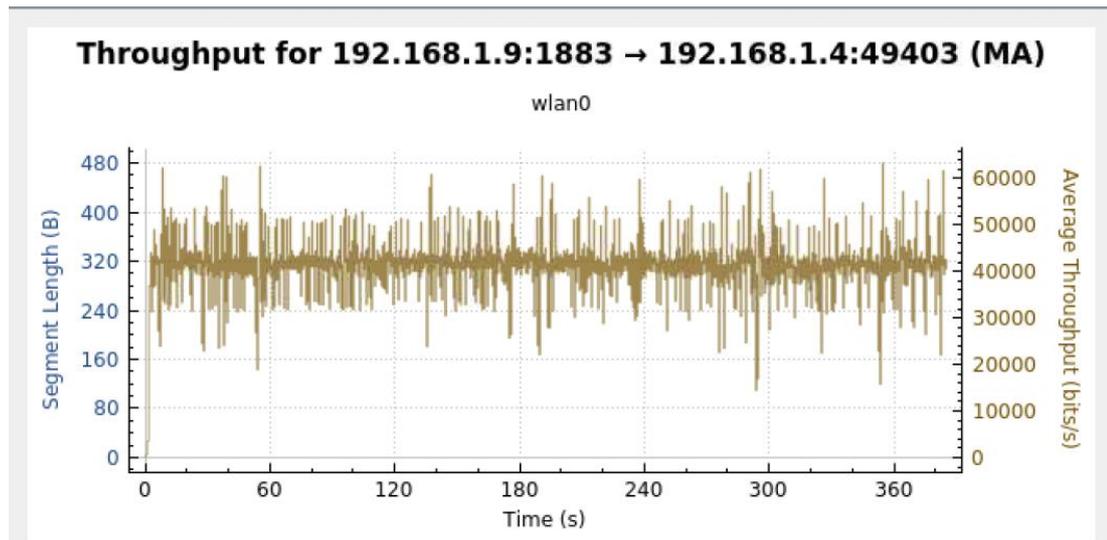


Figura N° 33: Grafica Del Throughput

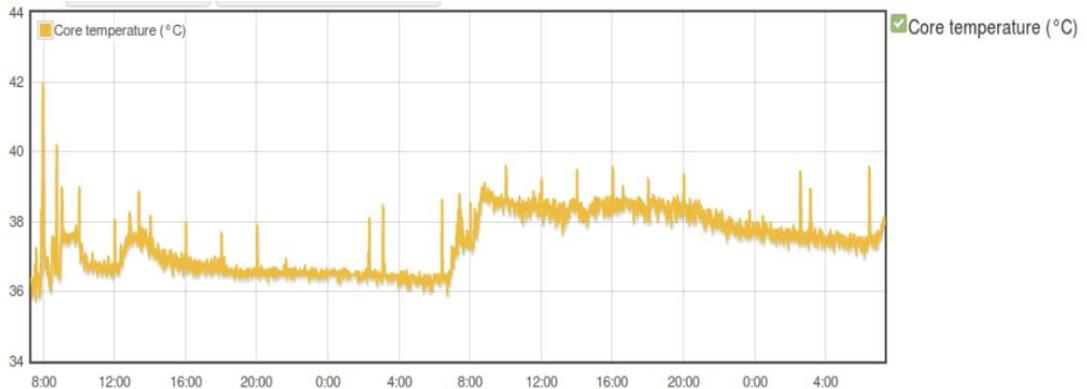
Se puede observar que el throughput mantiene una transmisión de paquetes de datos constante de 40000 bytes por segundo con picos bajos y altos cada cierto tiempo siendo el menor de 10500 bytes por segundo y el mayor de 60000 bytes por segundo.

Los resultados obtenidos para el **Throughput** reflejan que se tiene un ancho de banda suficiente para la transmisión de datos entre unity y raspberry pi, al tener un ancho de banda en mega bits asegura que no haya perdida de datos puesto que los datos a transmitir no sobrepasan los kilobits, esto garantiza que todos los datos que se envían desde el entorno virtual llegan completos hasta la tarjeta empotrada teniendo así el movimiento deseado en el robot Scorbot.

### Temperatura de Raspberry Pi

Para raspberry pi se tomaron datos de la temperatura tanto de la GPU como del procesador, cabe destacar que los datos registrados fueron antes durante y después de establecer la comunicación con unity, esto para apreciar de mejor manera en que

porcentaje aumenta el trabajo de la tarjeta al momento de poner en marcha el sistema de control.



Se puede observar que la gráfica tiene dos etapas perfectamente marcadas, al inicio la temperatura oscila entre 36 y 38 grados centígrados, la segunda etapa la temperatura oscila de 37.5 a 39 grados centígrados.

Los resultados obtenidos reflejan el uso de recursos de la tarjeta cuando el sistema de control se pone en acción, pues para la etapa inicial de la gráfica la tarjeta raspberry se encontraba encendida pero aún no establecía comunicación con la aplicación de unity, para la segunda etapa se puso en marcha la comunicación con unity, y se puede apreciar que la tarjeta empotrada aumenta mínimamente su temperatura lo cual indica que los recursos que ocupa son muy bajos, y conforme el tiempo avanza mantiene la temperatura garantizando que en ningún instante del tiempo la tarjeta llega a utilizar gran cantidad de recursos. Con esto se puede tener la seguridad de que la tarjeta seleccionada es óptima para llevar a cabo la aplicación desarrollada y que en ningún momento puede fallar por sobrecarga de sus recursos.

## **CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES**

### **4.1. Conclusiones**

- La utilización de la tarjeta Raspberry Pi como controlador del brazo robótico Scorbot ER 4U es un medio eficaz para dar movimiento y control al manipulador, pues se adapta perfectamente a la gran mayoría de protocolos de comunicación pudiendo así interactuar con diferentes plataformas y entornos. La velocidad de procesamiento que ofrece es de tipo industrial dando apertura al desarrollo de sistemas ciber físicos bastante robustos y esto a su vez sustenta

las necesidades de la industria 4.0, finalmente trabaja con un sistema de código abierto y software libre lo cual es importante ya que da la posibilidad de operar con sistemas multiplataformas y no requiere de ningún software desarrollado por entes privados convirtiéndola en una gran alternativa confiable y de bajo costo para el control de procesos en cualquier parte de la industria.

- El desarrollo de la cinemática del manipulador es de vital importancia para crear un entorno virtual que proporcione un control confiable y óptimo para el brazo robótico, pues determinar los valores que deben adoptar cada una de las articulaciones así como determinar la posición del extremo final del robot en base a parámetros o datos conocidos permite desarrollar un sin número de aplicaciones para dar cualquier tipo de uso al brazo manipulador, garantizando que este responderá de manera positiva al movimiento deseado, tomar como guía los métodos cinemáticos desarrollados y probados por diferentes autores asegura que no se tendrá errores al momentos de efectuar el control y movimiento del robot Scrobot.
- En la actualidad existen un sin número de aplicaciones virtuales para el control de brazos robóticos, sin embargo el uso de realidad aumentada en estas aplicaciones es muy escaso pues esta tecnología se utiliza en su gran mayoría para la creación de sistemas de entretenimiento, la utilización de gafas de realidad aumentada para manipular al brazo Scrobot ER 4U permite diseñar un entorno virtual amigable, entretenido y de fácil uso para el usuario, la facilidad de indicar con sus manos el punto a donde necesita trasladar el extremo final del robot hace que cualquier persona sin tener ningún conocimiento mayor en sistemas robóticos o de control pueda manipular con facilidad el brazo Scrobot, esto es apto para sistemas de entrenamiento y adiestramiento de personal sobre la manera de operar de los sistemas robóticos.
- Establecer comunicación con el controlador bajo los parámetros de MQTT aseguran éxito en el sistema de control del manipulador Scrobot, en la actualidad se manejan un sin número de protocolos de comunicación que emplean diferentes parámetros y brindan ciertas ventajas a la hora de establecer comunicaciones, internet de las cosas a entrado con gran fuerza en los últimos años y se han desarrollado nuevas comunicaciones que se basan principalmente en la calidad de datos y la seguridad con que se transmiten de un lugar a otro,

MQTT es uno de estos protocolos, la utilización de este sistema de comunicación hace que el objetivo planteado se lleve a cabo con éxito debido a que proporciona seguridad y calidad en la transmisión del mensaje desde el entorno virtual hasta la tarjeta controladora, no existe pérdida de información y su método de operar bajo tópicos jerárquicos garantiza que solo los clientes conectados al tópico reciban la información evitando así desviar u ocultar mensajes.

- A partir de la presente investigación se puede desarrollar aplicaciones para monitorización de parámetros en tiempo real de sistemas de control ya que al integrar realidad aumentada se puede visualizar en la pantalla de trabajo información importante, además se puede desarrollar un sistema para manipulación de cargas a través de brazos manipuladores en ambientes peligrosos sin la necesidad de que la persona se someta a estas condiciones pues con la utilización de un controlador de bajo costo y utilizando el sistema de comunicación implementado se puede manipular el brazo robótico a distancia y si se agrega realidad aumentada de una manera muy fácil e intuitiva.

#### **4.2. Recomendaciones**

- Al trabajar con raspberry largos periodos de tiempo puede significar que la tarjeta tienda a subir de temperatura en el procesador por ello es recomendable utilizar disipadores de calor o implementar ventiladores, garantizando así que la placa trabaje en su temperatura optima evitando colapsos de la misma o pérdida de datos en ciertos periodos de tiempo
- A la hora de instalar Unity para trabajar con las gafas de realidad aumentada de Meta II es recomendable verificar la versión que mas se adapta al uso de estas gafas, a pesar de que estas se lanzaron a la venta hace un par de años tienden a no ser compatibles con las últimas versiones de Unity 3D.
- Al momento de realizar las conexiones entre la palca raspberry y el cable DB50 que incorpora el manipulador Scorbot Er 4U se recomienda verificar los voltajes de que recibe la tarjeta empotrada pues los pines GPIO son muy sensibles y el suministrar un valor fuera del rango establecido implica dañar los pines y en el peor de los casos estropear toda la tarjeta.

## C. MATERIALES DE REFERENCIA

### Referencias Bibliográficas

- [1] Melo and Godoy, “Controller Interface for Industry 4.0 based on RAMI 4.0 and OPC UA,” *2019 IEEE Int. Work. Metrol. Ind. 4.0 IoT, MetroInd 4.0 IoT 2019 - Proc.*, pp. 229–234, 2019, doi: 10.1109/METROI4.2019.8792837.
- [2] Bohuslava, Martin, and H. Igor, “TCP/IP protocol utilisation in process of dynamic control of robotic cell according industry 4.0 concept,” *SAMI 2017 - IEEE 15th Int. Symp. Appl. Mach. Intell. Informatics, Proc.*, pp. 217–222, 2017, doi: 10.1109/SAMI.2017.7880306.
- [3] Ivorra, Ortega, Alcaniz, and Garcia-Aracil, “Multimodal Computer Vision Framework for Human Assistive Robotics,” *2018 Work. Metrol. Ind. 4.0 IoT, MetroInd 4.0 IoT 2018 - Proc.*, pp. 18–22, 2018, doi: 10.1109/METROI4.2018.8428330.
- [4] Bokade and Ratnaparkhe, “Video surveillance robot control using smartphone and Raspberry pi,” *Int. Conf. Commun. Signal Process. ICCSP 2016*, pp. 2094–2097, 2016, doi: 10.1109/ICCSP.2016.7754547.
- [5] “ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL Facultad de Ingeniería en Mecánica y Ciencias de la Producción Modelo Cinemático Brazo Robótico de cinco grados de libertad para uso didáctico TRABAJO FINAL DE GRADUACION Materia Integradora Previo la obtención del,” 2016.
- [6] Gonzalez, Alves, Viana, Carvalho, and Basilio, “Supervisory Control-Based Navigation Architecture: A New Framework for Autonomous Robots in Industry 4.0 Environments,” *IEEE Trans. Ind. Informatics*, vol. 14, no. 4, pp. 1732–1743, 2018, doi: 10.1109/TII.2017.2788079.
- [7] McLin, Hoang, Deneke, and McDowell, “A mirror world-based robot control system,” *Proc. - 2017 2nd Int. Conf. Inf. Syst. Eng. ICISE 2017*, vol. 2017-Janua, pp. 74–77, 2017, doi: 10.1109/ICISE.2017.10.
- [8] Mizuchi and Inamura, “Cloud-based multimodal human-robot interaction simulator utilizing ROS and unity frameworks,” *SII 2017 - 2017 IEEE/SICE Int. Symp. Syst. Integr.*, vol. 2018-Janua, pp. 948–955, 2018, doi: 10.1109/SII.2017.8279345.
- [9] Muñoz, La Piedra, and Moctezuma, “PD+G set-Point Controller for the Second and Third Joint of a ER-4u Scorbot Robot,” *Proc. - 2014 IEEE Int. Conf. Mechatronics, Electron. Automot. Eng. ICMEAE 2014*, pp. 173–178, 2015, doi: 10.1109/ICMEAE.2014.36.
- [10] Papoutsidakis, Piromalis, and Chamilothis, “Modern control interface for SCORBOT ER-III robot,” *2014 Int. Symp. Fundam. Electr. Eng. ISFEE 2014*, 2015, doi: 10.1109/ISFEE.2014.7050623.
- [11] Hutton, Sohre, Davis, Guy, and Rosenberg, “An augmented reality motion planning interface for robotics,” *26th IEEE Conf. Virtual Real. 3D User*

- Interfaces, VR 2019 - Proc.*, pp. 1313–1314, 2019, doi: 10.1109/VR.2019.8798010.
- [12] Leutert, Herrmann, and Schilling, “A spatial augmented reality system for intuitive display of robotic data,” *ACM/IEEE Int. Conf. Human-Robot Interact.*, pp. 179–180, 2013, doi: 10.1109/HRI.2013.6483560.
- [13] Williams, Szafir, Chakraborti, and Phillips, “Virtual, Augmented, and Mixed Reality for Human-Robot Interaction (VAM-HRI),” *ACM/IEEE Int. Conf. Human-Robot Interact.*, vol. 2019-March, pp. 671–672, 2019, doi: 10.1109/HRI.2019.8673207.
- [14] Hockstein, Gourin, Faust, and Terris, “A history of robots: From science fiction to surgical robotics,” *J. Robot. Surg.*, vol. 1, no. 2, pp. 113–118, 2007, doi: 10.1007/s11701-007-0021-2.
- [15] González E. and Jiménez Builes, “La robótica como herramienta para la educación en ciencias e ingeniería,” *IE Comun. Rev. Iberoam. Informática Educ.*, no. 10, pp. 31–36, 2009.
- [16] Ruiz Del Solar and R. Salazar, “Introducción a la Robotica,” pp. 1–20.
- [17] B. Williams, “An Introduction to Robotics in Catalonia,” *Catalònia*, no. 23, pp. 28–29, 2019.
- [18] Perception, “KINEMATICS,” *Synthesis (Stuttg.)*, pp. 1–21.
- [19] Kucuk and Bingul, *Robot Kinematics: Forward and Inverse Kinematics*, no. December. 2016.
- [20] P. Rpi, “Contenido,” 2017.
- [21] V. Suárez and G. Hacklab, “Introducción a Raspberry Pi.”
- [22] ALLIED ELECTRONICS & AUTOMATION, “Raspberry Pi RASPBERRY PI 3,” vol. X, pp. 1–7, 2016.
- [23] “Manual ODROID-C2.” [Online]. Available: <https://magazine.odroid.com/es/odroid-c2>. [Accessed: 05-Feb-2020].
- [24] “Jaguarboard : Jaguar One.” [Online]. Available: [http://www.jaguarboard.org/index.php/com\\_virtuemart\\_menu\\_configuration/products/buy/jaguarboard/207/jaguarboard-detail.html](http://www.jaguarboard.org/index.php/com_virtuemart_menu_configuration/products/buy/jaguarboard/207/jaguarboard-detail.html). [Accessed: 05-Feb-2020].
- [25] DFROBOT, “LattePanda ( 2G / 32GB / Without Win10 License ) - The Most Powerful Win10 Dev Board,” 2017.
- [26] “PDF de programación - Manual de Unity 3D - Trabajando con Unity.” [Online]. Available: <https://www.lawebdelprogramador.com/pdf/14130-Manual-de-Unity-3D-Trabajando-con-Unity.html>. [Accessed: 05-Feb-2020].
- [27] “9 plataformas con las que crear juegos para móviles.” [Online]. Available: <https://www.blog.andaluciaesdigital.es/crear-juegos-para-moviles/>. [Accessed: 05-Feb-2020].

- [28] “CRYENGINE V Manual - CRYENGINE V Manual - Documentation.” [Online]. Available: <https://docs.cryengine.com/>. [Accessed: 05-Feb-2020].
- [29] Eterovic Jorge, Cipriano Marcelo, and Nicolet Santiago, “Análisis de Protocolos de Comunicaciones para Internet de las Cosas,” *XX Work. Investig. en Ciencias la Comput.*, pp. 138–141, 2018, doi: <http://hdl.handle.net/10915/67176>.
- [30] “Conociendo MQTT.” [Online]. Available: <https://www.ibm.com/developerworks/ssa/library/iot-mqtt-why-good-for-iot/index.html>. [Accessed: 05-Feb-2020].
- [31] X. Chen, “Constrained Application Protocol for Internet of Things Abstract,” *Wirel. Mob. Netw.*, vol. 857, pp. 1–12, 2014.
- [32] O. Ozturk, “Introduction to XMPP protocol and developing online collaboration applications using open source software and libraries,” *2010 Int. Symp. Collab. Technol. Syst. CTS 2010*, pp. 21–25, 2010, doi: 10.1109/CTS.2010.5478530.
- [33] Silva, Oliveira, and Giraldi, “Introduction to Augmented Reality.”
- [34] “META 2 - Exclusive Augmented Reality Development Kit.” [Online]. Available: <https://www.schenker-tech.de/en/meta-2/>. [Accessed: 05-Feb-2020].
- [35] “Oculus Rift S: visor de realidad virtual para PC optimizados para la realidad virtual | Oculus.” [Online]. Available: [https://www.oculus.com/rift-s/?locale=es\\_ES](https://www.oculus.com/rift-s/?locale=es_ES). [Accessed: 05-Feb-2020].
- [36] “HoloLens 2—Pricing and Options | Microsoft HoloLens.” [Online]. Available: <https://www.microsoft.com/en-us/hololens/buy>. [Accessed: 05-Feb-2020].
- [37] Intelek,, “Robot educativo SCORBOT-ER 4u.”
- [38] “FrontPage - Raspbian.” [Online]. Available: <https://www.raspbian.org/FrontPage>. [Accessed: 05-Feb-2020].
- [39] Luis, Capellino, Jordi, and Linares Pellicer, “Desarrollo de videojuego 3D con Unity,” 2016.
- [40] “Unity.” [Online]. Available: [https://trinit.es/unity/tutoriales/1 Introducci3n Pr3ctica a Unity 3D.pdf](https://trinit.es/unity/tutoriales/1%20Introducci3n%20Pr3ctica%20a%20Unity%203D.pdf). [Accessed: 06-Feb-2020].
- [41] “IDE de Visual Studio, editor de código, Azure DevOps y App Center - Visual Studio.” [Online]. Available: <https://visualstudio.microsoft.com/es/>. [Accessed: 06-Feb-2020].
- [42] “Meta View Acquires Meta IP, Former Qualcomm Exec to Chart ‘new course’ as CEO – Road to VR.” [Online]. Available: <https://www.roadtovr.com/meta-view-acquires-meta-ip-brings-former-qualcomm-exec-chart-new-course/?fbclid=IwAR1Y1YwWF5-3Pvpmg3w5BnxFQCQe8clKCqWVQCLZ5QY0p-pVkdRtiESoru8>. [Accessed: 06-Feb-2020].

- [43] “blender.org - Home of the Blender project - Free and Open 3D Creation Software.” [Online]. Available: <https://www.blender.org/>. [Accessed: 06-Feb-2020].
- [44] Maneetham and Sivhour, “Scorbot-ER 4U Using Forward Kinematics Modelling and Analysis,” vol. 12, no. 2, pp. 162–168, 2018.

## Anexos

### Anexo.1 Código Unity

#### Librerías

```
using UnityEngine;
using UnityEngine.Networking;
using UnityEngine.Networking.Messages;
using System;
using System.Security.Cryptography.X509Certificates;
using System.Net.Security;
```

#### Código Cinemática Scorbot

```
//Teta 1
float angulo1 = Mathf.Atan(x3 / z3);
float teta1 = (Mathf.Rad2Deg * angulo1);
teta_robot = Mathf.Round( teta1 ) + 40;
//Teta 2
//Calculo rxz3
float rxz3 = z3 / (Mathf.Cos(teta1 * Mathf.PI * 2.0f / 360));
//calcula Rxyz3
float rxyz3 = Mathf.Sqrt((rxz3 * rxz3) + (y3 * y3));
//cálculo beta
float beta1 = Mathf.Atan(y3 / rxz3);
float beta = Mathf.Rad2Deg * beta1;
//Calculo alfa
float alfa1 = Mathf.Acos(((codo1 * codo1)+(rxyz3*rxyz3)- (hombro1 *
hombro1))/(2*codo1*rxyz3));
```

```

float alfa = Mathf.Rad2Deg * alfa1;
float teta2 = beta + alfa;
teta2_robot = Mathf.Round( teta2 * (-1));
//calculo teta 3
//Calculo y2, rxz2, rxzcodo y altura
float y2 = hombro1 * Mathf.Sin(teta2* Mathf.PI * 2.0f / 360);
float rxz2 = hombro1 * Mathf.Cos(teta2* Mathf.PI * 2.0f / 360);
float rxzcodo = rxz3 - rxz2;
float altura = y2 - y3;
//calculo de gama
float gama1 = Mathf.Atan(rxzcodo / altura);
float gama = Mathf.Rad2Deg * gama1;
//calculo de teta 3
teta3 = Mathf.Round( gama - beta + alfa);

```

#### Conexión MQTT

```

private MqttClient client;
private string brokerHostname = "IP Broker";
private int brokerPort = 1883;
static string subTopic = "Scorbot";

```

#### void Start()

```

{
    if (brokerHostname != null )
    {
        Debug.Log("connecting to " + brokerHostname + ":" + brokerPort);
        Connect();
        client.MqttMsgPublishReceived += client_MqttMsgPublishReceived;
        byte[] qosLevels = { MqttMsgBase.QOS_LEVEL_AT_LEAST_ONCE };
        client.Subscribe(new string[] { subTopic }, qosLevels);
    }
}

```

## **Anexo.2 Código Raspberry Pi**

```
import RPi.GPIO as GPIO
import os
import time
import paho.mqtt.client as mqtt
import json

broker_address="192.168.0.101"
client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message
#nos conectamos
client.connect(broker_address, 1883, 60)
client.loop_forever()

in1 = 23
in2 = 24
in3 = 27
in4 = 22
in5 = 16
in6 = 20
enc = 21
ena = 18
enb = 19

#configura los pines segun el microprocesador Broadcom
```

```

GPIO.setmode(GPIO.BCM)
#configura los pines como salidas
GPIO.setup(in1, GPIO.OUT)
GPIO.setup(in2, GPIO.OUT)
GPIO.setup(in3, GPIO.OUT)
GPIO.setup(in4, GPIO.OUT)
GPIO.setup(in5, GPIO.OUT)
GPIO.setup(in6, GPIO.OUT)
GPIO.setup(ena, GPIO.OUT)
GPIO.setup(enb, GPIO.OUT)
GPIO.setup(enc, GPIO.OUT)
pwm_a = GPIO.PWM(ena,500)
pwm_b = GPIO.PWM(enb,500)
pwm_c = GPIO.PWM(enc,500)
pwm_a.start(0)
pwm_b.start(0)
pwm_c.start(0)

def Giro_Favor_Relej_Motorbase():
    GPIO.output(in1,False)
    GPIO.output(in2,True)
def Giro_Contra_Relej_Motorbasea():
    GPIO.output(in1,True)
    GPIO.output(in2,False)
def Giro_Favor_Relej_hombro():
    GPIO.output(in3,True)
    GPIO.output(in4,False)
def Giro_Contra_Relej_hombroa():
    GPIO.output(in3,False)
    GPIO.output(in4,True)

```

```

def Giro_Favor_Relej_codo():
    GPIO.output(in5,True)
    GPIO.output(in6,False)

def Giro_Contra_Relej_codoa():
    GPIO.output(in5,False)
    GPIO.output(in6,True)

```

### Anexo.3 Pines de Conexión DB50 Scorbot

Servomotores		
Eje	Motor	Numero de Pin en cable DB50
<b>1</b>	+	50
	-	17
<b>2</b>	+	49
	-	16
<b>3</b>	+	48
	-	15
<b>4</b>	+	47
	-	14
<b>5</b>	+	46
	-	13
<b>Gripper</b>	+	45
	-	12
Encoders		
<b>1</b>	GND	33
	P1	5
	Vled	11
	P0	2
<b>2</b>	GND	32
	P1	21
	Vled	27

	P0	1
<b>3</b>	GND	31
	P1	4
	Vled	10
	P0	36
<b>4</b>	GND	30
	P1	20
	Vled	26
	P0	35
<b>5</b>	GND	29
	P1	3
	Vled	9
	P0	18
<b>Gripper</b>	GND	28
	P1	19
	Vled	25
	P0	34
<b>Detectores Inicio de Carrera</b>		
<b>1</b>	GND	33
	+	23
<b>2</b>	GND	32
	+	7
<b>3</b>	GND	31
	+	24
<b>4</b>	GND	30
	+	8
<b>5</b>	GND	29
	+	6