



**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**Robot humanoide controlado remotamente que permita interactuar y manipular objetos del entorno, mediante visión estéreo y teleoperación, en el laboratorio de Mecatrónica**

Catagua Cobos, Josseph Yaakob

Departamento de Ciencias de la Energía y Mecánica

Carrera de Ingeniería Mecatrónica

Trabajo de titulación, previo a la obtención del título de Ingeniero Mecatrónico

Ms.C. Mendoza Chipantasi, Darío José

28 de agosto del 2023

Latacunga



CATAGUA COBOS JOSSEPH YAAKOB TE...

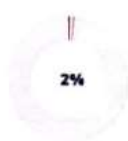
## Scan details

Scan time:  
August 28th, 2023 at 17:11 UTC

Total Pages:  
78

Total Words:  
19458

## Plagiarism Detection



Types of plagiarism		Words
Identical	0.8%	148
Minor Changes	0.2%	46
Paraphrased	1%	201
Omitted Words	0%	0

## AI Content Detection



Text coverage  
 AI text  
 Human text

## 🔍 Plagiarism Results: (27)

🌐 **Estándares Internacionales para la seguridad de los...** 0.4%

<https://roboticahoy.com/estandares-internacionales-para-la-...>

[Saltar al contenido](#) [Menú](#) [Menú](#) [Inicio](#) [Tipos de Robots ...](#)

🌐 **ISO 10218-1** 0.3%

<https://www.en-standard.eu/iso-10218-1-robots-and-robotic-...>

Preorder Form Name: \* E-mail; \* Please Insert your email, we will inform immediately after publishing this standard (include -...

🌐 **GitHub - PacktPublishing/Python-GUI-Programming...** 0.2%

<https://github.com/packtpublishing/python-gui-programmin...>

[Skip to content](#) [Toggle navigation ...](#)

Ms.C. Mendoza Chipantasi, Darío José

TUTOR DE TESIS



Departamento de Ciencias de la Energía y Mecánica

Carrera de Ingeniería Mecatrónica

#### Certificación

Certifico que el trabajo de titulación: **"Robot humanoide controlado remotamente que permita interactuar y manipular objetos del entorno, mediante visión estéreo y teleoperación, en el laboratorio de Mecatrónica"** fue realizado por el señor **Catagua Cobos, Josseph Yaakob**; el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizado en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Latacunga, 28 de Agosto del 2023

Ms.C. Mendoza Chipantasi, Darío José

C.C.: 0603110834



Departamento de Ciencias de la Energía y Mecánica  
Carrera de Ingeniería Mecatrónica

**Responsabilidad de Autoría**

Yo, **Catagua Cobos, Josseph Yaakob**, con cédula de ciudadanía n°0923678767, declaro que el contenido, ideas y criterios del trabajo de titulación: **“Robot humanoide controlado remotamente que permita interactuar y manipular objetos del entorno, mediante visión estéreo y teleoperación, en el laboratorio de Mecatrónica”** es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

**Latacunga, 28 de Agosto del 2023**

---

Catagua Cobos, Josseph Yaakob

C.C.: 0923678767



**Departamento de Ciencias de la Energía y Mecánica**

**Carrera de Ingeniería Mecatrónica**

**Autorización de Publicación**

Yo **Catagua Cobos, Joseph Yaakob**, con cédula de ciudadanía n°0923678767, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **"Robot humanoide controlado remotamente que permita interactuar y manipular objetos del entorno, mediante Visión Estéreo y Teleoperación, en el laboratorio de Mecatrónica"** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

**Latacunga, 28 de Agosto del 2023**

---

Catagua Cobos, Joseph Yaakob

C.C.: 0923678767

### **Dedicatoria**

Dedicado a las personas que han sido mi fuente inquebrantable de inspiración y apoyo en este emocionante viaje académico. A mi querida madre, Mónica Cobos, cuyo amor incondicional, motivación constante y confianza en mí han sido mi guía en cada paso que he dado.

A mi hermano Marco Aranda, quien no solo me ha brindado un futuro lleno de posibilidades, sino que también ha sido mi roca, mi consejero y mi motivación incansable. Tus palabras y apoyo siempre han sido un faro en mi camino.

A mi amiga Mayra López, quien ha sido como una segunda madre para mí. Tu apoyo inquebrantable, cariño sincero, consejos sabios y amistad le han dado profundidad y calidez a esta travesía. Tu presencia ha sido un regalo invaluable.

A mi amada abuelita, Raquel Alvarado, cuyo apoyo y confianza en mí me han dado la fortaleza para seguir adelante incluso en los momentos más desafiantes. Tu sabiduría y amor son tesoros que atesoro en mi corazón.

A todos ustedes, les dedico este logro. Sin su amor, aliento y guía, este camino habría sido mucho más difícil. Cada paso que he dado en esta tesis ha sido impulsado por su amor y confianza en mí. Gracias por ser mi luz y mi inspiración.

## **Agradecimiento**

En este punto culminante de mi viaje académico, deseo expresar mi sincero agradecimiento a aquellos que han sido pilares fundamentales en mi camino hacia el éxito.

A Dios, fuente de sabiduría y fortaleza, agradezco por iluminar mi camino y brindarme la determinación para superar obstáculos y alcanzar mis metas.

A mi querida madre, cuyo amor incondicional y apoyo constante han sido mi roca en cada etapa de esta travesía. Tus sacrificios y aliento han sido el motor que me impulsa hacia adelante.

A mi hermano, compañero de aventuras y amigo fiel, te agradezco por estar a mi lado en cada paso de este camino. Tu apoyo y consejos han sido invaluable.

A mi amiga y segunda madre, quien me ha brindado cariño, orientación y un refugio en momentos de necesidad. Tu presencia ha añadido un toque especial a este viaje y estoy agradecido/a por ello.

A mi tutor de tesis, cuya guía experta y paciencia han sido esenciales para dar forma a mis ideas y llevar a cabo este proyecto. Tu dedicación y conocimiento han sido una inspiración para mí.

A los docentes, cuyo compromiso y pasión por la enseñanza han sido la chispa que encendió mi sed de conocimiento. Sus enseñanzas han dejado una marca duradera en mi camino.

A los administrativos, quienes trabajan incansablemente entre bastidores para asegurar que nuestras experiencias académicas sean fluidas y exitosas. Su labor muchas veces pasa desapercibida, pero es profundamente valorada.

A todos ustedes, les debo un sincero agradecimiento por ser los cimientos en los que se sostiene este logro. Sus contribuciones han dejado una huella imborrable en mi corazón y en este trabajo. Este éxito no habría sido posible sin su apoyo constante y aliento.

## ÍNDICE DE CONTENIDO

Carátula .....	1
Reporte de Verificación del Contenido.....	2
Certificación .....	3
Responsabilidad de Autoría.....	4
Autorización de Publicación .....	5
Dedicatoria .....	6
Agradecimiento.....	7
Índice de Contenido.....	8
Índice de Figuras .....	11
Índice de Tablas .....	15
Resumen.....	17
Abstract .....	18
Capítulo I: Introducción y Marco Teórico.....	19
Antecedentes.....	19
Planteamiento del Problema.....	20
Justificación e Importancia .....	21
Objetivos.....	22
<i>Objetivo General</i> .....	22
<i>Objetivos Específicos</i> .....	22
Hipótesis de Investigación .....	23
Categorización de las Variables de Investigación .....	23
<i>Variable Independiente</i> .....	23
<i>Variable Dependiente</i> .....	23
Marco Teórico.....	23
<i>Normas de Seguridad al Implementar un Robot Humanoide</i> .....	23



<i>Visión Estéreo</i> .....	24
<i>Robot Humanoide</i> .....	26
<i>Marcas Comerciales de Robot Humanoide</i> .....	28
<i>Marcas Comerciales de Controladores O Unidad de Procesamiento</i> .....	32
<i>Análisis de los Movimientos</i> .....	34
<i>Cinemática en Robots</i> .....	37
<i>Conexión Remota</i> .....	38
<i>Aplicaciones de Programación y Comunicación para Computador</i> .....	40
<i>Puntos para el Análisis de Resultados</i> .....	51
<b>Capítulo II: Selección Y Configuración Inicial</b> .....	52
<b>Selección</b> .....	52
<i>Selección del Dispositivo de Cámara</i> .....	52
<i>Selección del Material para el Soporte de Cámara Estéreo</i> .....	57
<i>Selección del Robot Humanoide</i> .....	60
<i>Selección del Controlador o Unidad de Procesamiento</i> .....	61
<b>Configuración Inicial</b> .....	63
<i>Conexión con el Software PyCharm</i> .....	63
<b>Capítulo III: Desarrollo e Implementación</b> .....	68
<b>Diseño Mecánico del Soporte para las Cámaras, Cámara Estéreo</b> .....	68
<b>Diseño de la Interfaz Humano Máquina (HMI)</b> .....	72
<i>Apariencia de las Ventanas</i> .....	72
<i>Tema de las Ventanas</i> .....	72
<i>Disposición</i> .....	73
<b>Código del Programa en Pycharm para la Computadora</b> .....	76
<i>Creación de las Librerías</i> .....	76
<i>Diagrama de Flujo</i> .....	77

<i>Triangulación de los Puntos de Referencia</i> .....	79
<b>Manipulación del Robot por Software</b> .....	79
<b>Relación entre las Articulaciones de la Persona y los GDL del Robot</b> .....	80
<b>Diseño del Entorno de Unity</b> .....	81
<b>Código y Configuración del Programa en Unity</b> .....	82
<i>Configuración Inicial</i> .....	82
<i>Configuración de los Eventos del Mando</i> .....	83
<i>Diagrama de Flujo del Código</i> .....	84
<b>Límites de las Articulaciones del Robot Humanoide</b> .....	86
<i>Límites en el Brazo Izquierdo</i> .....	86
<i>Límites en el Brazo Derecho</i> .....	88
<i>Método de Cinemática Inversa</i> .....	89
<b>Código del Robot en PyCharm</b> .....	91
<b>Capítulo IV: Pruebas, Resultados y Validación de la Hipótesis</b> .....	93
<b>Pruebas y Resultados</b> .....	93
<i>Visión Estéreo</i> .....	93
<i>Control Remoto</i> .....	100
<i>Conversión entre los GDL de una Persona a GDL de un Robot</i> .....	100
<i>Movimientos Gestuales y Espaciales del Robot</i> .....	103
<b>Validación de la Hipótesis</b> .....	108
<b>Capítulo V: Conclusiones y Recomendaciones</b> .....	114
<b>Conclusiones</b> .....	114
<b>Recomendaciones</b> .....	116
<b>Bibliografía</b> .....	117
<b>Anexos</b> .....	121

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> <i>Representación gráfica de Visión Estéreo.</i> .....	25
<b>Figura 2.</b> <i>Disparidad en una cámara estéreo.</i> .....	25
<b>Figura 3.</b> <i>Caso ideal de un sistema estéreo.</i> .....	26
<b>Figura 4.</b> <i>Estabilidad, locomoción bípeda.</i> .....	27
<b>Figura 5.</b> <i>a) Activa-estable, b) Pasiva-dinámica.</i> .....	28
<b>Figura 6.</b> <i>TonyPi Pro Hiwonder robot humanoide, kit de desarrollo.</i> .....	29
<b>Figura 7.</b> <i>Meccanoid G15 Meccano robot humanoide, kit de desarrollo.</i> .....	30
<b>Figura 8.</b> <i>Alpha 1 Pro Ubtech Robotics, robot humanoide.</i> .....	31
<b>Figura 9.</b> <i>Dispositivo Raspberry Pi 4 Modelo B.</i> .....	32
<b>Figura 10.</b> <i>Dispositivo NVIDIA Jetson Nano.</i> .....	33
<b>Figura 11.</b> <i>Dispositivo BeagleBone Black.</i> .....	34
<b>Figura 12.</b> <i>Tipos de articulaciones.</i> .....	36
<b>Figura 13.</b> <i>Logotipo de la aplicación Pycharm Profesional.</i> .....	40
<b>Figura 14.</b> <i>Logotipo del lenguaje de programación Python.</i> .....	41
<b>Figura 15.</b> <i>Caja o ventana de mensaje para mostrar información.</i> .....	42
<b>Figura 16.</b> <i>Caja o ventana de mensaje para mostrar una advertencia.</i> .....	42
<b>Figura 17.</b> <i>Caja o ventana de mensaje para mostrar un error.</i> .....	42
<b>Figura 18.</b> <i>Caja o ventana de mensaje para mostrar una pregunta.</i> .....	43
<b>Figura 19.</b> <i>Logotipo de la Biblioteca personalizable para Tkinter, CustomTkinter.</i> .....	44
<b>Figura 20.</b> <i>Logotipo de la biblioteca de código abierto OpenCV.</i> .....	46
<b>Figura 21.</b> <i>Logotipo para la biblioteca de importación CVZone.</i> .....	47
<b>Figura 22.</b> <i>33 puntos de referencia de CVZone con sus aristas.</i> .....	48
<b>Figura 23.</b> <i>Logotipo de la marca y software Unity.</i> .....	49

<b>Figura 24.</b> <i>Logotipo de la marca y software Visual Studio.....</i>	50
<b>Figura 25.</b> <i>Logotipo del lenguaje de programación C#.....</i>	50
<b>Figura 26.</b> <i>Logotipo del programa VNC Viewer.....</i>	51
<b>Figura 27.</b> <i>Genius FaceCam 1000X.....</i>	53
<b>Figura 28.</b> <i>Logitech C920e V-U0028.....</i>	54
<b>Figura 29.</b> <i>SPEEDMIND HD 1080P.....</i>	55
<b>Figura 30.</b> <i>León impreso 3D con filamento ABS.....</i>	57
<b>Figura 31.</b> <i>León impreso 3D con filamento PLA.....</i>	58
<b>Figura 32.</b> <i>León esculpido en madera.....</i>	59
<b>Figura 33.</b> <i>Ventana emergente para configurar el proyecto.....</i>	64
<b>Figura 34.</b> <i>Opción para el entorno por comunicación SSH.....</i>	64
<b>Figura 35.</b> <i>Ventana de configuración para la comunicación con el entorno.....</i>	65
<b>Figura 36.</b> <i>Ventana de configuración de acceso y seguridad.....</i>	65
<b>Figura 37.</b> <i>Opción de configuración general del proyecto en PyChram.....</i>	66
<b>Figura 38.</b> <i>Configuración Deployment Connection para la codificación.....</i>	66
<b>Figura 39.</b> <i>Configuración Deployment Mappings para la codificación.....</i>	67
<b>Figura 40.</b> <i>Representación gráfica de la funcionalidad del proyecto.....</i>	68
<b>Figura 41.</b> <i>Bosquejo de la vista frontal de la cámara, Inventor Autodesk.....</i>	69
<b>Figura 42.</b> <i>Modelo 3D del soporte para las cámaras.....</i>	69
<b>Figura 43.</b> <i>Fuerza que aplica la cámara por la gravedad.....</i>	70
<b>Figura 44.</b> <i>Cargas aplicadas. a) Gravedad b) Por manipulación c) Por las cámaras.....</i>	70
<b>Figura 45.</b> <i>Cargas aplicadas en el soporte de cámara, Inventor Autodesk.....</i>	71
<b>Figura 46.</b> <i>Tensión de Van Mises en el soporte de cámara.....</i>	71
<b>Figura 47.</b> <i>Desplazamiento en el soporte de cámara.....</i>	71

<b>Figura 48.</b> <i>Coeficiente de seguridad en el soporte de cámara.</i> .....	72
<b>Figura 49.</b> <i>Disposición de las herramientas en la ventana principal.</i> .....	74
<b>Figura 50.</b> <i>Disposición de las herramientas en la ventana de ayuda.</i> .....	75
<b>Figura 51.</b> <i>Diagrama de Flujo de PyCharm.</i> .....	78
<b>Figura 52.</b> <i>Aplicación interna para manipular los Servomotores.</i> .....	80
<b>Figura 53.</b> <i>Ventana de la aplicación con controles deslizantes.</i> .....	80
<b>Figura 54.</b> <i>Relación entre GDL. a) Persona detectada b) Robot TonyPi Pro</i> .....	81
<b>Figura 55.</b> <i>Representación en Unity de las articulaciones.</i> .....	82
<b>Figura 56.</b> <i>Posición, rotación y escala.</i> .....	83
<b>Figura 57.</b> <i>Botones que contiene un mando de XBOX genérico.</i> .....	84
<b>Figura 58.</b> <i>Eventos de mando en Input Actions.</i> .....	84
<b>Figura 59.</b> <i>Diagrama de Flujo de Unity.</i> .....	85
<b>Figura 60.</b> <i>Representación gráfica de los límites para el brazo izquierdo.</i> .....	87
<b>Figura 61.</b> <i>Representación gráfica de los límites para el brazo derecho.</i> .....	89
<b>Figura 62.</b> <i>Configuración de Hybrid IK, del brazo izquierdo.</i> .....	90
<b>Figura 63.</b> <i>Configuración de límites para el brazo izquierdo.</i> .....	90
<b>Figura 64.</b> <i>Configuración de Hybrid IK, del brazo derecho.</i> .....	91
<b>Figura 65.</b> <i>Configuración de límites para el brazo derecho.</i> .....	91
<b>Figura 66.</b> <i>Diagrama de Flujo del robot.</i> .....	92
<b>Figura 67.</b> <i>Prueba contra luz captando el exterior.</i> .....	93
<b>Figura 68.</b> <i>Prueba contra luz sin captar el exterior.</i> .....	94
<b>Figura 69.</b> <i>Prueba con iluminación clara.</i> .....	94
<b>Figura 70.</b> <i>Prueba de detección parcial.</i> .....	95
<b>Figura 71.</b> <i>Prueba de detección con varias personas.</i> .....	96

<b>Figura 72.</b> <i>Representación gráfica de la medición de la profundidad.</i> .....	96
<b>Figura 73.</b> <i>Medición vista desde el programa.</i> .....	96
<b>Figura 74.</b> <i>Prueba de comunicación entre PyCharm y Unity.</i> .....	100
<b>Figura 75.</b> <i>Prueba de comunicación entre Unity y TonyPi Pro.</i> .....	100
<b>Figura 76.</b> <i>Representación gráfica, medición de ángulos.</i> .....	101
<b>Figura 77.</b> <i>Representación gráfica, medición de ángulos en el robot.</i> .....	101
<b>Figura 78.</b> <i>Robot agachado.</i> .....	104
<b>Figura 79.</b> <i>Vista frontal del código para levantar al Robot.</i> .....	104
<b>Figura 80.</b> <i>Resultado entre PyCharm y Unity.</i> .....	105
<b>Figura 81.</b> <i>Resultados del robot TonyPi Pro.</i> .....	105
<b>Figura 82.</b> <i>Comparación de coordenadas para la muñeca izquierda.</i> .....	106
<b>Figura 83.</b> <i>Comparación de coordenadas para la muñeca derecha.</i> .....	106
<b>Figura 84.</b> <i>Gesto para abrir los brazos del robot.</i> .....	107
<b>Figura 85.</b> <i>Gesto para cerrar los brazos del robot.</i> .....	107
<b>Figura 86.</b> <i>Robot agarrando una caja de cartón.</i> .....	107
<b>Figura 87.</b> <i>Robot empujando un objeto.</i> .....	108
<b>Figura 88.</b> <i>Pruebas para interactuar y manipular objetos.</i> .....	109

## ÍNDICE DE TABLAS

<b>Tabla 1.</b> <i>Tabla de especificaciones de los elementos del robot TonyPi Pro.</i> .....	29
<b>Tabla 2.</b> <i>Tabla de especificaciones de los elementos del robot Meccanoid G15.</i> .....	30
<b>Tabla 3.</b> <i>Tabla de especificaciones de los elementos del robot Alpha 1 Pro.</i> .....	31
<b>Tabla 4.</b> <i>Características de la cámara Genius FaceCam 1000X.</i> .....	52
<b>Tabla 5.</b> <i>Características de la cámara Logitech C920e V-U0028.</i> .....	53
<b>Tabla 6.</b> <i>Características de la cámara SPEEDMIND HD 1080P.</i> .....	54
<b>Tabla 7.</b> <i>Comparación entre características para las cámaras.</i> .....	56
<b>Tabla 8.</b> <i>Características del filamento termoplástico ABS.</i> .....	57
<b>Tabla 9.</b> <i>Características del filamento de material termoplástico PLA.</i> .....	58
<b>Tabla 10.</b> <i>Características de la madera.</i> .....	59
<b>Tabla 11.</b> <i>Comparación entre características para los materiales.</i> .....	60
<b>Tabla 12.</b> <i>Comparación entre características para los robots humanoides.</i> .....	60
<b>Tabla 13.</b> <i>Características de la unidad de procesamiento Raspberry Pi 4B.</i> .....	61
<b>Tabla 14.</b> <i>Características de la unidad de procesamiento NVIDIA Jetson Nano.</i> .....	62
<b>Tabla 15.</b> <i>Características de la unidad de procesamiento Beaglebone Black.</i> .....	62
<b>Tabla 16.</b> <i>Comparación entre características para los controladores.</i> .....	63
<b>Tabla 17.</b> <i>Límites para el brazo izquierdo.</i> .....	86
<b>Tabla 18.</b> <i>Límites para el brazo derecho.</i> .....	88
<b>Tabla 19.</b> <i>Valores calculados, valor real 99 cm.</i> .....	97
<b>Tabla 20.</b> <i>Valores calculados, valor real 111 cm.</i> .....	97
<b>Tabla 21.</b> <i>Valores calculados, valor real 124 cm.</i> .....	98
<b>Tabla 22.</b> <i>Valores calculados, valor real 136 cm.</i> .....	98
<b>Tabla 23.</b> <i>Valores calculados, valor real 149 cm.</i> .....	99

<b>Tabla 25.</b> <i>Valores calculados, valor real 251 cm.</i> .....	99
<b>Tabla 26.</b> <i>Error de resultado al convertir GDL.</i> .....	102
<b>Tabla 27.</b> <i>Error acumulativo al convertir los GDL.</i> .....	102
<b>Tabla 28.</b> Resultados obtenido por interacción.....	110
<b>Tabla 29.</b> <i>Distribución Chi Cuadrado</i> .....	111
<b>Tabla 30.</b> Resultados obtenido por manipulación. ....	112



## Resumen

Esta tesis se centra en implementar un robot humanoide para controlarlo remotamente, que el robot interactúe y manipule objetos en el entorno mediante Visión Estéreo y Teleoperación, en el Laboratorio de Mecatrónica. Se utiliza el modelo de cámara Genius FaceCam 1000X con dos unidades idénticas para mantener una paridad y perspectiva precisa a una separación de  $63\text{ mm}$  entre ellas. El soporte de las cámaras está diseñado para manejar las cargas aplicadas por las cámaras y la manipulación del operador, con un desplazamiento máximo de  $0.38\text{ mm}$  y un coeficiente de seguridad de  $1.03\text{ unidades}$ . El sistema está respaldado por la Unidad de Procesamiento Raspberry Pi 4B, que facilita la comunicación entre el robot TonyPi Pro y dispositivos externos como computadoras. Se han desarrollado librerías para optimizar el código, reducir repeticiones y mejorar la legibilidad del programa completo. Para lograr una triangulación de profundidad precisa, se establece una distancia mínima de  $111\text{ cm}$  entre la cámara y el punto de referencia de la persona observada, asegurando que el punto de interés esté dentro del rango de observación en la imagen capturada. Se evaluó la latencia en la comunicación entre PyCharm y Unity, así como entre Unity y el sistema operativo de la Raspberry Pi, obteniendo  $0.19\text{ ms}$  y  $18.16\text{ ms}$  respectivamente. Además, se aborda el desafío del control remoto del robot humanoide, permitiendo la interacción y manipulación de objetos en un entorno específico. Se identificó un error acumulativo del  $3.7\%$  en los Servomotores del brazo izquierdo debido a factores como el redondeo de ángulos y limitaciones en los movimientos. En resumen, esta investigación presenta un sistema integral que combina visión estéreo, teleoperación y control de robots humanoides, demostrando su viabilidad en un entorno de laboratorio de mecatrónica. El trabajo realizado proporciona una base sólida para futuros desarrollos en la interacción y manipulación precisas de objetos en entornos.

*Palabras Clave:* Visión Estéreo, Genius FaceCam 1000X, Robot TonyPi Pro, Raspberry Pi 4B, Teleoperación.

## Abstract

This thesis focuses on implementing a humanoid robot to remotely control, interact with and manipulate objects in the environment using Stereo Vision and Teleoperation in the Mechatronics Laboratory. The Genius FaceCam 1000X camera model is used with two identical units to maintain accurate parity and perspective at 63 mm separation between them. The camera support is designed to handle the loads applied by the cameras and operator manipulation, with a maximum displacement of 0.38 mm and a safety coefficient of 1.03 units. The system is supported by the Raspberry Pi 4B Processing Unit, which facilitates communication between the TonyPi Pro robot and external devices such as computers. Libraries have been developed to optimize the code, reduce repetitions and improve the readability of the entire program. To achieve accurate depth triangulation, a minimum distance of 111 cm is set between the camera and the reference point of the observed person, ensuring that the point of interest is within the observation range in the captured image. The communication latency between PyCharm and Unity, as well as between Unity and the Raspberry Pi operating system, was evaluated, obtaining 0.19 ms and 18.16 ms respectively. In addition, the challenge of remote control of the humanoid robot is addressed, allowing interaction and manipulation of objects in a specific environment. A cumulative error of 3.7% was identified in the Servomotors of the left arm due to factors such as angle rounding and limitations in the movements. In summary, this research presents a comprehensive system that combines stereo vision, teleoperation and humanoid robot control, demonstrating its feasibility in a mechatronics laboratory environment. The work performed provides a solid foundation for future developments in the precise interaction and manipulation of objects in environments.

*Keywords:* Stereo Vision, Genius FaceCam 1000X, TonyPi Pro Robot, Raspberry Pi 4B, Teleoperation.

## Capítulo I

### Introducción y Marco Teórico

#### Antecedentes

El ser humano ha buscado distintas maneras y herramientas que permitan aumentar el alcance de su capacidad de manipulación y la manipulación de objetos peligrosos para el ser humano. La intervención del operador humano puede ser de forma total o parcial, teniendo que controlar cada una de sus articulaciones o especificar los movimientos que se realizarán de forma automática. (Nuño Ortega & Basañez Villaluenga, 2004)

En la Universidad de Alicante, se ha desarrollado una Interfaz Humano-Máquina (HMI) mediante técnicas de visión por computador e IA; la cual debe poder ser posible teleoperar cualquier o casi cualquier tipo de brazo robot haciendo uso de gestos o poses. Para este proyecto se paralelizó procesos lanzando hilos y creando servidores TCP/IP para la transmisión de datos entre el detector de POSE y la interfaz. (Sanchis Reig, 2022)

En la Escuela Politécnica Nacional, se diseñó y simuló un sistema de teleoperación de un robot humanoide NAO; además de, realizar el modelo cinemático del mismo. En este proyecto, se implementó el método Denavit Hartenberg para la solución de la cinemática directa del robot NAO. Se implementó un HMI que enlaza el modelo 3D del robot y permite modificar los valores articulares de cada grado de libertad del robot y visualizar el movimiento que llevaría de forma física. (Nandar Carvajal, 2022)

En la Escuela Superior Politécnica de Chimborazo, se diseñó e implementó un sistema de teleoperación para controlar un robot humanoide mediante sensor Kinect. En este proyecto se utilizó una comunicación serial, siendo una manera fiable y segura para la transmisión de datos desde el computador hacia el controlador de los servomotores. (Ñacato Estrella, 2014)

En la Universidad de las Fuerzas Armadas ESPE, se ha elaborado un sistema capaz de detectar y manipular sustancias peligrosas, empleando sensores químicos, un sistema robótico teleoperado, un sistema de visión remota; mediante comandos de movimiento provenientes de

la extremidad superior para controlar la plataforma robótica. En este proyecto se caracterizó un grupo de comandos gestuales básicos, provenientes de la extremidad superior del operario para controlar el sistema robótico; estos gestos están conformados por los movimientos: flexión y extensión de los dedos, y flexión y extensión de la muñeca. También, se estableció un sistema de visión remota, que permita la navegación teleoperada del sistema robótico en entornos distantes y sin línea de vista para aislar al operario de entornos peligrosos; dónde, el mecanismo de comunicación inalámbrico Sockets, permitió crear un servidor que sirva de medio de traspaso de información entre el Servidor Python y los clientes Myo y Raspberry. (Inga Ortiz & Zoria Ortega, 2018)

También, se ha diseñado e implementado un robot móvil teleoperado con capacidades de reconocimiento de imágenes a través de Cloud Computing para la exploración de zonas distantes. En este proyecto se acopló y usó tecnologías de comunicación a larga distancia con el protocolo LoRa para establecer conectividad en el medio físico y el virtual; además, se implementó una arquitectura Cloud IoT estructurada en capas teniendo primero al robot móvil equipado con un controlador Raspberry Pi 4, sensores y geolocalización, así como los diferentes actuadores para el desplazamiento y movilidad del robot. (Espinoza Castro & Nasimba Nasimba, 2021)

### **Planteamiento del Problema**

En la Universidad de las Fuerzas Armadas ESPE Sede Latacunga, no existe alguna investigación que abarque sobre la conversión entre los grados de libertad de una persona a los de un robot humanoide, como el control remoto de un robot humanoide que permita interactuar y manipular objetos del entorno mediante visión estéreo y teleoperación.

En la institución, no existe algún programa o interfaz que capture y transforme los movimientos de una persona en código necesario para que un robot humanoide lo pueda interpretar e imitar.

La academia no cuenta con un proyecto con este tema, desconociendo los métodos y técnicas de seguimiento por visión estéreo, los métodos y técnicas para el control remoto por teleoperación del robot, y la relación que puede tener un robot humanoide y una persona entre sus grados de libertad para poder cumplir un movimiento específico.

La tecnología existe, pero no llega a abordarse, a pesar de tener a disposición herramientas que faciliten y permitan analizar, convertir y ejecutar los objetivos del proyecto. Además, no existe una relación directa del cómo replicar los movimientos de una persona a un robot de forma sencilla y eficaz.

El ser humano no puede manipular sustancias peligrosas, siendo una limitación en el campo profesional, buscando herramientas y máquinas le permitan cumplir este objetivo de forma remota y segura. Por lo anterior mencionado, se plantea proponer un robot humanoide controlado remotamente que permita interactuar y manipular objetos del entorno, mediante visión estéreo y teleoperación.

### **Justificación e Importancia**

Este proyecto busca abrir paso al conocimiento y a habilitar nuevas investigaciones a partir del documento y sistema generado. También, busca crear de forma física un sistema con un robot humanoide controlado remotamente que permita interactuar y manipular objetos del entorno, mediante Visión Estéreo y Teleoperación; dejando al Laboratorio de Mecatrónica de la Universidad de las Fuerzas Armadas ESPE Sede Latacunga, una manera de interactuar e intervenir para la generación de movimientos en el robot. La tecnología aplicada está presente en el tiempo de realización del proyecto, siendo novedoso y permitiendo la ampliación de uso de estas herramientas, métodos y técnicas que implementarán.

También, este proyecto, daría paso a la etapa inicial de entender el cómo replicar los movimientos de una persona a un robot mediante la conversión entre sus grados de libertad. El control remoto por teleoperación permitiría el control a distancia del robot sin tener que escribir

la codificación de cada articulación evitando la necesidad de que el operador manipule los objetos del entorno.

Además, puede cumplir la función de etapa inicial como proyecto investigativo para analizar las limitaciones que posee y luego puede optar por manipular objetos y sustancias peligrosas y dañinas para el ser humano; evitando que el operador entre a la necesidad de estar en contacto con el entorno.

## **Objetivos**

### ***Objetivo General***

Controlar remotamente un robot humanoide que permita interactuar y manipular objetos dentro de un entorno, mediante visión estéreo y teleoperación, en el laboratorio de Mecatrónica.

### ***Objetivos Específicos***

- Indagar sobre el estado del arte.
- Generar los conceptos básicos y necesarios dentro del proyecto.
- Diseñar un sistema de visión estéreo que proporcione imágenes y que permita visualizar los movimientos del operador.
- Implementar un robot humanoide y modificarlo para que pueda interpretar señales de control remoto.
- Convertir los grados de libertad de una persona a grados de libertad de un robot humanoide.
- Generar, en el robot, movimientos gestuales, espaciales y de interacción dentro de un entorno específico.
- Realizar pruebas de funcionamiento.
- Validar la hipótesis.

## Hipótesis de Investigación

¿Se podrá controlar remotamente un robot humanoide que permita interactuar y manipular objetos dentro de un entorno, mediante visión estéreo y teleoperación, en el laboratorio de Mecatrónica?

## Categorización de las Variables de Investigación

De hecho, se pueden identificar dos elementos de la hipótesis propuesta. Aquí están los detalles:

### ***Variable Independiente***

Robot humanoide que utiliza visión estéreo y teleoperación.

### ***Variable Dependiente***

Interacción y manipulación de los objetos dentro de un entorno.

## Marco Teórico

### ***Normas de Seguridad al Implementar un Robot Humanoide***

#### **ISO 9283 - Robots y Manipuladores Industriales. Métodos de Ensayo y**

**Presentación de los Resultados.** Esta norma especifica criterios de rendimiento y métodos de prueba asociados para robots industriales. El estándar define criterios para evaluar el rendimiento de los robots industriales en términos de repetibilidad, precisión, velocidad, aceleración, carga y rigidez. La norma también describe métodos de prueba para medir estos parámetros. (Standardization, ISO 9238: Robots and robotic devices - Industrial robots - Performance criteria and related test methods, 1998)

**ISO 9286 - Robots y Manipuladores Industriales. Controladores de Robots.** Esta norma establece los requisitos para los controladores de robots industriales. El estándar define los requisitos de interfaz, seguridad, programación, diagnóstico y comunicación para los controladores de robots industriales. La información también debe proporcionarse en los datos técnicos y manuales de usuario del controlador de datos. (Standardization, ISO 9286: Robots and robotic devices - Industrial robots - Controllers, 1997)

**ANSI/RIA R15.07-1 - Robots y Sistemas Robóticos para Espacios de Trabajo no Estructurados. Parte 1: Requisitos Generales de Seguridad.** Esta norma establece requisitos de seguridad para robots industriales y sistemas de robots. El estándar también establece requisitos para la evaluación de riesgos, la reducción de riesgos, la seguridad eléctrica, la seguridad mecánica, la seguridad de los sistemas de control y la seguridad cuando se utilizan robots. (Association, 1999)

**ISO 10218-1 - Robots y Manipuladores Industriales. Seguridad de los Robots. Parte 1: Requisitos Generales.** Esta norma define los requisitos de seguridad para los robots industriales. La norma establece requisitos para la seguridad de los robots, incluidos el diseño, la construcción y la instalación del robot, así como su mantenimiento y uso. También se imponen requisitos de seguridad adicionales en la interfaz humano-robot, incluida la distancia de separación, la velocidad y la fuerza de contacto. (Standardization, ISO 10218-1: Robots and robotic devices - Safety requirements for industrial robots - Part 1: Robots, 2011)

### ***Visión Estéreo***

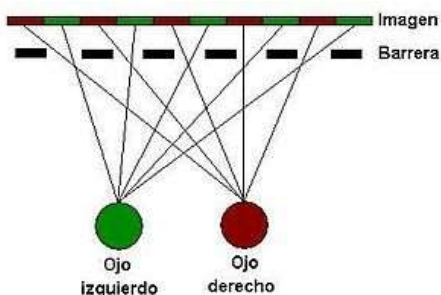
La visión estereo, conocida también como visión estereoscópica para el ser humano, es aquella que a partir de dos imágenes capturadas es capaz de integrarlas en una sola imagen para crear una matriz tridimensional con la profundidad del objeto o entorno, como se observa en la *¡Error! No se encuentra el origen de la referencia..* (Bhatti, ResearchGate, 2008)

La visión estereo, se refiere a la percepción de la estructura tridimensional del mundo. Esta toma como referencia el sistema de visión de una persona. El ser humano puede percibir la profundidad a partir de vistas ligeramente distantes dando imágenes con diferentes perspectivas. (Basulto Lantsova, 2020)



## Figura 1

*Representación gráfica de Visión Estéreo.*

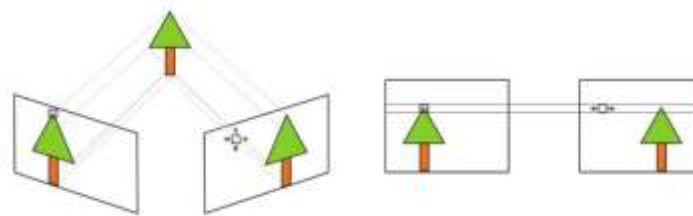


*Nota.* Tomado de “La visión estereoscópica: avance evolutivo en el espacio”. (Rodríguez Antón, 2015)

**Cámara Estéreo.** Una cámara estereo o cámara estereoscópica, puede estar compuesta por dos o más cámaras de iguales características y especificaciones técnicas. Las cámaras que lo conforman captarán imágenes a tiempo real que, al integrarlas mediante algoritmos específicos, dan una salida con la profundidad analizando la distancia entre dos puntos de interés o disparidad como se muestra en la *¡Error! No se encuentra el origen de la referencia..* (Bhatti, ResearchGate, 2008)

## Figura 2

*Disparidad en una cámara estereo.*



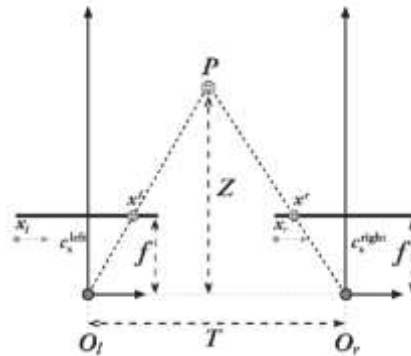
*Nota.* Tomado de “Stereo vision disparity”. (Bhatti, ReasearchGate, 2020)

**Triangulación.** La triangulación entre dos imágenes consiste en la obtención de las coordenadas X, Y y Z de un objeto a partir de puntos 2D de los píxeles de las imágenes con diferentes perspectivas. Para esto, hay que considerar el caso ideal de dos cámaras cuyas características son: sus planos de imagen son coplanarios entre sí, los ejes ópticos de las cámaras son paralelos a una distancia conocida y con distancias focales iguales como se

muestra en la *¡Error! No se encuentra el origen de la referencia..* Con estos casos ideales es posible calcular de forma fácil la profundidad de un punto usando triángulos similares. (Basulto Lantsova, 2020)

### Figura 3

*Caso ideal de un sistema estéreo.*



*Nota.* Tomado de “Emparejamiento de puntos de triángulo multicámara utilizando DIC”.

(Basulto Lantsova, 2020)

$$\frac{T - (x_l - x_r)}{Z - f} = \frac{T}{Z} \rightarrow Z = \frac{fT}{x_l - x_r} \quad (1)$$

### **Robot Humanoide**

Un robot humanoide está en la clasificación denominado robot móvil y bípedo, su diseño se basa en imitar y simular el movimiento de las articulaciones de una persona como se muestra en la *¡Error! No se encuentra el origen de la referencia..* Los movimientos más representativos en el robot son las características del torso, las dos extremidades superiores, las dos extremidades inferiores y la cabeza. Tiene como aplicación el de interactuar con las herramientas, los objetos, las personas para fines experimentales y prácticos. Algunos robots poseen la capacidad de replicar los rasgos faciales del ser humano como: los ojos, la boca y/o la frente. (Ortiz Morales, y otros, 2020)

Las investigaciones realizadas y analizadas actualmente, sobre robots humanoides, dan a luz los siguientes puntos importantes a seguir y a considerar la locomoción bípeda y estable,

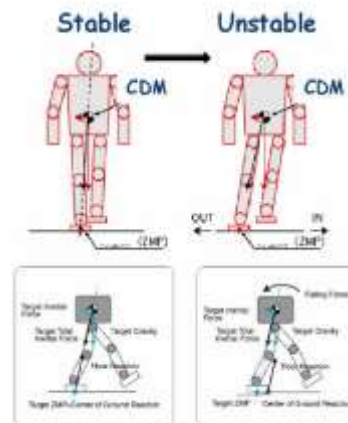
para que el robot humanoide imite caminar, no está resuelta completamente. Se pueden distinguir dos tipos de locomoción bípeda como se muestra en la *¡Error! No se encuentra el origen de la referencia.*, para poder hacer que el robot camine al igual que el ser humano:

**Activa - Estática.** Es una técnica que siempre es estable, consiste en movimientos lentos y que el centro de gravedad permanezca en las plantas de los pies, de esta forma no hay descompensación y se evita caer.

**Pasiva - Dinámica.** Es una técnica que algunas veces es inestable, consiste en movimientos más rápidos y que el centro no siempre permanece en la planta de los pies, teniendo la posibilidad de que el centro de gravedad esté en los extremos y provocando caídas inesperadas. (Falomir Llansola, 2006)

#### Figura 4

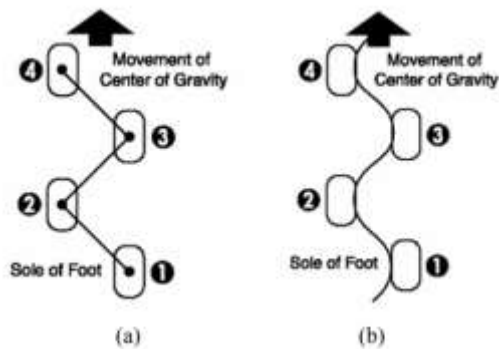
*Estabilidad, locomoción bípeda.*



*Nota.* Tomado de “Estabilidad y locomoción bípeda estable e inestable”. (Falomir Llansola, 2006)

### Figura 5

a) Activa-estable, b) Pasiva-dinámica.



*Nota.* Tomado de “Estabilidad y locomoción bípida estable e inestable”. (Falomir Llansola, 2006)

### **Marcas Comerciales de Robot Humanoide**

**TonyPi Pro de Hiwonder.** TonyPi Pro (¡Error! No se encuentra el origen de la referencia.) es un kit de desarrollo de robot humanoide basado en Raspberry Pi 4B, siendo Hiwonder la marca responsable de su desarrollo. Esta versión mejorada del kit Tony Pi de Hiwonder ofrece una variedad de conjuntos de habilidades y herramientas de IA, lo que permite a los usuarios realizar tareas más avanzadas, como evitar obstáculos, subir y bajar escaleras, recolección inteligente, reconocimiento de máscaras, etc. (Hiwonder, 2023)

## Figura 6

*TonyPi Pro Hiwonder robot humanoide, kit de desarrollo.*



*Nota.* Tomado de “TonyPi Pro”. (Hiwonder, 2023)

### **Especificaciones de TonyPi Pro.**

**Tabla 1**

*Tabla de especificaciones de los elementos del robot TonyPi Pro.*

Elementos	Especificaciones
<b>Dimensiones del robot</b>	373x186x106 mm (altura x ancho de hombros x grosor).
<b>Peso</b>	1.8 kg (sin las manos/pinzas).
<b>Material</b>	Aleación de aluminio duro.
<b>Cabeza</b>	2 grados de libertad.
<b>Hombro</b>	2 grados de libertad.
<b>Brazo</b>	2 grados de libertad.
<b>Pierna</b>	3 grados de libertad.
<b>Pie</b>	2 grados de libertad.
<b>Software</b>	Aplicación móvil + software para PC + control desde PC.
<b>Comunicación</b>	Wi-Fi y Ethernet.
<b>Forma de venta del Servomotor</b>	Venta de forma individual.
<b>Método de control</b>	Control desde PC / control desde aplicación móvil / control mediante mando (controlador).

*Nota.* Tomado de “TonyPi Pro”. (Hiwonder, 2023)

**Meccanoid G15 de Meccano.** Meccano o Erector (**Figura 7**) es la empresa encargada de la creación del kit de desarrollo conocido como robot Meccanoid G15. Este robot permite a los usuarios realizar tareas avanzadas como la construcción personalizada, programación de

sus motores, control remoto mediante conexión por computador o móvil, etc. (Meccano/Erector, 2023)

### Figura 7

*Meccanoid G15 Meccano robot humanoide, kit de desarrollo.*



*Nota.* Tomado de “Meccanoid G15”. (Meccano/Erector, 2023)

### **Especificaciones de Meccanoid G15.**

**Tabla 2**

*Tabla de especificaciones de los elementos del robot Meccanoid G15.*

Elementos	Especificaciones
<b>Dimensiones del robot</b>	450x380x89 mm (altura x ancho de hombros x grosor).
<b>Peso</b>	1.5 kg.
<b>Material</b>	Plástico y metal.
<b>Cabeza</b>	0 grados de libertad.
<b>Hombro</b>	2 grados de libertad.
<b>Brazo</b>	0 grados de libertad.
<b>Pierna</b>	2 grados de libertad.
<b>Pie</b>	0 grados de libertad.
<b>Software</b>	Aplicación móvil.
<b>Comunicación</b>	Bluetooth.
<b>Forma de venta del Servomotor</b>	Venta de forma conjunta con el robot.
<b>Método de control</b>	Control desde aplicación móvil / control mediante mando (controlador).

*Nota.* Tomado de “Meccanoid G15”. (Meccano/Erector, 2023)

**Alpha 1 Pro de Ubtech Robotics.** Ubtech Robotics (**Figura 8**) es la compañía responsable de desarrollar el robot Alpha 1 Pro. Este innovador robot humanoide brinda a los

usuarios la capacidad de ejecutar tareas sofisticadas, como la programación de sus motores y el control remoto a través de conexiones desde computadoras o dispositivos móviles, entre otras funcionalidades destacadas. (Ubtech, 2023)

### Figura 8

*Alpha 1 Pro Ubtech Robotics, robot humanoide.*



*Nota.* Tomado de “Alpha 1 Pro”. (Ubtech, 2023)

### **Especificaciones de Alpha 1 Pro.**

**Tabla 3**

*Tabla de especificaciones de los elementos del robot Alpha 1 Pro.*

<b>Elementos</b>	<b>Especificaciones</b>
<b>Dimensiones del robot</b>	406x203x127 mm (altura x ancho de hombros x grosor).
<b>Peso</b>	1.36 kg.
<b>Material</b>	Plástico.
<b>Cabeza</b>	1 grado de libertad.
<b>Hombro</b>	2 grados de libertad.
<b>Brazo</b>	2 grados de libertad.
<b>Pierna</b>	3 grados de libertad.
<b>Pie</b>	2 grados de libertad.
<b>Software</b>	Aplicación móvil + software de pc.
<b>Comunicación</b>	Bluetooth.
<b>Forma de venta del Servomotor</b>	Venta de forma conjunta con el robot.
<b>Método de control</b>	Control desde PC / control desde aplicación móvil / control mediante mando (controlador).

*Nota.* Tomado de “Alpha 1 Pro”. (Ubtech, 2023)

### **Marcas Comerciales de Controladores O Unidad de Procesamiento**

Las características de una Unidad de Procesamiento determinan el rendimiento y la capacidad de respuesta de una máquina. Entre las características a considerar se encuentran la Unidad Central de Procesamiento (CPU) y la Unidad de Procesamiento Gráfico (GPU), ambas desempeñando roles esenciales en el procesamiento de datos. Además, la cantidad y velocidad de la Memoria de Acceso Aleatorio (RAM) desempeña un papel crucial en la optimización de las operaciones. La conectividad del sistema también se vuelve esencial, ya que la capacidad de transferir datos rápidamente entre componentes y con periféricos externos influye en la eficiencia general. (intel, 2023)

**Raspberry Pi 4B.** La Raspberry Pi 4 (**Figura 9**) es una pequeña pero poderosa placa de desarrollo que es ideal para proyectos de visión artificial debido a su procesador rápido, opciones de memoria, conectividad Wi-Fi y Ethernet, y su cámara dedicada. Puedes usar librerías para detectar objetos, reconocer rostros y realizar análisis de imágenes. Aunque no es tan potente como sistemas más grandes, es asequible y versátil, siendo capaz de ejecutar tareas de aprendizaje automático y aprendizaje profundo, aunque en proyectos más complejos es necesario considerar el enfriamiento para mantener su rendimiento óptimo. (Raspberry, 2023)

#### **Figura 9**

*Dispositivo Raspberry Pi 4 Modelo B.*



*Nota.* Tomado de “Raspberry Pi 4”. (Raspberry, 2023)



**NVIDIA Jetson Nano.** Es una solución versátil y potente para aplicaciones de visión artificial y comunicación con otros dispositivos. Equipado con una GPU capaz y un procesador ARM Cortex-A57 de cuatro núcleos, el Jetson Nano (**Figura 10**) ofrece un rendimiento excepcional para tareas de procesamiento de imágenes en tiempo real, como detección de objetos, seguimiento de movimiento y reconocimiento facial. Su capacidad para ejecutar redes neuronales profundas y algoritmos de aprendizaje automático lo convierte en una elección ideal para aplicaciones de visión por computadora, como sistemas de seguridad, robótica autónoma y monitoreo inteligente. Además, el Jetson Nano facilita la comunicación con otros dispositivos a través de sus puertos GPIO, UART, I2C y SPI, permitiendo la integración con sensores, actuadores y sistemas de control externos. Esto lo convierte en un componente esencial para la creación de sistemas complejos de visión artificial conectados en red y aplicaciones de Internet de las cosas (IoT). (NVIDIA, 2023)

### **Figura 10**

*Dispositivo NVIDIA Jetson Nano.*



*Nota.* Tomado de “NVIDIA Jetson Nano”. (NVIDIA, 2023)

**BeagleBone Black.** Es una plataforma de desarrollo altamente funcional y versátil que se destaca en aplicaciones de visión artificial y comunicación con otros dispositivos. Equipada con un procesador ARM Cortex-A8 y una variedad de interfaces de conectividad, la BeagleBone Black (**Figura 11**) es ideal para implementaciones de visión por computadora en tiempo real, como detección de objetos, análisis de imágenes y procesamiento de señales visuales. Su capacidad para interactuar con una amplia gama de sensores y periféricos a

través de pines GPIO y buses I2C y SPI la hace adecuada para proyectos que requieren interacción con el entorno físico. Además, la BeagleBone Black ofrece conectividad Ethernet y USB, lo que facilita la comunicación con otros dispositivos y sistemas en red, permitiendo su integración en sistemas de control, monitoreo y automatización. Con su combinación de potencia de procesamiento y capacidades de conectividad, la BeagleBone Black se presenta como una solución eficaz para proyectos de visión artificial y aplicaciones de IoT que involucran interacción en tiempo real y comunicación con otros dispositivos. (BeagleBoard.org, 2022)

### **Figura 11**

*Dispositivo BeagleBone Black.*



*Nota.* Tomado de “BeagleBone Black”. (BeagleBoard.org, 2022)

### **Análisis de los Movimientos**

**Grados de Libertad en el Movimiento del Ser Humano.** El cuerpo humano es muy complejo, es un sistema totalmente móvil que puede dividirse en diferentes componentes segmentales. Cada segmento se conecta por una o varias articulaciones que permiten o restringen parcial o totalmente el movimiento. Un cuerpo suspendido en el aire, por conocimiento básico y macro, posee 6 grados de libertad (GDL) que son: 3 de traslación y 3 de rotación.

**Tipos de Articulaciones.** El cuerpo humano posee una variedad de tipos de articulaciones, como se observa en la **Figura 12**, que conforman y permiten la movilidad del ser humano, estas articulaciones se pueden identificar como:

- **Sinartrosis:** Son las articulaciones fibrosas e inmóviles, como el cráneo.

- **Anfiartrosis:** Son las articulaciones cartilaginosas y más o menos móviles
- **Diartrrosis:** Son las articulaciones más móviles, estas tienen una clasificación más, que son:
  - **Articulaciones de bisagra o troclear:** Estas articulaciones permiten dos tipos de movimientos de flexión y de extensión, tienen 1 grado de libertad que es de rotación.
  - **Articulaciones planas, deslizantes o artrodias:** Estas articulaciones permiten movimientos de deslizamiento, tienen 2 grados de libertad que son de traslación.
  - **Articulaciones en pivote:** Estas articulaciones permiten movimientos de rotación lateral y rotación medial, tienen 1 grado de libertad que es de rotación.
  - **Articulaciones esféricas:** Estas articulaciones permiten cualquier movimiento, sin restricciones como las anteriores, tienen 3 grados de libertad siendo dos de rotación y uno de traslación.
  - **Articulaciones en silla de montar o de encaje recíproco:** Estas articulaciones tienen 2 grados de libertad siendo uno de rotación y uno de traslación.
  - **Articulaciones condiloideas o elipsoidales:** Estas articulaciones tienen 2 grados de libertad siendo uno de rotación y uno de traslación.

## Figura 12

*Tipos de articulaciones.*



*Nota.* Tomado de “Grados de libertad en el cuerpo humano”. (Segundo Cruz, 2022)

**Cálculo de los Grados de Libertad.** Para el cálculo de la cantidad teórica de grados de libertad que conforman el cuerpo humano, se utilizan las ecuaciones (2) y (3) que son fórmulas levemente modificadas de Gruebler. El cuerpo humano posee 244 grados de libertad que permiten el movimiento y el generar actividades diarias según los resultados de la ecuación (5). (Segundo Cruz, 2022)

$$MCT = 6N - \sum_{i=3}^5 i * A_i \quad (2)$$

$$i = 6 - f \quad (3)$$

Dónde:

$MCT$  → Movilidad total corporal.

$N$  → Número total de huesos móviles.

$i$  → Clase de articulación.

$f$  → Número de GDL reales de la articulación.

$A_i$  → Número de articulaciones de clase  $i$ .

Por lo tanto:

$$MCT = 6(148) - [(3 * 29) + (4 * 33) + (5 * 85)] \quad (4)$$

$$MCT = 244 \text{ GDL} \quad (5)$$

**Grados de Libertad en el Movimiento de un Robot Humanoide.** El número de grados de libertad que un robot cuenta determina la posición y orientación de cada punto final de cada eslabón. Se pueden diferenciar tres tipos de movimientos de articulaciones: de traslación o prismática, de rotación o angular, y mixtos. La articulación tipo prismático permite al robot moverse en línea recta en una dirección específica, puede ser en uno de los ejes X, Y o Z, son de tipo deslizante. La articulación tipo angular permite al robot rotar alrededor de un eje específico, puede ser X, Y o Z, son de tipo giratorio. La articulación de movimientos mixtos puede poseer una combinación variada entre los anteriores mencionados, teniendo en una sola articulación dos o más movimientos de traslación y/o rotación. (Rodríguez, 2023)

### ***Cinemática en Robots***

La cinemática robótica es una rama esencial de la robótica que se centra en el estudio y análisis del movimiento de los robots. Su objetivo principal es comprender matemáticamente y describir el comportamiento de las articulaciones y extremidades del robot a medida que se desplaza por el espacio. Esta rama se divide en dos categorías principales: cinemática directa y cinemática inversa. (Alberto, 2014)

**Cinemática Directa.** La cinemática directa implica calcular la posición y la orientación de la punta del robot (el "brazo" o el punto final) en función de los valores conjuntos conocidos. (Alberto, 2014)

$$x = F_x(q_1, q_2, \dots, q_n) \quad (6)$$

$$y = F_y(q_1, q_2, \dots, q_n) \quad (7)$$

$$z = F_z(q_1, q_2, \dots, q_n) \quad (8)$$

$$\alpha = F_\alpha(q_1, q_2, \dots, q_n) \quad (9)$$

$$\beta = F_\beta(q_1, q_2, \dots, q_n) \quad (10)$$

$$\gamma = F_\gamma(q_1, q_2, \dots, q_n) \quad (11)$$

Dónde:

$x, y, z \rightarrow$  Valor paralelo de deslizamiento con respecto a los ejes X, Y o Z correspondientes.

$\alpha, \beta, \gamma \rightarrow$  Valor angular con respecto a los ejes X, Y o Z respectivamente.

$q \rightarrow$  Valor angular de cada articulación rotacional de un robot humanoide.

**Cinemática Inversa.** La cinemática inversa consiste en encontrar los valores de articulación necesarios para colocar la punta del robot en una posición y orientación determinadas. (Alberto, 2014)

$$q_n = F_n(x, y, z, \alpha, \beta, \gamma) \quad (12)$$

### **Conexión Remota**

La teleoperación es la unión de herramientas tecnológicas que habilitan el gobernar u operar a distancia un dispositivo, por un ser humano conocido como operador. Sistema de teleoperación es aquello que ayude y permita teleoperar un dispositivo, el cual se denomina como dispositivo teleoperado. (Nuño Ortega & Basañez Villaluenga, 2004)

**Elementos que Conforman un Sistema de Teleoperación.** Un Sistema de teleoperación consta de los siguientes elementos importantes: (Nuño Ortega & Basañez Villaluenga, 2004) (Padilla, 2018)

**Operador o Teleoperador.** Es el ser humano que controla las operaciones a distancia. Su acción puede ser continua (todo el tiempo de ejecución) o intermitente (monitorizar, identificar objetivos y planes cada cierto tiempo).

**Dispositivo teleoperado.** Este puede ser un manipulador, robot, vehículo o algún dispositivo de similares características.

**Interfaz.** Es un conjunto de elementos que ayudan y permiten la comunicación e interacción entre el operador y el sistema.

**Canal de comunicación.** Es un conjunto de elementos y dispositivos que transmiten e interpretan la señal emitida por el sistema hacia el dispositivo.

**Sensores.** Es un conjunto de dispositivos que captan la información necesaria para ser utilizada por el sistema.

**Aplicaciones.** La teleoperación es muy usada en varias aplicaciones. Uno de los principales consumidores está en la industria nuclear. Además de esta aplicación existen otros sectores como: (Nuño Ortega & Basañez Villaluenga, 2004)

**Aplicaciones en el Espacio.** Las razones por las que se usa en este sector son:

- **Seguridad:** El espacio es peligroso para el ser humano y puede ser causa de muerte de los astronautas.
- **Costo:** El equipo que usan los pasajeros humanos es muy caro y pesado a comparación del sistema teleoperado.
- **Tiempo:** El tiempo necesario para cumplir los objetivos en el espacio, son muy largos.

**Aplicaciones en la Industria Nuclear.** Una de las razones por la que se usa en este sector, es el poder tratar y manipular sustancias radioactivas, y poder moverse sin peligro en áreas contaminadas.

**Aplicaciones Submarinas.** Una de las razones por la que se usa en este sector es el de poder acceder a zonas y profundidades donde le es imposible o peligroso al ser humano.

**Aplicaciones Militares.** Una de las razones por la que se usa en este sector es el de usar vehículos aéreos no tripulados para operaciones militares en cubierto y sin arriesgar vidas humanas.

**Aplicaciones Médicas.** Una de las razones por la que se usa en este sector es el de usarlo como dispositivos de asistencia a discapacitados, desarrollo de prótesis, telecirugía y telediagnóstico.

**Otras Aplicaciones.** También son usadas para la construcción, la minería, el mantenimiento de líneas de tensión, el mantenimiento de instalaciones, la intervención de desastres naturales y entretenimiento.

### ***Aplicaciones de Programación y Comunicación para Computador***

Las aplicaciones, librerías y lenguajes de programación expuestos, son resultados de un proceso de experiencia tanto en el estudio como en afición de desarrollo de aplicaciones con respecto al fin del proyecto. Este análisis experimental garantiza la eficiencia y confiabilidad para cumplir el propósito del proyecto.

**Pycharm JetBrains.** PyCharm (**Figura 13**) es una buena aplicación por computador que trabaja bajo varios lenguajes de programación como lo es Python. Gracias a su licencia gratuita proporcionada por el correo de la institución, es posible conectarse de forma remota a un dispositivo de control conectado en la misma red que el computador que lo ejecuta. El entorno de Python en el programa puede ser el mismo del ordenador, un entorno virtual o un entorno predefinido por un dispositivo externo. También, posee un repositorio de todas las librerías disponibles para facilitar la instalación mediante botones y/o por código.

#### **Figura 13**

*Logotipo de la aplicación Pycharm Profesional.*



*Nota.* Tomado de “PyCharm”. (JetBrains, 2023)

**Python.** Python (**Figura 14**) es un lenguaje de programación de propósito general, interpretado y de alto nivel. Creado por Guido van Rossum, se publicó por primera vez en 1991. La filosofía de diseño de Python se centra en la legibilidad del código y enfatiza la simplicidad y la claridad. Python se usa en desarrollo web, ciencia de datos, inteligencia artificial, automatización, secuencias de comandos, etc. Por su simplicidad y versatilidad, Python se ha convertido en uno de los lenguajes de programación más populares y es ampliamente utilizado en la industria y la comunidad de desarrolladores. (PSF & Foundation, 2023)



## Figura 14

Logotipo del lenguaje de programación Python.



Nota. Tomado de "Python". (PSF & Foundation, 2023)

- **Tkinter:**

Tkinter es una biblioteca de interfaz gráfica de usuario (GUI) integrada en el lenguaje de programación Python. Su nombre proviene de "Tk", que es el conjunto de herramientas gráficas utilizado para implementarlo, e "inter", que hace referencia a su integración con Python. Con Tkinter, los programadores pueden crear ventanas, botones, cuadros de texto, menús y otros elementos gráficos para desarrollar aplicaciones con interfaces visuales interactivas. Tkinter proporciona una forma rápida y fácil de crear GUI en Python, lo que la convierte en una opción popular para proyectos que requieren la interacción del usuario. (Sidek, Ramli, & Saad, 2020)

- **MessageBox:**

En Tkinter, un cuadro de mensaje es un cuadro de diálogo predefinido que muestra un mensaje o pide confirmación al usuario de forma sencilla. Es una forma rápida y fácil de mostrar información importante o interactuar con el usuario en aplicaciones GUI. Se puede utilizar las diferentes funciones del módulo de cuadro de mensaje para mostrar diferentes tipos de cuadros de diálogo. Algunas características comunes incluyen: (Python, 2023)

- **Información.** - Muestra un cuadro de diálogo de información con un título y un mensaje, como se puede observar en la **Figura 15**.

```
from tkinter import messagebox
tkinter.messagebox.showinfo(title=None, message=None, **options)
```

### Figura 15

*Caja o ventana de mensaje para mostrar información.*

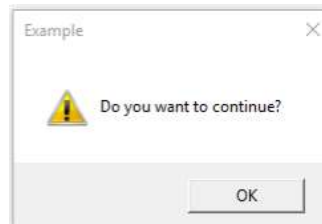


- **Advertencia y error.** - Muestra un cuadro de diálogo de alerta con un título y un mensaje, como se puede observar en la **Figura 16** y la **Figura 17**.

```
from tkinter import messagebox
tkinter.messagebox.showwarning(title=None, message=None, **options)
tkinter.messagebox.showerror(title=None, message=None, **options)
```

### Figura 16

*Caja o ventana de mensaje para mostrar una advertencia.*



### Figura 17

*Caja o ventana de mensaje para mostrar un error.*



- **Preguntas.** - Muestra un cuadro de diálogo con un título y una pregunta, ofrece botones dando una salida dependiente de la respuesta, como se puede observar en la **Figura 18**.

```

from tkinter import messagebox
tkinter.messagebox.askquestion(title=None, message=None, **options)
tkinter.messagebox.askokcancel(title=None, message=None, **options)
tkinter.messagebox.askretrycancel(title=None, message=None, **options)
tkinter.messagebox.askyesno(title=None, message=None, **options)
tkinter.messagebox.askyesnocancel(title=None, message=None, **options)

```

**Figura 18**

*Caja o ventana de mensaje para mostrar una pregunta.*



- **FileDialog:**

El módulo de diálogo de archivos de Tkinter proporciona una manera fácil de crear cuadros de diálogo para seleccionar archivos o carpetas en aplicaciones GUI de Python. Permite a los usuarios navegar por su sistema de archivos y seleccionar una ubicación o archivo específico. El módulo de diálogo de archivo tiene varias funciones que le permiten mostrar diferentes tipos de cuadros de diálogo, algunos de los más comunes son: (Python, 2023)

Para abrir un archivo:

```

from tkinter import filedialog
tkinter.filedialog.askopenfile(mode='r', **options)

```

Para guardar un archivo:

```

from tkinter import filedialog
tkinter.filedialog.asksaveasfile(mode='w', **options)

```

Para preguntar la dirección de la carpeta:

```

from tkinter import filedialog
tkinter.filedialog.askopenfilename(**options)

```

- **Custom Tkinter:**

Es una biblioteca de Python dedicada basada en Tkinter, una biblioteca de interfaz de usuario popular para crear aplicaciones de escritorio. La característica especial de CustomTkinter (**Figura 19**) es su capacidad para dar a los widgets un diseño moderno y atractivo, como botones, cuadros de texto, barras de estado y otros elementos de la GUI. La principal ventaja de CustomTkinter es que proporciona una apariencia uniforme en todas las principales plataformas de escritorio, como Windows, macOS y Linux. Esto es esencial para garantizar una experiencia de usuario uniforme, independientemente del sistema operativo en el que se ejecute la aplicación. (Chimansky, 2023)

### Figura 19

*Logotipo de la Biblioteca personalizable para Tkinter, CustomTkinter.*



*Nota.* Tomado de “Custom TKinter”. (Chimansky, 2023)

Para el tema a usar, existen 3 predefinidos que son: “blue” (azul), “green” (verde), “dark-blue” (azul oscuro). Con el siguiente código se define el tema de las ventanas a utilizar en el programa, caso contrario por defecto estará en azul, o para un tema propio se puede direccionar al archivo que contiene la información.

```
customtkinter.set_default_color_theme("dark-blue")
customtkinter.set_default_color_theme("path/to/your/custom_theme.json")
```

Para la apariencia de la ventana, existen 3 modos que son: “system” (sistema), “dark” (oscuro), “light” (claro). Con el siguiente código se puede definir, caso contrario por defecto estará cambiando según el sistema operativo entre claro y oscuro.

```
customtkinter.set_appearance_mode("system")
```

Algunas de las herramientas que posee de forma personalizada, son:

- **Top Level:**

Aparte de la ventana principal de la aplicación, es una extra con las mismas características. Para poder iniciarla se necesita de:

```
toplevel = CTkToplevel(app)
```

- **Button:**

Para poder colocar un botón en la ventana e iniciarla es necesario de su código y de una función que se activará cuando este botón sea presionado.

```
def button_event():
    print("button pressed")
button = customtkinter.CTkButton(app, text="CTkButton", command=button_event)
```

- **Label:**

Para poder colocar un texto, no editable por teclado, en la ventana e iniciarla es necesario de su código.

```
label = customtkinter.CTkLabel(app, text="CTkLabel", fg_color="transparent")
```

- **Combo Box:**

Para poder colocar un cuadro de lista desplegable en la ventana e iniciarla es necesario de su código y de una función que te permite acceder al valor por cada cambio. (Chimansky, 2023)

```
def combobox_callback(choice):
    print("combobox dropdown clicked:", choice)
combobox = customtkinter.CTkComboBox(app, values=["option 1", "option 2"],
command=combobox_callback)
```

- **OpenCV:**

OpenCV (Open Source Computer Vision Library) es una biblioteca de código abierto especializada en visión artificial y procesamiento de imágenes. Originalmente lo desarrolló Intel y luego lo mantuvo la comunidad de código abierto. OpenCV (**Figura 20**) está escrito en C++ e interactúa con los lenguajes de programación más populares, incluidos Python, Java y C++. Es ampliamente utilizado en diversas aplicaciones, como visión artificial, robótica, automatización industrial, aplicaciones médicas, monitoreo, etc. Su versatilidad, eficiencia y ricas características lo convierten en una poderosa herramienta para aquellos que trabajan en visión por computadora y procesamiento de imágenes. (OpenCV, 2023)

### **Figura 20**

*Logotipo de la biblioteca de código abierto OpenCV.*



*Nota.* Tomado de "OpenCV". (Atom, 2018)

- **CVZone:**

CVZone (Computer Vision Zone) es una poderosa herramienta para el trabajo relacionado con la visión artificial. Con CVZone (**Figura 21**), los usuarios pueden realizar tareas de procesamiento de imágenes y funciones de inteligencia artificial de manera fácil y eficiente. Ampliamente reconocida y utilizada en visión artificial, la biblioteca OpenCV proporciona un conjunto completo de herramientas para el procesamiento de imágenes, la detección de objetos, el seguimiento de objetos y otras funciones avanzadas. Por otro lado, la biblioteca Mediapipe es conocida por su enfoque en el seguimiento de manos, rostros y poses en tiempo real y la detección de características especiales en imágenes y videos. La combinación de estas dos bibliotecas en un paquete de software brinda a los desarrolladores una amplia gama

de funciones para procesar imágenes y ejecutar algoritmos de inteligencia artificial. Al simplificar y agilizar las muchas tareas complejas asociadas con la visión artificial, la herramienta permite a los usuarios centrarse en el desarrollo de soluciones innovadoras de visión artificial e inteligencia artificial. (Zone, 2023)

### **Figura 21**

*Logotipo para la biblioteca de importación CVZone.*



*Nota.* Tomado de “CVZone”. (Zone, 2023)

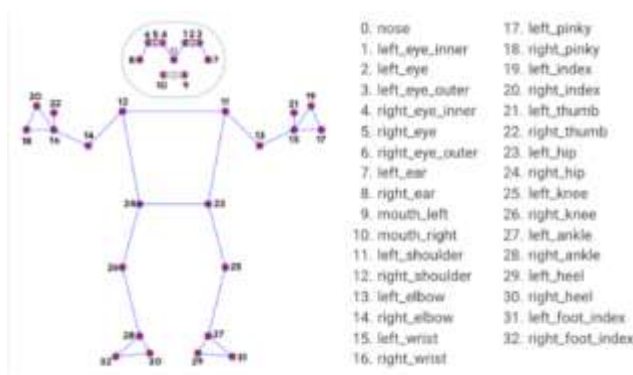
- **POSE Detector:**

CVZone POSE Detector es parte del paquete de software CVZone que se enfoca específicamente en detectar y rastrear poses humanas en imágenes o videos. Es una herramienta valiosa para desarrolladores y entusiastas de la visión por computadora que desean realizar tareas relacionadas con la estimación de la POSE humana. La herramienta utiliza técnicas avanzadas de aprendizaje profundo y visión por computadora para identificar y rastrear las articulaciones humanas en tiempo real como se puede observar en la **Figura 22**. Usando estos métodos, el detector de poses CVZone puede proporcionar resultados precisos y confiables en la detección de poses humanas. Entre las características comunes de un detector de POSE como CVZone se encuentran: (CVZone, 2023)

- Detección de articulaciones.
- Estimación de posturas.
- Seguimiento en tiempo real.
- Integración con otras herramientas.

**Figura 22**

33 puntos de referencia de CVZone con sus aristas.



Nota. Tomado de "CVZone". (CVZone, 2023)

- **Socket:**

Los sockets son la herramienta básica para la comunicación entre diferentes procesos de red. La API de sockets es una interfaz de programación que permite a los desarrolladores interactuar con sockets para enviar y recibir datos a través de una red. Los sockets se utilizan para permitir la comunicación entre aplicaciones y dispositivos en una red, ya sea una red de área local o una red más grande como Internet. Cuando dos dispositivos necesitan intercambiar información, cada uno crea un socket y se conecta a la red mediante un protocolo específico, como TCP (Protocolo de control de transmisión) o UDP (Protocolo de datagramas de usuario). (Foundation, 2023) (Hunt, 2019)

**Unity.** Es una poderosa plataforma de desarrollo de juegos y experiencia 3D interactiva popular en la industria del desarrollo de juegos, así como en aplicaciones de realidad virtual y realidad aumentada. Lo creó Unity Technologies (**Figura 23**) y lo lanzó originalmente en 2005. La plataforma tiene una amplia comunidad de desarrolladores por sus características y facilidad de uso. Se ha utilizado para crear una variedad de juegos y experiencias interactivas en múltiples plataformas, incluidas PC, consolas de juegos, dispositivos móviles y realidad virtual.

Algunas de las características clave que hacen que Unity se destaque es su potente motor de juego que facilita la creación de juegos en 2D y 3D. Sus capacidades multiplataforma



también son muy apreciadas, lo que permite a los desarrolladores impulsar su trabajo en todos los dispositivos y sistemas operativos. Otra ventaja es su editor visual intuitivo, que te permite diseñar y animar fácilmente escenas y personajes usando arrastrar y soltar. Además, se puede programar en C# o JavaScript/UnityScript para ampliar la funcionalidad de los juegos y aplicaciones, dando a los desarrolladores la opción de su lenguaje de programación preferido. (Technologies, 2023)

### **Figura 23**

*Logotipo de la marca y software Unity.*



*Nota.* Tomado de "Unity". (Technologies, 2023)

**Microsoft Visual Studio.** La combinación de las dos herramientas brinda a los desarrolladores una experiencia de desarrollo poderosa y eficiente que les permite crear juegos y experiencias interactivas de alta calidad. La capacidad de depurar y editar en tiempo real junto con los asistentes de Unity y las potentes herramientas de Visual Studio (**Figura 24**) hacen que esta combinación sea muy valorada por la comunidad de desarrolladores de Unity. Con esta integración, los desarrolladores pueden crear fácilmente proyectos más complejos y optimizados, brindando una experiencia de desarrollo más eficiente y satisfactoria. (Microsoft, 2022)

### Figura 24

*Logotipo de la marca y software Visual Studio.*



*Nota.* Tomado de "Visual Studio". (Microsoft, 2022)

**Lenguaje de Programación C#.** C# (pronunciado "C sharp") es el lenguaje de programación de Unity preferido y ampliamente utilizado debido a sus muchas ventajas en términos de rendimiento, eficiencia y herramientas de desarrollo avanzadas. Su clara integración con el motor de Unity, la amplia comunidad y la disponibilidad de recursos, y la compatibilidad con funciones del motor hacen de C# (**Figura 25**) sea la opción más popular para los desarrolladores de proyectos de Unity. Al aprender C#, los desarrolladores pueden usar toda la potencia del motor para crear juegos emocionantes y experiencias interactivas con mayor eficiencia y rendimiento. (GUÉRIN, 2018)

### Figura 25

*Logotipo del lenguaje de programación C#.*



*Nota.* Tomado de "C#". (GUÉRIN, 2018)

**VNC Viewer.** Es una aplicación que permite a un usuario acceder y controlar de forma remota una computadora o dispositivo desde otro dispositivo. VNC (**Figura 26**), abreviatura de "computación de red virtual", es un protocolo que permite compartir la pantalla de forma remota y controlar las computadoras. Un visor VNC se usa junto con un servidor VNC que se ejecuta

en el dispositivo que desea controlar de forma remota. El servidor VNC envía capturas de pantalla y recibe eventos de teclado y mouse de dispositivos externos. Por otro lado, el cliente del visor VNC recibe estas imágenes y envía eventos de teclado y mouse al servidor, lo que permite al usuario interactuar con el dispositivo remoto como si estuviera frente a él. (Ariyanto, 2015)

### **Figura 26**

*Logotipo del programa VNC Viewer.*



*Nota.* Tomado de “VNC Viewer”. (Ariyanto, 2015)

### ***Puntos para el Análisis de Resultados***

En un robot humanoide controlado de forma remota, el objeto se puede manipular mediante sensores y sistemas de visión artificial. La manipulación se realiza mediante presión lateral al objeto a manipular mediante los brazos robóticos, actuando como pinzas de agarre, utilizando un enfoque basado en el destino del punto final de cada brazo y utilizando la predicción manual a la ubicación del objeto manipulado y destino.

Una de las pruebas que se pueden aplicar en un robot humanoide es el analizar la precisión en la cinemática para el seguimiento entre las posiciones propuestas por el análisis por visión artificial y el resultado obtenido por el giro de los Servomotores. (Y. Liu, 2021)

## Capítulo II

### Selección Y Configuración Inicial.

#### Selección

##### *Selección del Dispositivo de Cámara*

Los dispositivos de cámara a utilizar para obtener una Visión Estéreo deben tener características idénticas o similares. Por lo tanto, solo se elegirá una cámara, y la otra deberá ser del mismo fabricante y modelo según la empresa. Se tomó en cuenta los siguientes dispositivos de cámara:

**Cámara 1 (C1): Genius FaceCam 1000X.** La primera cámara analizada (**Figura 27**) posee las siguientes características:

**Tabla 4**

*Características de la cámara Genius FaceCam 1000X.*

<b>Elementos</b>	<b>Especificaciones</b>
<b>Dimensiones</b>	20x24x60 mm (altura x profundidad x ancho)
<b>Resolución</b>	1280x720 o 640x480 píxeles
<b>Cuadros por segundo</b>	30 FPS
<b>Soporte</b>	Pantallas LCD, Laptop, sobre mesa
<b>Ángulo de visión</b>	60°
<b>Balance de blancos</b>	Automático y manual
<b>Enfoque</b>	Automático y manual
<b>Peso</b>	50g
<b>Precio</b>	\$25.00 aproximadamente

*Nota.* Tomado de "Genius FaceCam 1000X". (Inicio, 2023)

**Figura 27***Genius FaceCam 1000X.*

*Nota.* Tomado de “Genius FaceCam 1000X”. (Vinicio, 2023)

**Cámara 2 (C2): Logitech C920e.** La segunda cámara analizada (**Figura 28**) posee las siguientes características:

**Tabla 5**

*Características de la cámara Logitech C920e V-U0028.*

<b>Elementos</b>	<b>Especificaciones</b>
<b>Dimensiones</b>	43x71x94 mm (altura x profundidad x ancho)
<b>Resolución</b>	1920x1080 píxeles
<b>Cuadros por segundo</b>	30 FPS
<b>Soporte</b>	Pantallas LCD, Laptop, sobre mesa
<b>Ángulo de visión</b>	78°
<b>Balance de blancos</b>	Automático
<b>Enfoque</b>	Automático
<b>Peso</b>	162g
<b>Precio</b>	\$100.00 aproximadamente

*Nota.* Tomado de “Logitech C920e”. (Andrezero, 2023)

**Figura 28**

*Logitech C920e V-U0028.*



*Nota.* Tomado de “Logitech C920e”. (Andrezero, 2023)

**Cámara 3 (C3): SPEEDMIND HD 1080P.** La tercera cámara analizada (**Figura 29**)

posee las siguientes características:

**Tabla 6**

*Características de la cámara SPEEDMIND HD 1080P.*

<b>Elementos</b>	<b>Especificaciones</b>
<b>Dimensiones</b>	39x80x118 mm (altura x profundidad x ancho)
<b>Resolución</b>	1920x1080 píxeles
<b>Cuadros por segundo</b>	45 FPS
<b>Soporte</b>	Sobre mesa
<b>Ángulo de visión</b>	60°
<b>Balance de blancos</b>	Automático
<b>Enfoque</b>	Automático
<b>Peso</b>	170g
<b>Precio</b>	\$30.00 aproximadamente

*Nota.* Tomado de “SPEEDMIND HD 1080P”. (Electrónica, 2023)

**Figura 29**

*SPEEDMIND HD 1080P.*



*Nota.* Tomado de "SPEEDMIND HD 1080P". (Electrónica, 2023)

**Comparación de Características.** Entre las características expuestas, para el proyecto; es fundamental tener en cuenta de forma primordial las dimensiones de las cámaras ya que de esta depende la distancia de separación a la que se encontrarían y formaría parte para determinar y triangular la profundidad de un objeto o punto de análisis en la imagen; siendo lo menos importante el número de herramientas para realizar un balance de blancos ya que en la imagen no perjudica en el análisis de Visión Artificial, pero la podría mejorar. Según el método de los factores ponderados, las características se puntuarán en orden desde la que menos afecta al proyecto hasta un total de 1 que resulta el 100% de la forma en que afectaría al proyecto. El número de herramientas para realizar un balance de blancos dependiente de la iluminación, haciendo que beneficie o perjudique según el caso, como el número de superficies u objetos compatibles el soporte ya incluido, ya que, aunque servirían de sostén para las cámaras de forma individual, pueden adaptarse, son la menor importancia por la poca influencia que poseen en afectar el resultado del proyecto. Las dimensiones y resolución de la cámara poseen la mayor importancia ya que de esta depende el diseño del soporte y la distancia entre las cámaras para poder triangular y obtener la profundidad de un punto en las imágenes.

Cada cámara es evaluada del 1 al 3 siendo 3 la característica que favorezca más al proyecto y 1 la característica que menos favorezca. El total de ponderación para cada cámara está dado por:

$$Total = \sum_{i=1}^{i=n} C_i * W_i \quad (13)$$

Dónde:

$i \rightarrow$  Es el número de fila que corresponde la característica.

$C_i \rightarrow$  Es el valor de ponderación (1 a 3) que se encuentra en la columna de la cámara evaluada y la fila  $i$ .

$W_i \rightarrow$  Es el valor del peso de la característica evaluada en la fila  $i$ .

**Tabla 7**

Comparación entre características según su valor de importancia para el proyecto.

Características	Cámaras				
	Nombre	Peso	C1	C2	C3
<b>Dimensiones</b>		0.2	3	2	1
<b>Resolución</b>		0.2	1	3	3
<b>Cuadros por segundo</b>		0.1	1	1	3
<b>Soporte</b>		0.025	2	3	1
<b>Ángulo de visión</b>		0.1	1	3	1
<b>Balance de blancos</b>		0.025	3	1	1
<b>Enfoque</b>		0.15	3	1	1
<b>Peso</b>		0.15	3	2	1
<b>Precio</b>		0.05	3	1	2
<b>Total</b>		<b>1</b>	<b>2.18</b>	<b>2.00</b>	<b>1.65</b>

La **Tabla 7** muestra que la primera cámara de modelo Genius FaceCam 1000X con un valor de 2.18, del 3 ideal para características con ponderación máxima, presenta mejores propiedades relación costo-beneficio para obtener el mejor resultado para un dispositivo de cámaras estéreo. Por lo tanto, ambas cámaras deben cumplir estas mismas características para que no existan problemas al triangular la profundidad.



### **Selección del Material para el Soporte de Cámara Estéreo**

Para que las cámaras estén a una distancia y no se muevan o roten entre sí es necesario de un soporte que los inmovilice entre las cámaras.

**Material 1 (M1): Filamento Termoplástico ABS.** El primer material para el soporte de las cámaras (**Figura 30**) posee las siguientes características:

**Tabla 8**

*Características del filamento termoplástico ABS.*

Elementos	Especificaciones
<b>Módulo de elasticidad</b>	2.1 a 2.7 GPa
<b>Dureza</b>	70 a 105 Shore D.
<b>Tolerancia a la temperatura</b>	80 a 85 °C
<b>Facilidad de procesamiento</b>	Se puede imprimir en impresoras 3D comunes, pero puede requerir una cama caliente debido a su tendencia a deformarse.
<b>Densidad</b>	1.04 a 1.07 g/cm <sup>3</sup>

*Nota.* Tomado de “Filamento ABS”. (Tractus3D, Filamento ABS, 2020)

**Figura 30**

*León impreso 3D con filamento ABS.*



*Nota.* Tomado de “Filamento ABS”. (Tractus3D, Filamento ABS, 2020)

**Material 2 (M2): Filamento de Material Termoplástico PLA.** El segundo material para el soporte de las cámaras (**Figura 31**) posee las siguientes características:

**Tabla 9**

*Características del filamento de material termoplástico PLA.*

<b>Elementos</b>	<b>Especificaciones</b>
<b>Resistencia</b>	3.5 a 7.5 GPa
<b>Dureza</b>	70 a 85 Shore D.
<b>Tolerancia a la temperatura</b>	60 a 70 °C
<b>Facilidad de procesamiento</b>	Fácil de imprimir en impresoras 3D, no requiere cama caliente y emite menos olores durante la impresión.
<b>Densidad</b>	1.24 a 1.27 g/cm <sup>3</sup>

*Nota.* Tomado de “Filamento PLA”. (Tractus3D, Filamento PLA, 2020)

**Figura 31**

*León impreso 3D con filamento PLA.*



*Nota.* Tomado de “Filamento PLA”. (Tractus3D, Filamento PLA, 2020)

**Material 3 (M3): Madera.** El tercer material para el soporte de las cámaras (**Figura 32**) posee las siguientes características:

**Tabla 10***Características de la madera.*

	<b>Especificaciones</b>
<b>Resistencia</b>	1.5 a 5 GPa
<b>Dureza</b>	70 Shore D o menor
<b>Tolerancia a la temperatura</b>	150 a 180 °C
<b>Facilidad de procesamiento</b>	Puede imprimirse en impresoras 3D, pero puede requerir ajustes debido a la presencia de partículas de madera, o por manufactura.
<b>Densidad</b>	1.0 a 1.2 g/cm <sup>3</sup>

*Nota.* Tomado de “Madera”. (Yudi, 2023)

**Figura 32***León esculpido en madera.*

*Nota.* Tomado de “Madera”. (Yudi, 2023)

**Comparación de Características.** Para el proyecto es fundamental considerar de forma primordial la densidad del material ya que de esta depende el peso del soporte y estabilidad; siendo lo menos importante la tolerancia de la temperatura, ya que la temperatura emitida por el ambiente y los propios dispositivos de cámara no afectaría en el resultado para el proyecto, pero la podría mejorar.

**Tabla 11**

*Comparación entre características según su valor de importancia para el proyecto.*

Características	Materiales				
	Nombre	Peso (Wi)	M1	M2	M3
<b>Resistencia</b>		0.25	3	1	2
<b>Dureza</b>		0.1	3	2	1
<b>Tolerancia a la temperatura</b>		0.1	2	1	3
<b>Facilidad de procesamiento</b>		0.25	2	3	1
<b>Densidad</b>		0.3	3	2	3
<b>Total</b>		<b>1</b>	<b>2.65</b>	<b>1.90</b>	<b>2.05</b>

La **Tabla 11** muestra que el filamento de material ABS con un valor de 2.65, del 3 ideal para características con ponderación máxima, presenta mejores propiedades relación costo-beneficio para obtener el mejor resultado en el desarrollo de un soporte para las cámaras ya seleccionadas.

### **Selección del Robot Humanoide**

Teniendo en cuenta la **Tabla 1**, la **Tabla 2** y la **Tabla 3** expuestas anteriormente, se puede realizar la siguiente comparación buscando características favorables al proyecto:

**Tabla 12**

*Comparación entre características según su valor de importancia para el proyecto.*

Características	Robot Humanoide (RH)				
	Nombre	Peso	RH1	RH2	RH3
<b>Dimensión</b>		0.05	1	3	2
<b>Peso</b>		0.1	1	2	3
<b>Resistencia del material</b>		0.05	3	2	1
<b>Grados de libertad</b>		0.2	3	1	3
<b>Plataformas soportadas</b>		0.25	3	1	2
<b>Métodos de comunicación</b>		0.1	3	1	2
<b>Venta de repuestos</b>		0.15	3	1	1
<b>Precio</b>		0.1	2	3	1
<b>Total</b>		<b>1</b>	<b>2.60</b>	<b>1.45</b>	<b>2.00</b>

La **Tabla 12** muestra que el robot humanoide TonyPi Pro de la marca Hiwonder con un valor del 2.6, del 3 ideal para características con ponderación máxima, presenta mejores

características relación costo-beneficio para obtener el mejor resultado en la programación, comunicación y movilidad con respecto al beneficio del proyecto.

### **Selección del Controlador o Unidad de Procesamiento**

Teniendo en cuenta lo ya seleccionado con el robot humanoide en la **Tabla 12**, se selecciona el controlador o unidad de procesamiento para lograr una comunicación remota, entre las cuales están:

**Unidad de Procesamiento 1 (UP1): Raspberry Pi 4B.** El primer controlador, que se puede observar en la **Figura 9**, posee las siguientes características:

**Tabla 13**

*Características de la unidad de procesamiento Raspberry Pi 4B.*

<b>Elementos</b>	<b>Especificaciones</b>
<b>CPU</b>	Broadcom BCM2711, Quad-core Cortex-A72 a 1.5 GHz.
<b>GPU</b>	VideoCore VI con soporte para OpenGL ES 3.x.
<b>RAM</b>	Opciones de 2 GB, 4 GB y 8 GB de LPDDR4-3200.
<b>Conectividad</b>	Gigabit Ethernet, Wi-Fi 802.11ac, Bluetooth 5.0.
<b>Aplicaciones</b>	Programación, servidores ligeros, proyectos multimedia, emulación, IoT.
<b>Precio</b>	\$190.00 aproximadamente

*Nota.* Tomado de “Características del Raspberry Pi 4B”. (GeekPi, 2023)

**Unidad de Procesamiento 2 (UP2): NVIDIA Jetson Nano.** El segundo controlador, que se puede observar en la **Figura 10**, posee las siguientes características:

**Tabla 14**

*Características de la unidad de procesamiento NVIDIA Jetson Nano.*

Elementos	Especificaciones
<b>CPU</b>	ARM Cortex-A57 quad-core a 1.43 GHz.
<b>GPU</b>	GPU Maxwell de 128 núcleos con capacidad para CUDA y aceleración de AI.
<b>RAM</b>	4 GB de LPDDR4.
<b>Conectividad</b>	Gigabit Ethernet, Wi-Fi y Bluetooth (dependiendo del modelo).
<b>Aplicaciones</b>	Proyectos de AI, computación de borde, visión por computadora, robótica.
<b>Precio</b>	\$350.00 aproximadamente

*Nota.* Tomado de “Características del NVIDIA Jetson Nano”. (Makeronics, 2023)

**Unidad de Procesamiento 3 (UP3): Beaglebone Black.** El tercer controlador, que se puede observar en la **Figura 11**, posee las siguientes características:

**Tabla 15**

*Características de la unidad de procesamiento Beaglebone Black.*

Elementos	Especificaciones
<b>CPU</b>	AM335x ARM Cortex-A8 single-core a 1 GHz.
<b>GPU</b>	PowerVR SGX530.
<b>RAM</b>	512 MB de DDR3.
<b>Conectividad</b>	10/100 Ethernet, USB 2.0, Wi-Fi mediante Tomadores USB.
<b>Aplicaciones</b>	Proyectos de robótica, automatización, control en tiempo real.
<b>Precio</b>	\$90.00 aproximadamente

*Nota.* Tomado de “Características del Beaglebone Black”. (Components, 2023)

**Comparación de Características.** Entre las características expuestas, para el proyecto; es fundamental tener en cuenta de forma primordial si el controlador ya viene incluido en el robot humanoide seleccionado o no ya que de esta depende la complejidad del proyecto; siendo lo menos importante el precio del controlador ya que de esta no depende el rendimiento del proyecto, pero podría facilitar la accesibilidad económica del operador.

**Tabla 16**

*Comparación entre características según su valor de importancia para el proyecto.*

Características	Unidad de Procesamiento				
	Nombre	Peso (Wi)	UP1	UP2	UP3
<b>CPU</b>		0.1	3	2	1
<b>GPU</b>		0.1	2	3	1
<b>RAM</b>		0.1	3	2	1
<b>Conectividad</b>		0.2	3	3	1
<b>Aplicaciones</b>		0.15	2	1	3
<b>Incluido en el robot</b>		0.3	3	1	1
<b>Precio</b>		0.05	2	3	1
<b>Total</b>		<b>1</b>	<b>2.70</b>	<b>1.90</b>	<b>1.30</b>

La **Tabla 16** muestra que la unidad de procesamiento Raspberry Pi 4B con un valor de 2.7, del 3 ideal para características con ponderación máxima, presenta mejores características relación costo-beneficio para obtener el mejor resultado para un buen manejo del Robot ya seleccionado.

## **Configuración Inicial**

### **Conexión con el Software PyCharm**

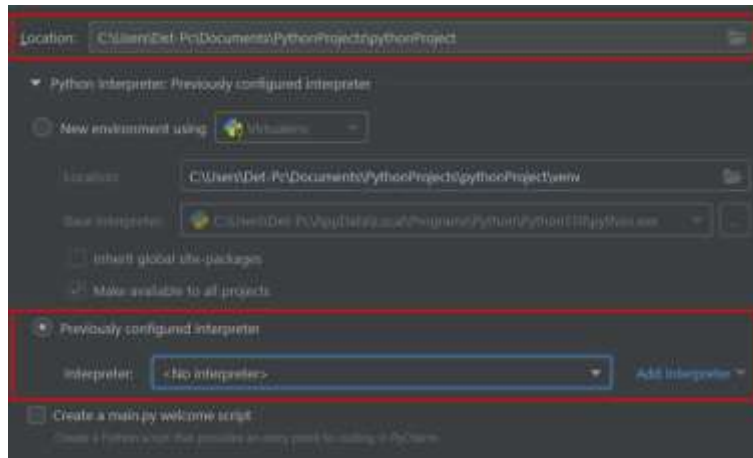
Hay que verificar que el robot esté encendido y que la computadora donde se realizará la comunicación se encuentre en la misma red creada por el mismo robot, estos pasos se encuentran en **Anexo 2**. Para una correcta conexión remota de PyCharm con el robot es necesario seguir los siguientes pasos:

**Paso 1.** Crear un nuevo proyecto desde cero como se muestra en la **Figura 33**, pero a diferencia de crear un entorno virtual en este caso se utilizará el entorno del mismo Raspberry. *Location: Dirección* (Dirección de la computadora donde se guardará el proyecto, como copia de inicialización del programa)

*Previously configured interpreter: Activado* (Es para activar el uso de un entorno ya creado anteriormente)

**Figura 33**

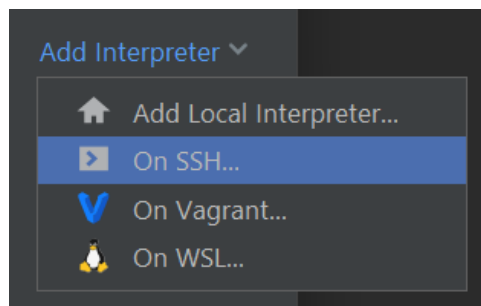
Ventana emergente para configurar el proyecto.



**Paso 2.** En la misma ventana se selecciona la opción “Add Interpreter” para volver a seleccionar “On SSH” y comunicarse de una forma similar a la de VNC Viewer al entorno del Robot como se observa en la **Figura 34**.

**Figura 34**

Opción para el entorno por comunicación SSH.



**Paso 3.** En la nueva ventana emergente como se observa en la **Figura 35**, se ingresan los siguientes datos:

*Host: raspberrypi.local* (dirección general independiente del cambio de la red y dirección IP)

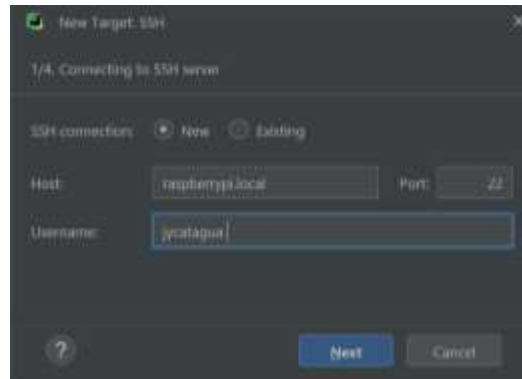
*Username: jycatagua* (por defecto es “pi” el usuario de conexión)

*Port: 22* (por defecto está ese valor en PyCharm)



### Figura 35

Ventana de configuración para la comunicación con el entorno.

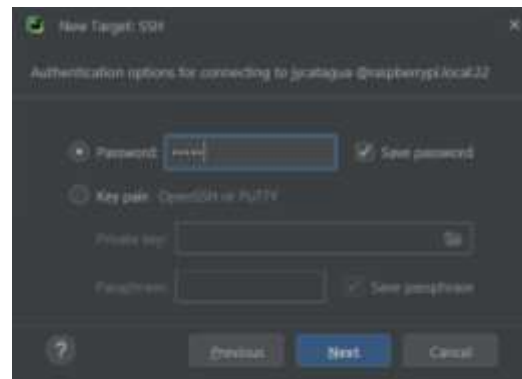


**Paso 4.** Para la siguiente etapa se ingresarán los siguientes datos como se observa en la **Figura 36**:

*Password: espel* (por defecto, la contraseña es “raspberry”)

### Figura 36

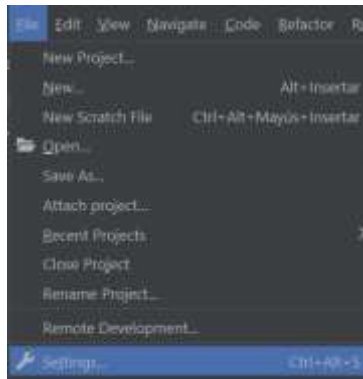
Ventana de configuración de acceso y seguridad.



**Paso 5.** Una vez creado el entorno, se necesita configurar la dirección de codificación en el Robot como se puede observar en la **Figura 37**, para esto se ingresa a “File” y a “Settings...” o usando el atajo Ctrl+Alt+S.

### Figura 37

Opción de configuración general del proyecto en PyChram.



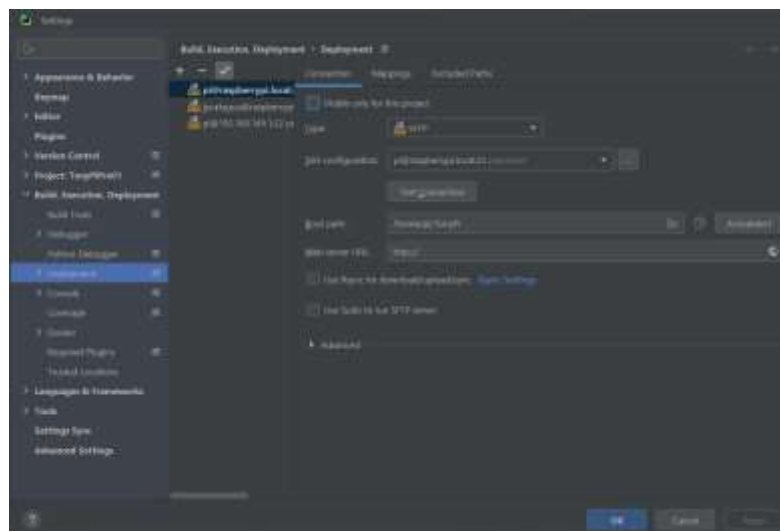
**Paso 6.** D de configuración, buscar y desplegar la opción “Build, Execution, Deployment” y “Deployment”. Dentro de esta opción se edita la comunicación con los siguientes valores:

Dentro de “Connection” (**Figura 38**): *Root path: /home/pi/TonyPi* (Puede variar según la configuración del usuario de la Raspberry)

Dentro de “Mappings” (**Figura 39**): *Deployment Path: /Example* (Carpeta que se actualizará y editará para codificar)

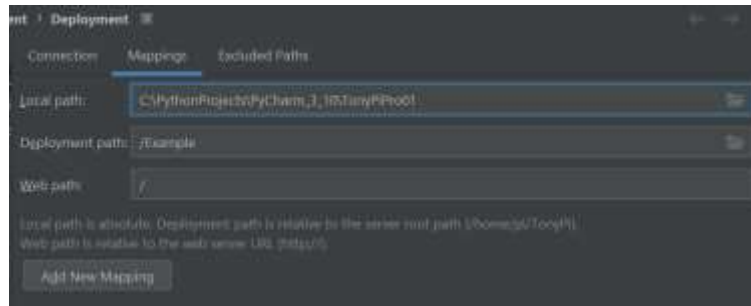
### Figura 38

Configuración Deployment Connection para la codificación.



**Figura 39**

*Configuración Deployment Mappings para la codificación.*



Terminando los pasos anteriores ya es posible crear carpetas, archivos de código como de anexos. Para verificar que se han realizado bien los pasos, se puede crear un archivo de extensión “.txt” que contenga el mensaje “conexión correcta” y no debería saltar ningún error en la consola de PyCharm, caso contrario se puede crear un nuevo proyecto y comenzar desde el primer paso de la comunicación entre PyCharm y la computadora en la que se desea configurar.

## Capítulo III

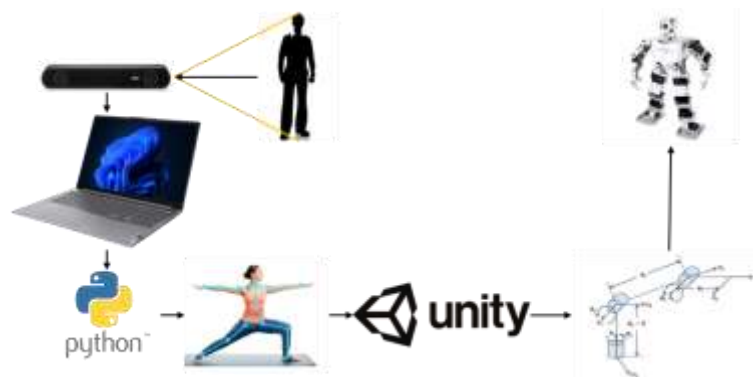
### Desarrollo e Implementación

Como se puede observar en la **Figura 40**, el proyecto se compone de 3 partes: Visión Estéreo emitida por la cámara a diseñar y seleccionar a la computadora, y el código realizado por Python. Método híbrido de cinemática inversa emitida por Python a Unity. Movimientos realizados por el robot.

La Visión Estéreo muestra los puntos de interés emitidos por CVZone y los envía a Unity. Unity se encarga de recopilar las coordenadas de los puntos de interés para aplicar un método híbrido de cinemática inversa y encontrar los ángulos que serán enviados al robot. El robot toma los ángulos correspondientes y los ejecuta para cada uno de sus Servomotores a usar.

#### Figura 40

*Representación gráfica de la funcionalidad del proyecto.*



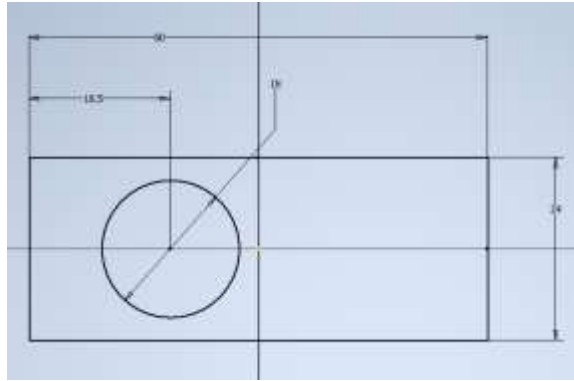
#### Diseño Mecánico del Soporte para las Cámaras, Cámara Estéreo

La cámara Genius FaceCam 1000X posee dimensiones ya predeterminadas como se puede observar en el bosquejo de la **Figura 41**. Por lo tanto, se realiza un diseño de soporte que contenga las dos cámaras a una distancia de  $63mm$  entre el centro de sus respectivos lentes debido a las dimensiones de las cámaras o a  $2.5mm$  entre la parte más externa del dispositivo de cámara como se observa en la **Figura 42** y en **Anexos** para que no haya choque entre ellas. El soporte cumple la función de inmovilizar las cámaras seleccionadas para que

permanezcan de forma paralela y la perspectiva de imagen captada no se modifique ya sea por el propio peso de las cámaras o por una fuerza externa por manipulación.

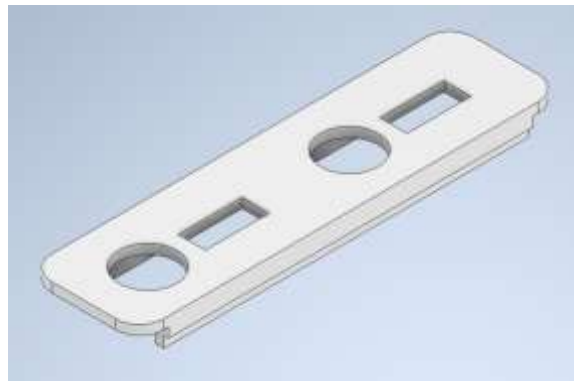
#### Figura 41

*Bosquejo de la vista frontal de la cámara, Inventor Autodesk.*



#### Figura 42

*Modelo 3D del soporte para las cámaras.*



A este soporte, se le realiza un análisis de tensión en Inventor Autodesk, sabiendo que el peso de cada cámara es de  $50g$  o  $0.491N$ , esta fuerza se aplica en varios puntos influenciado por la gravedad como se muestra en la **Figura 43**. Además, el operador ejerce una fuerza de agarre que se encuentra en un rango de hasta  $29kg$ , como en la manipulación no se busca aplastar al soporte si no movilizarlo, entonces la fuerza ejercida tomada en cuenta será un máximo igual al peso promedio de una laptop, que es de  $2.1kg$  o  $20.601 N$  y la fuerza ejercida por la gravedad como se observa en la **Figura 44**. Estas fuerzas son aplicadas de tal

forma que queden como se visualiza en la **Figura 45**. Las restricciones se encuentran en los agujeros para las cámaras.

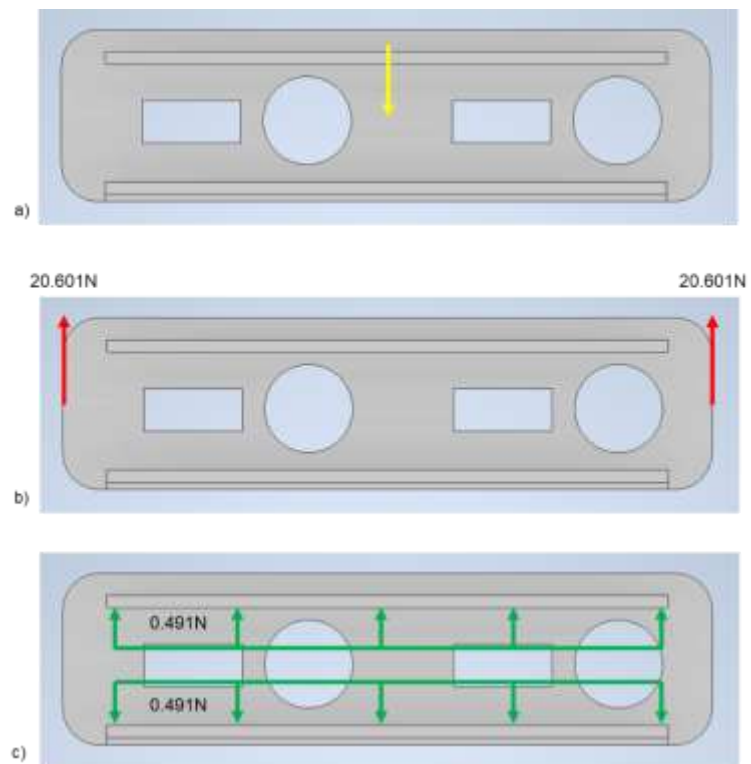
### Figura 43

*Fuerza que aplica la cámara por la gravedad.*



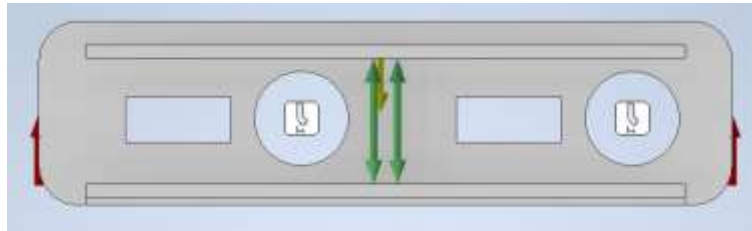
### Figura 44

*Cargas aplicadas. a) Gravedad b) Por manipulación c) Por las cámaras.*



### Figura 45

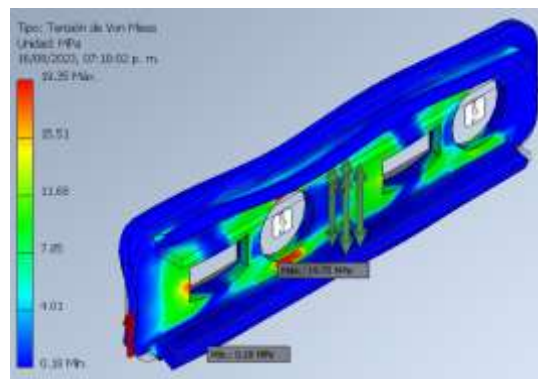
*Cargas aplicadas en el soporte de cámara, Inventor Autodesk.*



En la **Figura 46** se puede observar un esfuerzo máximo de  $19.35\text{MPa}$  y un mínimo de  $0.18\text{MPa}$ . En la **Figura 47** se puede observar un desplazamiento máximo de  $0.3771\text{mm}$ . En la **Figura 48** se puede observar un coeficiente de seguridad mínimo de  $1.03$  unidades dando como resultado un diseño aceptable para soportar desde cargas con solo las cámaras hasta cargas donde el operador lo manipula de forma brusca.

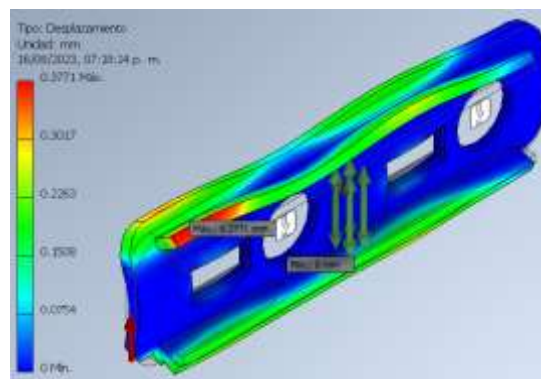
### Figura 46

*Tensión de Van Mises en el soporte de cámara.*



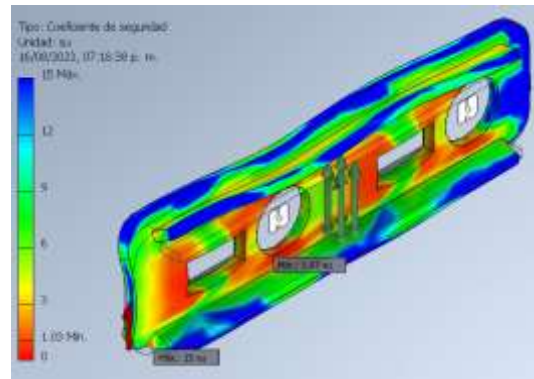
### Figura 47

*Desplazamiento en el soporte de cámara.*



## Figura 48

*Coefficiente de seguridad en el soporte de cámara.*



## Diseño de la Interfaz Humano Máquina (HMI)

El HMI se compone de dos ventanas que ayudan en la visualización e interacción con el programa principal, y mostrar la función de cada elemento de la ventana principal.

### ***Apariencia de las Ventanas***

Para la apariencia de diseño del programa se ha establecido que sea dependiente al Sistema Operativo que se usa para ejecutarlo. En este caso Windows 11, la apariencia dependerá de su configuración entre modo Oscuro y modo Claro.

```
ctk.set_appearance_mode('System')
```

### ***Tema de las Ventanas***

Para el tema, la librería Custom Tkinter, posee 3 diseños predefinidos y además se puede crear archivos de configuración personalizados para temas. Por defecto se ha escogido el tema verde ('green') aunque se puede editar el código para escoger uno de los temas existentes como el tema azul ('blue'), el tema azul oscuro ('dark-blue') o el tema personalizado ('dirección-del-archivo').

```
ctk.set_default_color_theme(Functions.default_values['screen_theme'])
```



## **Disposición**

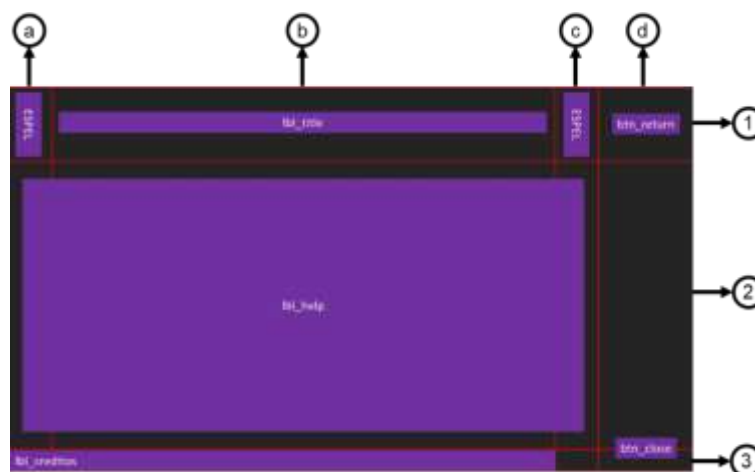
**Ventana Principal.** Siguiendo las pautas de la firma Nielsen Norman Group y verificándolo con las normas establecidas por la Organización Internacional de Normalización (ISO), la disposición de las herramientas está dada como en la **Figura 49**. Las imágenes a mostrar desde las cámaras izquierda y derecha se encuentran en la parte central y ocupan la mayor parte de la pantalla ya que de esta forma permite una mejor visualización para el operador, los botones a interactuar se encuentran en la parte derecha ordenados de arriba para abajo de tal forma que pueda ayudar al operador a seguir los pasos (mostrados en la ventana de ayuda) para un buen funcionamiento, el título es el que sigue en nivel de importancia ubicándose en la parte superior central dando información principal de la ventana, las cajas desplegadas para los indicadores de las cámaras derecha e izquierda se encuentran respectivamente y sobre las imágenes mostradas para poder relacionarlas con su respectiva imagen, los logos de la Institución se encuentran limitando a la izquierda y derecha del título para acompañar y dejar en claro la Institución a la que el proyecto está orientado, el error o advertencia de ejecución se encuentra en la parte inferior de las imágenes para una mejor visualización y los indicadores de texto no editables tienen el más bajo nivel de importancia como lo son los cuadros por segundo (FPS), los créditos y el identificador de cada imagen ya que el operador no interactúa ni cumplen una función más allá que la de dar información no crucial o importante.



**Ventana de Ayuda.** Para esta ventana, como se ve en la **Figura 50**, la imagen de pasos o funcionamiento de cada elemento se encuentra en la parte central usando varios espacios ya que es la parte más importante, el título se encuentra en la parte superior de la ventana, los logos de la institución se encuentran en la parte izquierda y derecha del título, los botones se encuentran en la parte superior e inferior derecha al igual que la anterior ventana y siendo el menos importante el indicador de texto no editable.

### Figura 50

*Disposición de las herramientas en la ventana de ayuda.*



Las filas están divididas en 3 teniendo como referencia las imágenes del logo, la imagen de los pasos o funciones de cada elemento de la ventana principal y el indicador de texto no editable para los créditos. Las columnas están divididas en 4 teniendo como referencia las imágenes del logo, el título y los botones.

Como las dimensiones de la ventana son iguales a la ventana principal, se tiene que:

El peso 1 está dado por la fila 3, y por las columnas a y c.

El peso 2 está dado por la fila 1, y por la columna d.

El peso 3 está dado por la fila 2, y por la columna b.

## Código del Programa en Pycharm para la Computadora

### Creación de las Librerías

Aparte de las librerías ya definidas que son anexos del lenguaje Python, existen dos archivos de este mismo lenguaje que sirven como anexos de código para mejorar el entendimiento al momento de revisarlo y editarlo. En este caso, se utilizan dos librerías:

**Librería de Diseño.** Esta librería tiene como nombre de archivo “Colors.py”, tiene como objetivo el guardar los valores en hexadecimal para los colores tanto de los textos como el diseño de las herramientas usadas por la librería *customtkinter*, y los valores en bits (de forma decimal) de los colores para la apariencia de la ventana según sea el caso del sistema (claro y oscuro).

```

###COLORES EN HEXADECIMAL

white = '#ffffff' #Blanco
black = '#000000' #Negro
green = '#2fa572' #Verde
orange = '#ff8c40' #Naranja
blue = '#1f6aa5' #Azul
dark_blue = '#1f538d' #Azul oscuro
red = '#cc0000' #Rojo
gray = '#808080' #Gris

###APARIENCIA

light_appearance = 242 #Modo Claro
dark_appearance = 26 #Modo Oscuro

```

**Librería de Funciones.** Esta librería tiene como nombre de archivo “Functions.py”, tiene como objetivo el guardar las funciones y variables repetitivas que pueden llamarse de forma externa.

```

import Colors

import customtkinter as ctk
import tkinter as tk
import numpy as np
import cv2

file_name = 'Assets/CameraIDs.txt'

```

```

default_values = {
    'frame_size': [640, 480],
    'screen_theme': 'green', #green, blue, dark-blue
    'screen_format': 16/9,
    'screen_width': 1280,
    'general_arguments': [10, 10],
    'button_arguments': [100, 0, 8, ('Consolas', 15, 'bold')],
    'title_arguments': [8, ('Consolas', 34, 'bold')],
    'label_arguments': [8, ('Consolas', 12, 'bold')],
    'error_arguments': [8, ('Consolas', 24, 'bold')],
    'credits_arguments': [8, ('Consolas', 9, 'italic')],
    'combobox_arguments': [8, ('Consolas', 14, 'bold')],
}

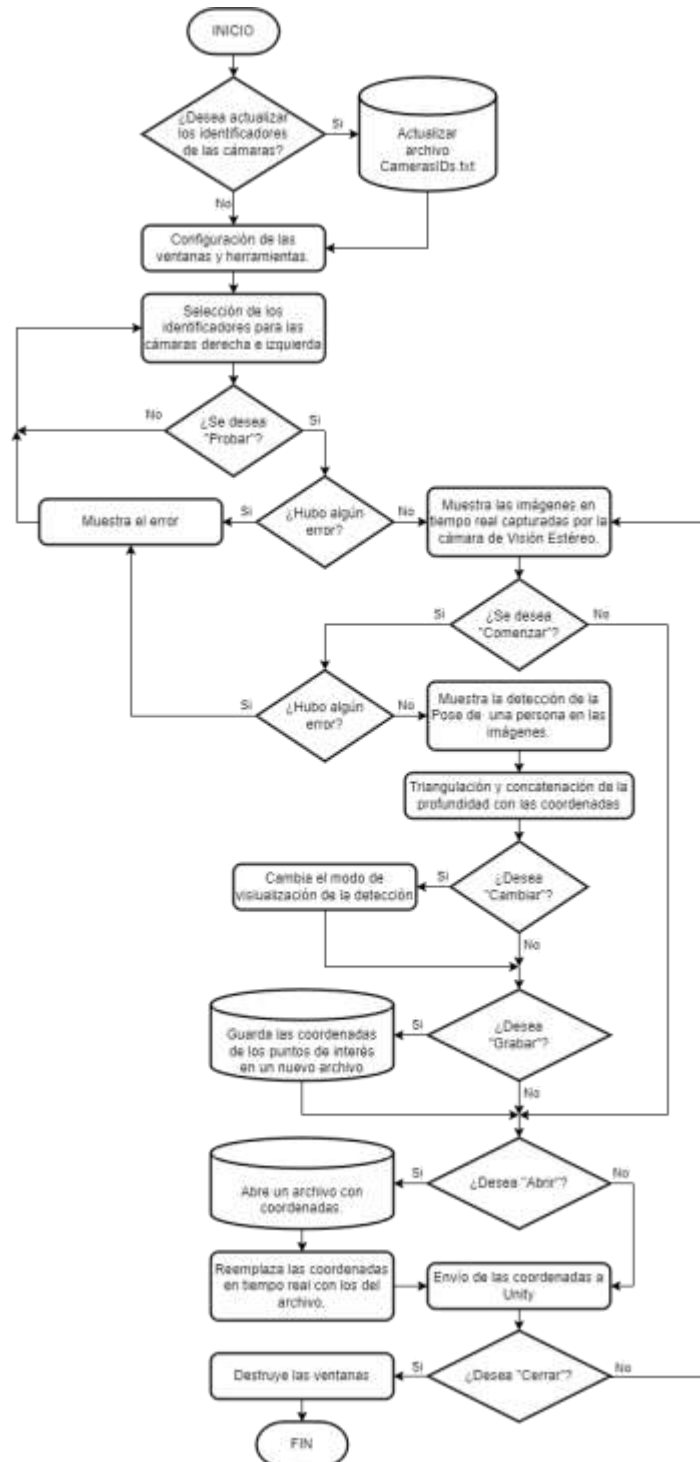
```

### ***Diagrama de Flujo***

El programa en PyCharm busca cumplir varias funciones que se pueden observar en el diagrama de la **Figura 51**. Busca actualizar los identificadores de las cámaras de forma opcional. Permite probar los identificadores escogidos para poder seguir el proceso. Se puede comenzar la detección de la POSE de la persona o abrir un archivo de formato “.txt” con las coordenadas ya guardadas anteriormente. Si se ha comenzado la detección, se muestra los puntos de referencia de CVZone y calcula la profundidad por triangulación para concatenar con las coordenadas en X e Y, y habilita el cambiar la forma de visualización de estos puntos y el guardar las coordenadas completas en un archivo de extensión “.txt”. Ya sea por abrir el archivo o por la detección se envían la trama completa de coordenadas por cada POSE a Unity, los puntos que se enviarían son aquellos que se encuentren en los brazos según CVZone en la **Figura 52**, estos son los que poseen un número igual a: 11, 12, 13, 14, 15, 16, 23 y 24. Por último, permite cerrar el programa completamente.

Figura 51

Diagrama de Flujo de PyCharm.



### **Triangulación de los Puntos de Referencia**

Para triangular la profundidad entre dos puntos de dos imágenes dispares por la perspectiva, es necesario utilizar las posiciones en X e Y de los puntos de referencia o articulaciones brindados por CVZone de la POSE de una persona. Esta está basada en la ecuación (1), dónde:

*Distancia entre cámaras* → 63mm

*Distancia focal* → 12

$$\bar{X} = |(X \text{ en la cámara derecha}) - (X \text{ en la cámara izquierda})| \quad (14)$$

$$\bar{Y} = |(Y \text{ en la cámara derecha}) - (Y \text{ en la cámara izquierda})| \quad (15)$$

$$d = \sqrt{\bar{X}^2 + \bar{Y}^2} \quad (16)$$

$$\text{profundidad} = \frac{(\text{Distancia focal})(\text{Distancia entre cámaras})}{d} \quad (17)$$

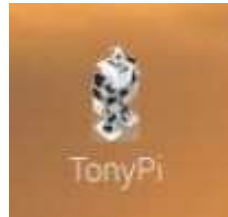
Siempre y cuando la diferencia entre las posiciones de las cámaras para las coordenadas X e Y en cada fotograma de las diferentes cámaras no sea igual a 0. Esta triangulación se realiza para los puntos que se enviarían a Unity.

### **Manipulación del Robot por Software**

Dentro del Sistema Operativo de la Raspberry del robot ya conectado a la computadora en su configuración inicial antes expuesta, se encuentra una aplicación (**Figura 52**) con la cual se puede manipular los límites de los Servomotores del robot y así poder relacionar el ángulo al que se encuentra cada Servo con el valor dado por el programa interno de control.

## Figura 52

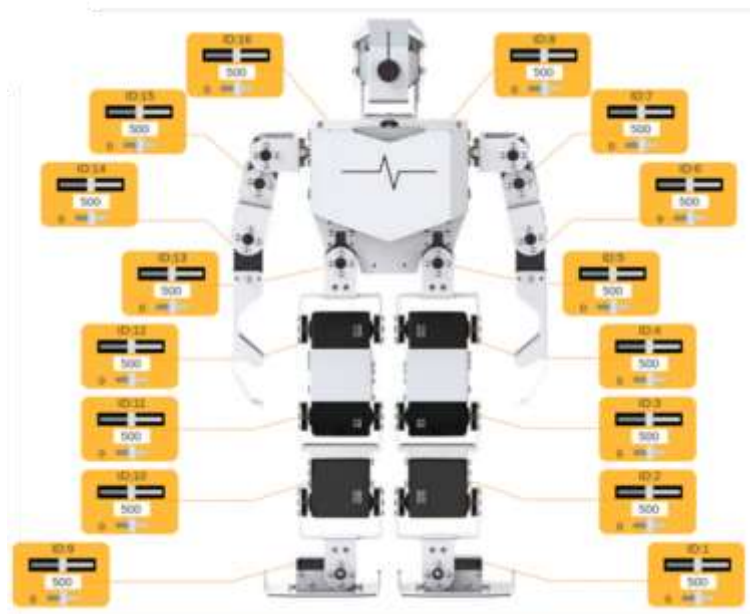
*Aplicación interna para manipular los Servomotores.*



Al ejecutarlo se puede observar una ventana (**Figura 53**) dónde los controles deslizantes permitirán modificar el ángulo del Servomotor con el identificador específico según la gráfica que posee esta ventana.

## Figura 53

*Ventana de la aplicación con controles deslizantes.*



## Relación entre las Articulaciones de la Persona y los GDL del Robot

Para el proyecto se le cambiará la designación de cada articulación y Servomotor, dando referencia a su nomenclatura dada por CVZone.

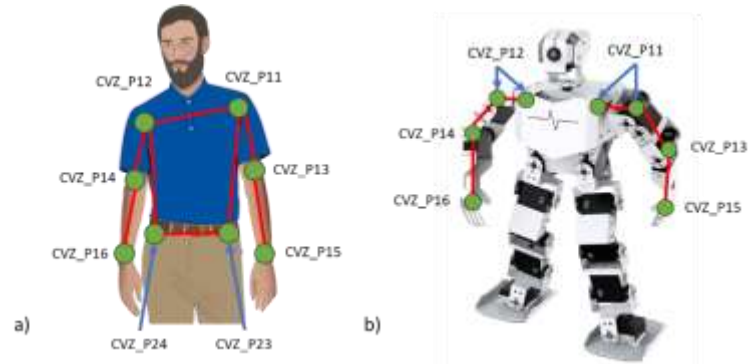
Como se muestra en la **Figura 54**: Las articulaciones CVZ\_P23, CVZ\_P24 (Pelvis) se las toman en cuenta como referencia para el resto de las articulaciones mas no porque tenga una articulación de relación. Las articulaciones CVZ\_P11 y CVZ\_P12 (Hombros) tienen como



relación hacia el robot, dos articulaciones siendo la una rotacional y la otra angular. Las articulaciones CVZ\_P13 y CVZ\_P14 (Codos) tiene una relación con el robot de solo una articulación. Las articulaciones CVZ\_P15 y CVZ\_P16 (Muñecas) son puntos finales de referencia para saber dónde llegará el robot.

### Figura 54

Relación entre GDL. a) Persona detectada b) Robot TonyPi Pro

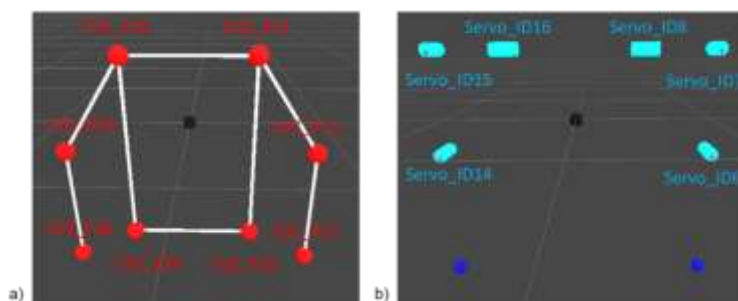


### Diseño del Entorno de Unity

El entorno de Unity está formado por dos cuerpos o conjunto de elementos, como se puede observar en la **Figura 55**, los cuales son: Los puntos dados por CVZone que son las articulaciones que posee la persona y los Servomotores del robot. El centro de los cuerpos posee una esfera de color negro para que destaque menos y sea posible tener una referencia de las ubicaciones de los elementos. Las articulaciones (siendo 8 ya explicadas en la programación de PyCharm) serán representadas de la misma forma que CVZone lo presenta en el HMI, las esferas/puntos de las articulaciones de color rojo y los eslabones de color blanco. La representación de los Servomotores (siendo 6 que se explicarán después) serán cilindros que tendrán un color cian para destacar y diferenciarlos con las articulaciones del robot orientándolos con la parte redonda al eje de giro, y los puntos finales (muñeca del robot) serán representados por una esfera de color azul.

**Figura 55**

Representación en Unity. a) Articulaciones de la persona b) Servomotores del robot



Para que los Servomotores funcionen como debería ser en Unity, estos deben ser hijos de los que dependa. Por ejemplo: El punto final azul del brazo izquierdo del robot es dependiente del SERVO\_ID8 en palabras del programa Unity, el punto es hijo del SERVO\_ID8.

### Código y Configuración del Programa en Unity

#### Configuración Inicial

Es necesario dar valor de coordenadas, escala y rotación como en la **Figura 56** para cada elemento colocado en el diseño del entorno de Unity, de forma inicial. Todos estos valores se seleccionan para que no interfieran entre sí en su visualización.

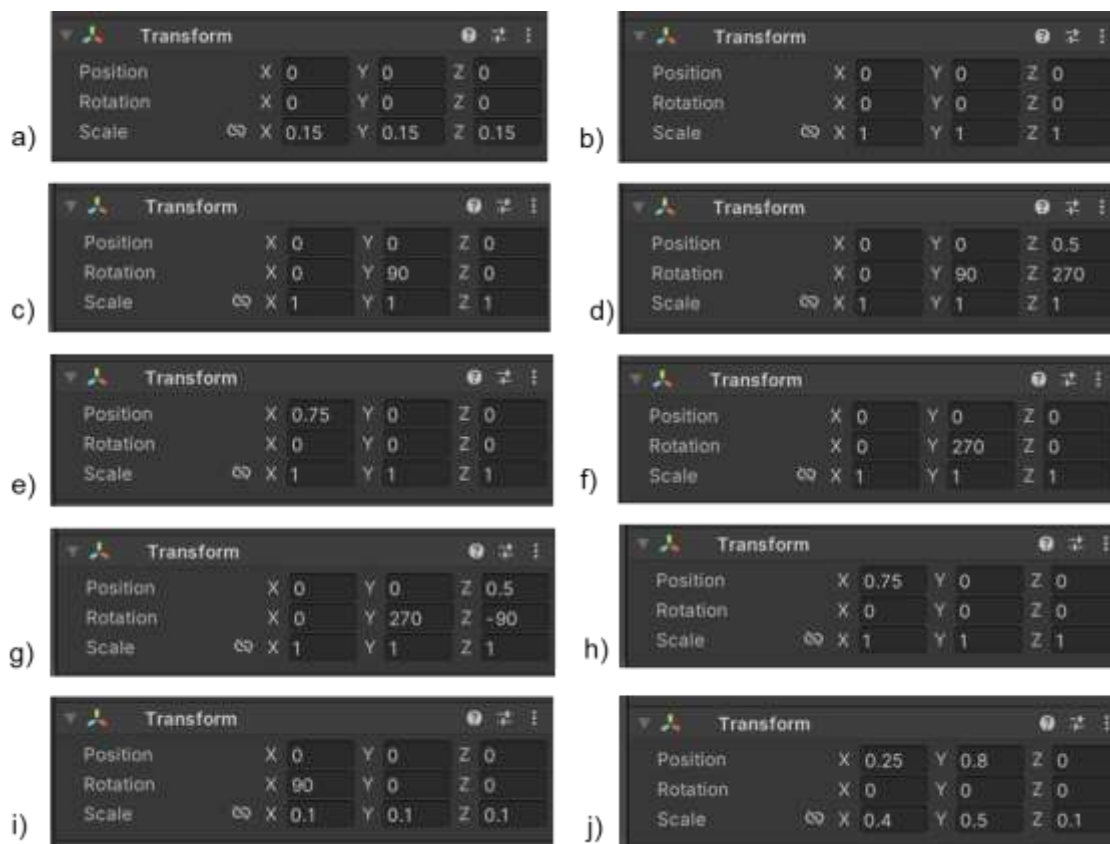
**Coordenadas de los Elementos.** Las articulaciones pueden ubicarse en todo el espacio de trabajo, aunque se recomienda dejarlos en el centro con una ubicación 0 en  $X$ , 0 en  $Y$  y 0 en  $Z$ ; al igual que los eslabones que los une. Los Servomotores tienen su propia distancia, son dependientes y están escalados con respecto a los Servomotores reales del robot.

**Escala de los Elementos.** Las articulaciones poseen una escala en todos los ejes de 0.15 *unidades*. Los Servomotores poseen una escala en todos los ejes de 0.1 *unidad*.

**Rotación de los Elementos.** Las articulaciones no sufren ninguna rotación, por lo tanto, el ángulo en las con respecto a los tres ejes es 0; al igual que los eslabones que los une. Los Servomotores tienen su propia rotación que depende de la orientación de los Servomotores reales del robot.

**Figura 56**

Posición, rotación y escala. a) Articulaciones b) Eslabones c) SERVO\_ID8 d) SERVO\_ID7 e) SERVO\_ID6 f) SERVO\_ID16 g) SERVO\_ID15 h) SERVO\_ID14 i) Cilindros j) Cubos del cilindro



### Configuración de los Eventos del Mando

El mando genérico de XBOX posee una comunicación por un Tomador que puede ser conectado por USB. El mando posee varios botones que se pueden ingresar en el paquete "Input Action" como se encuentran en la **Figura 57** para que quede como la **Figura 58**.

### Figura 57

*Botones que contiene un mando de XBOX genérico.*



Dónde:

1 y 9 representan al evento “A”, “B”, “X” e “Y”.

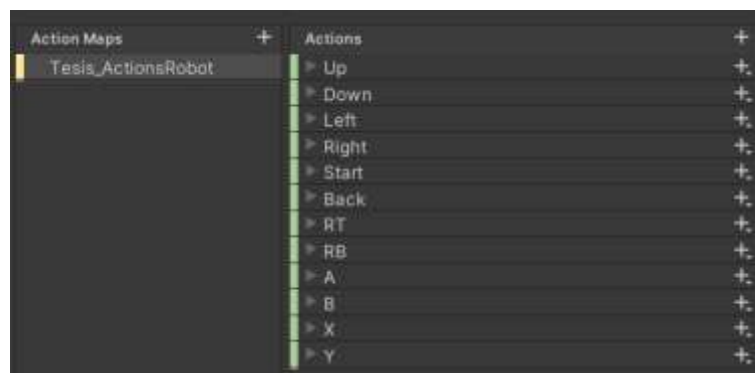
2 y 10 representan al evento “Up”, “Down”, “Left”, “Right”.

3 y 8, y 4 y 7 representan al evento “RB” y “RT”

5 y 6 representan al evento “Back” y “Start”

### Figura 58

*Eventos de mando en Input Actions.*



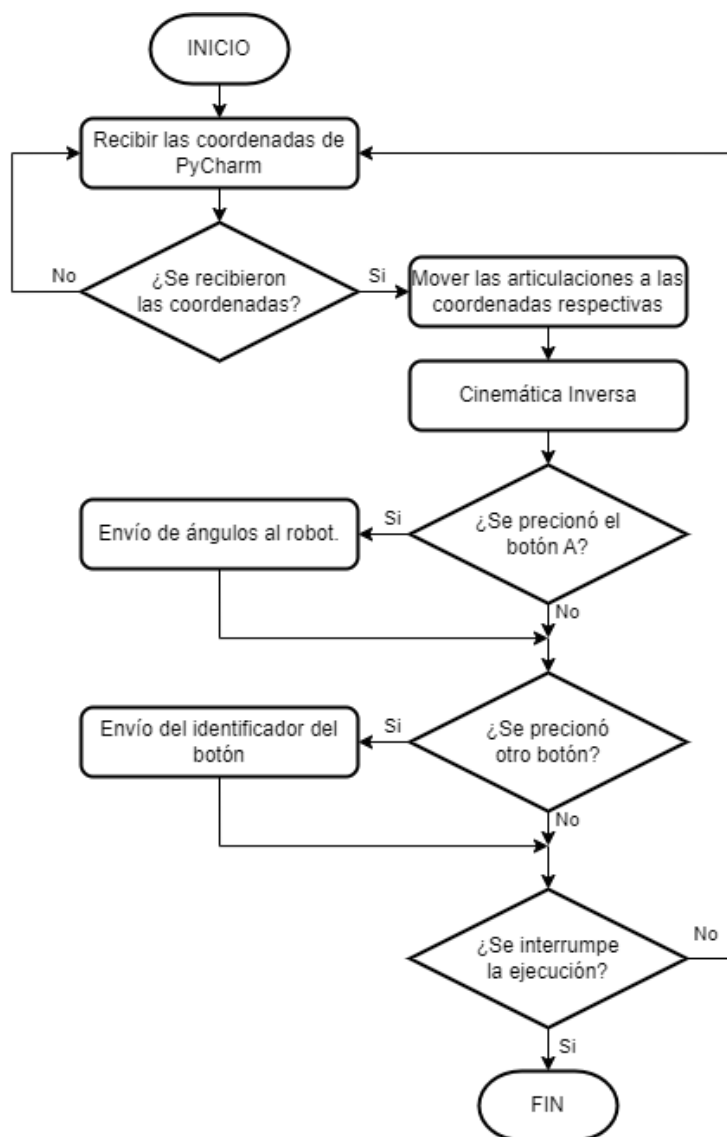
### Diagrama de Flujo del Código

El programa en Unity busca cumplir varias funciones que se pueden observar en el diagrama de la **Figura 59**. Busca recibir las coordenadas que se enviaron desde PyCharm. Si existen y cumplen las condiciones de trama, las representaciones de las articulaciones se mueven en el espacio según la trama recibida. Se aplica Cinemática Inversa (CI) para las

representaciones de los Servomotores para que el punto final llegue más o menos a la articulación final correspondiente. La CI está siendo aplicada gracias a un paquete externo de Unity "Hybrid IK" que trata de un algoritmo híbrido entre FABRIK y CCD para las interpolaciones suaves con restricciones de límites. Por último, espera una señal del mando remoto para mandar o no los ángulos en tiempo real, o el enviar el identificador al Robot.

**Figura 59**

*Diagrama de Flujo de Unity.*



## Límites de las Articulaciones del Robot Humanoide

Los Servomotores se manejan por código con valores entre 0 y 1000. Los valores “límite de trabajo” se dará según el programa interno de la **Figura 53** en cambio los valores de “límite de proyecto” serán seleccionados de manera que puedan imitar el movimiento de las articulaciones de una persona detectada, considerando sus restricciones. Por lo tanto, los límites según la posición en Unity serán:

### Límites en el Brazo Izquierdo

**Tabla 17**

*Límites de trabajo, origen y límites de proyecto para el brazo izquierdo.*

	Ángulo (°)	Valor
<b>SERVO_ID8</b>		
<b>Límites de trabajo</b>	-60	1000
	180	0
<b>Origen</b>	0	$O_8$
<b>Límites de proyecto</b>	-15	$A_8$
	165	$B_8$
<b>SERVO_ID7</b>		
<b>Límites de trabajo</b>	300	1000
	60	0
<b>Origen</b>	180	$O_7$
<b>Límites de proyecto</b>	270	$A_7$
	150	$B_7$
<b>SERVO_ID6</b>		
<b>Límites de trabajo</b>	135	1000
	-135	0
<b>Origen</b>	0	$O_6$
<b>Límites de proyecto</b>	90	$A_6$
	-30	$B_6$

$$O_8 = \frac{(0) - (180)}{(-60) - (180)} [(1000) - (0)] = 750 \quad (18)$$

$$A_8 = \frac{(-15) - (180)}{(-60) - (180)} [(1000) - (0)] = 812.5 \cong 813 \quad (19)$$

$$B_8 = \frac{(165) - (180)}{(-60) - (180)} [(1000) - (0)] = 62.5 \cong 63 \quad (20)$$

$$O_7 = \frac{(180) - (60)}{(300) - (60)} [(1000) - (0)] = 500 \quad (21)$$

$$A_7 = \frac{(270) - (60)}{(300) - (60)} [(1000) - (0)] = 875 \quad (22)$$

$$B_7 = \frac{(150) - (60)}{(300) - (60)} [(1000) - (0)] = 375 \quad (23)$$

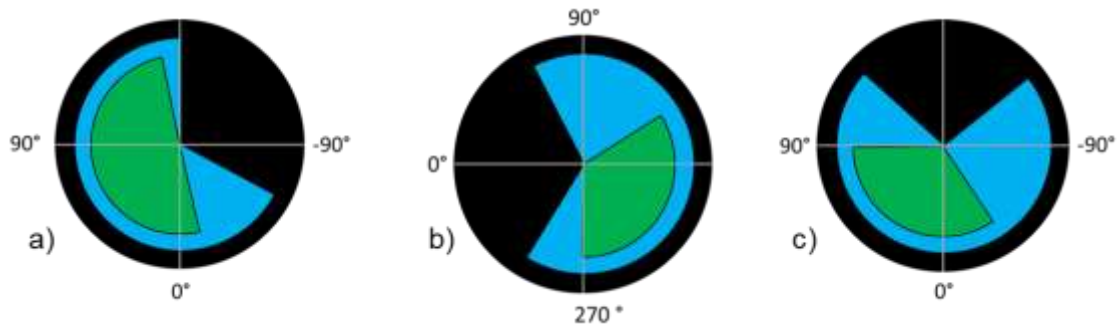
$$O_6 = \frac{(0) - (-135)}{(135) - (-135)} [(1000) - (0)] = 500 \quad (24)$$

$$A_6 = \frac{(90) - (-135)}{(135) - (-135)} [(1000) - (0)] = 833.33 \cong 833 \quad (25)$$

$$B_6 = \frac{(-30) - (-135)}{(135) - (-135)} [(1000) - (0)] = 388.88 \cong 389 \quad (26)$$

### Figura 60

Representación gráfica de los límites, proyecto (verde) y trabajo (azul). a) SERVO\_ID8 b) SERVO\_ID7 c) SERVO\_ID6.



### Límites en el Brazo Derecho

**Tabla 18**

*Límites de trabajo, origen y límites de proyecto para el brazo derecho.*

	Ángulo (°)	Valor
<b>SERVO_ID16</b>		
<b>Límites de trabajo</b>	-180	1000
	60	0
<b>Origen</b>	0	$O_{16}$
<b>Límites de proyecto</b>	-165	$A_{16}$
	15	$B_{16}$
<b>SERVO_ID15</b>		
<b>Límites de trabajo</b>	120	1000
	-120	0
<b>Origen</b>	0	$O_{15}$
<b>Límites de proyecto</b>	30	$A_{15}$
	-90	$B_{15}$
<b>SERVO_ID14</b>		
<b>Límites de trabajo</b>	135	1000
	-135	0
<b>Origen</b>	0	$O_{14}$
<b>Límites de proyecto</b>	30	$A_{14}$
	-90	$B_{14}$

$$O_{16} = \frac{(0) - (60)}{(-180) - (60)} [(1000) - (0)] = 250 \quad (27)$$

$$A_{16} = \frac{(-165) - (60)}{(-180) - (60)} [(1000) - (0)] = 937.5 \cong 938 \quad (28)$$

$$B_{16} = \frac{(15) - (60)}{(-180) - (60)} [(1000) - (0)] = 187.5 \cong 188 \quad (29)$$

$$O_{15} = \frac{(0) - (-120)}{(120) - (-120)} [(1000) - (0)] = 500 \quad (30)$$

$$A_{15} = \frac{(30) - (-120)}{(120) - (-120)} [(1000) - (0)] = 625 \quad (31)$$

$$B_{15} = \frac{(-90) - (-120)}{(120) - (-120)} [(1000) - (0)] = 125 \quad (32)$$

$$O_{14} = \frac{(0) - (-135)}{(135) - (-135)} [(1000) - (0)] = 500 \quad (33)$$

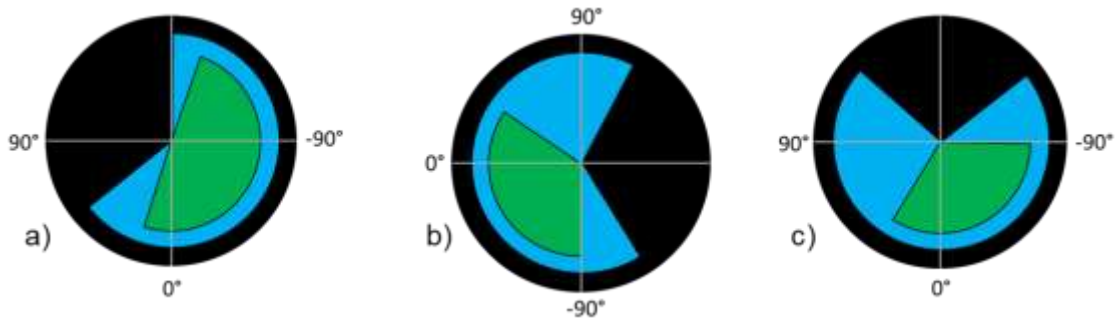


$$A_{14} = \frac{(30) - (-135)}{(135) - (-135)} [(1000) - (0)] = 611.11 \cong 611 \quad (34)$$

$$B_{14} = \frac{(-90) - (-135)}{(135) - (-135)} [(1000) - (0)] = 166.67 \cong 167 \quad (35)$$

### Figura 61

Representación gráfica de los límites, proyecto (verde) y trabajo (azul). a) SERVO\_ID16 b) SERVO\_ID15 c) SERVO\_ID14



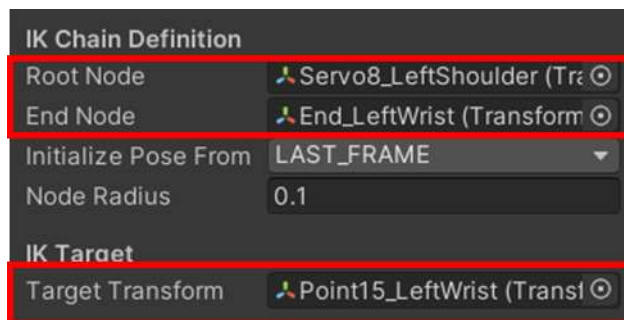
### Método de Cinemática Inversa.

Para la cinemática inversa es necesario configurar los elementos de inicio del sistema, punto final del sistema y punto de seguimiento del sistema, además, se debe ingresar los valores límites del proyecto para cada Servomotor. El ingreso de los límites depende del eje de giro, el rango de grados y el desfase con respecto al origen. Donde:

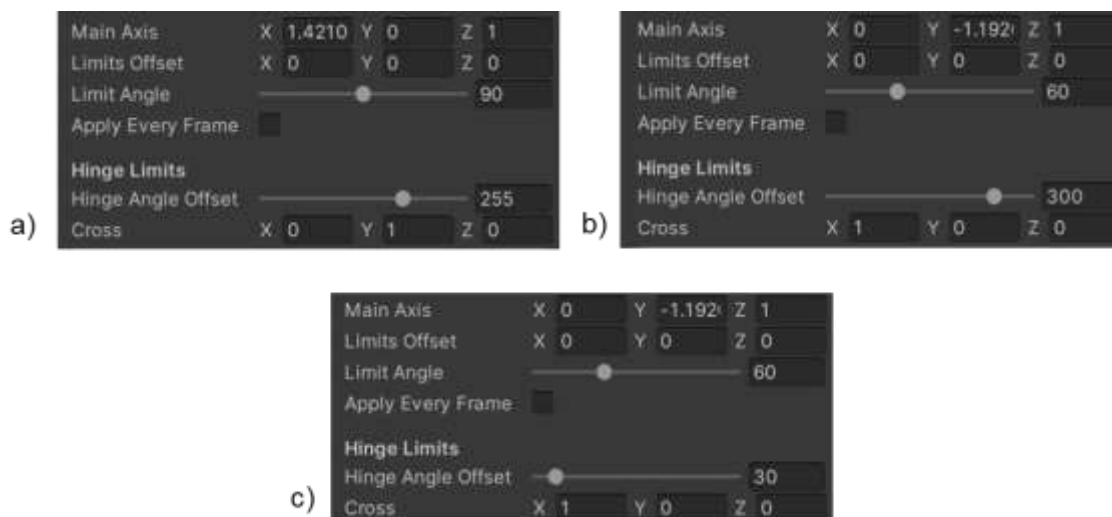
Para el brazo izquierdo, con nombres de Unity, el Servomotor inicial del sistema es el "Servo8\_LeftShoulder", el punto final es el "End\_LeftWrist" y el punto de seguimiento es "Point15\_LeftWrist" como se muestra en la **Figura 62**. La configuración de los límites está dada por la **Figura 63**.

**Figura 62**

Configuración de Hybrid IK, del brazo izquierdo.

**Figura 63**

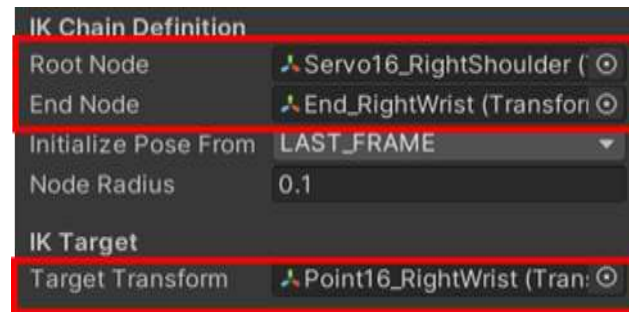
Configuración de límites en Hybrid IK. a) SERVO\_ID8 b) SERVO\_ID7 c) SERVO\_ID6.



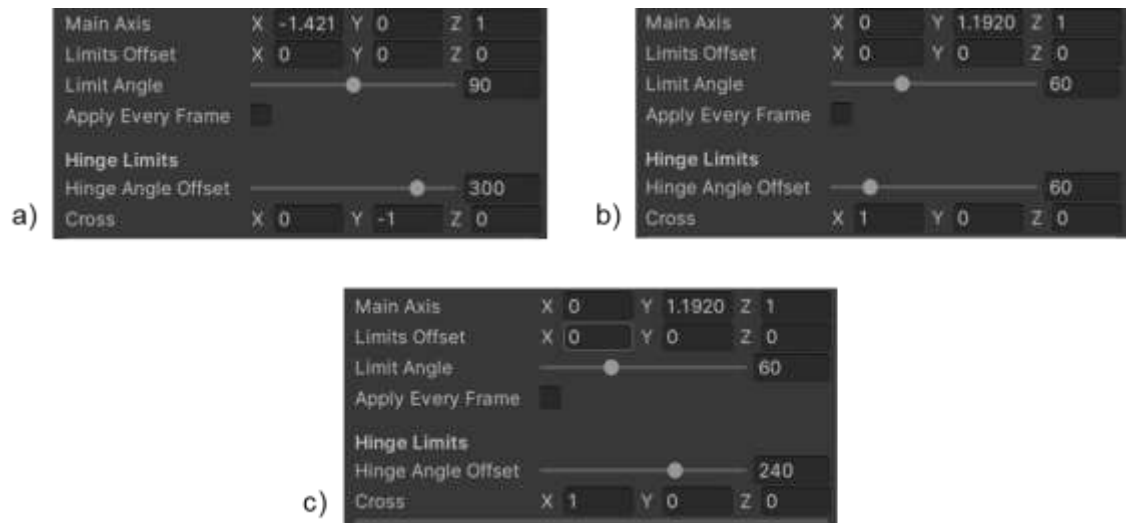
Para el brazo derecho, con nombres de Unity, el Servomotor inicial del sistema es el “Servo16\_LeftShoulder”, el punto final es el “End\_RightWrist” y el punto de seguimiento es “Point15\_RightWrist” como se puede observar en la **Figura 64**. La configuración de los límites está dada por la **Figura 65**.

**Figura 64**

Configuración de Hybrid IK, del brazo derecho.

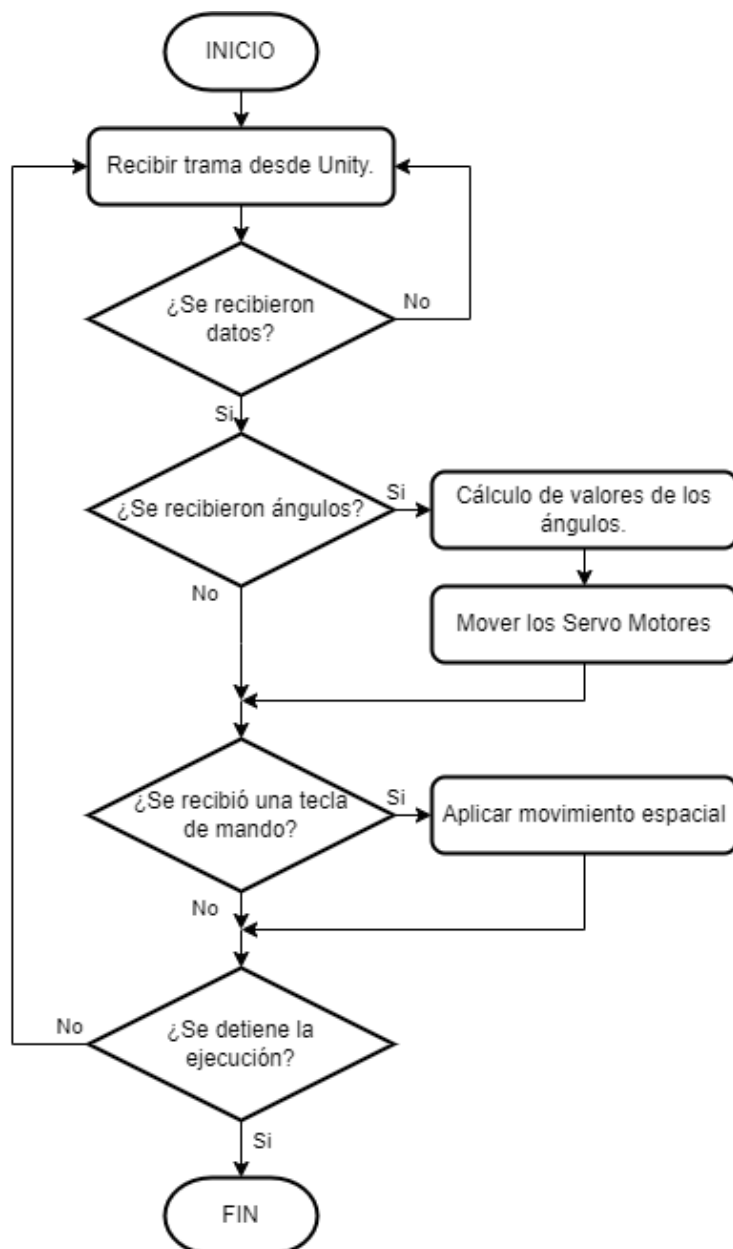
**Figura 65**

Configuración de límites en Hybrid IK. a) SERVO\_ID16 b) SERVO\_ID15 c) SERVO\_ID14.



### Código del Robot en PyCharm

El programa en PyCharm para programar y manejar el robot, busca cumplir varias funciones que se pueden observar en el diagrama de la **Figura 66**. Busca el recibir la trama enviada desde Unity al robot de forma remota. Si la trama contiene los ángulos obtenidos, procede a calcular el valor correspondiente para el Servomotor, y los mueve llegando a un ángulo aproximado. Si la trama contiene el identificador del botón presionado, realiza la acción predeterminada como es el agacharse, levantarse, caminar o girar. Por último, si se detiene el programa, termina la comunicación y el manejo del robot.

**Figura 66***Diagrama de Flujo del robot.*

## Capítulo IV

### Pruebas, Resultados y Validación de la Hipótesis

#### Pruebas y Resultados

##### *Visión Estéreo*

Para una prueba de iluminación contra luz haciendo que el exterior sea captado por las cámaras como se muestra en la **Figura 67**, la detección de la persona puede ser nula, haciendo que no se muestre el resultado de CVZone en cada imagen.

##### **Figura 67**

*Prueba contra luz captando el exterior.*



Por lo contrario, para una iluminación contra luz, pero haciendo que el exterior no sea captado por las cámaras como en la **Figura 68**, la detección puede mostrar los resultados de CVZone en cada imagen siendo la ropa clara una buena manera de dar más detalles a la detección.

### Figura 68

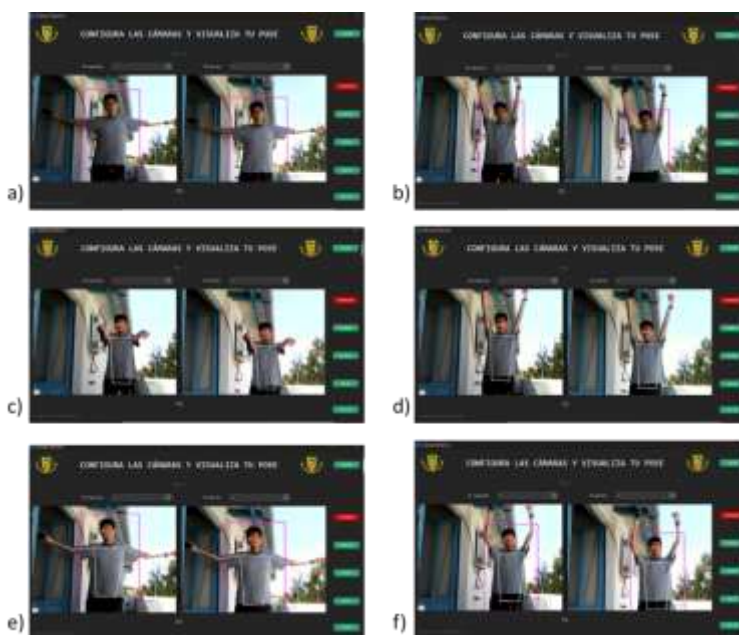
Prueba contra luz sin captar el exterior. a) Ropa oscura b) Ropa clara.



Para una iluminación clara como en la **Figura 69**, la detección es mucho más eficiente permitiendo observar con mayor fluidez el resultado emitido por CVZone. El programa permite la visualización de la detección de 3 formas distintas: solo un cuadro que contenga la detección sin tener en cuenta los brazos, solo los puntos de referencia emitidos por CVZone, ambas visualizaciones a la vez.

### Figura 69

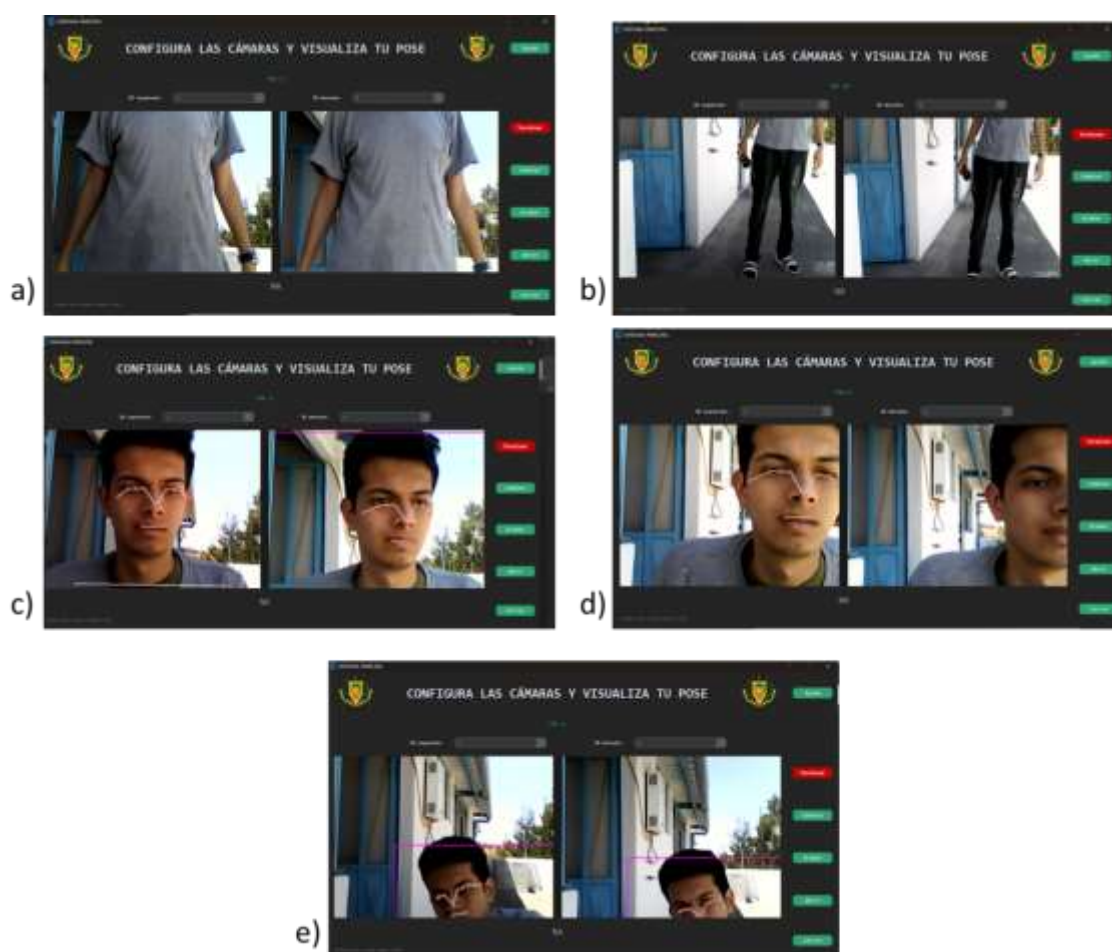
Prueba con iluminación clara. a) Cuadro de detección, brazos estirados. b) Cuadro de detección, brazos alzados. c) Puntos de referencia, brazos estirados. d) Puntos de referencia, brazos alzados e) Cuadro de detección y puntos de referencia, brazos estirados. f) Cuadro de detección y puntos de referencia, brazos alzados.



Según la **Figura 70**, la detección toma el rostro como el área de mayor valor para diferenciar entre una persona y un objeto del fondo. Si el rostro no aparece, por lo menos las primeras veces de aparición, no existirá algún resultado. Si el rostro aparece de forma parcial, la detección puede mostrar resultados según el seguimiento y estimación de la POSE del cuerpo.

### Figura 70

*Prueba de detección parcial. a) Solo el torso b) Solo las piernas c) Solo el rostro d) Mitad lateral del rostro e) Mitad vertical del rostro*



Para más de una persona en la imagen como se puede observar en la **Figura 71**, la detección tomará aquella con mejores características priorizando aquella que se encuentre más a la derecha de la imagen.



### Figura 71

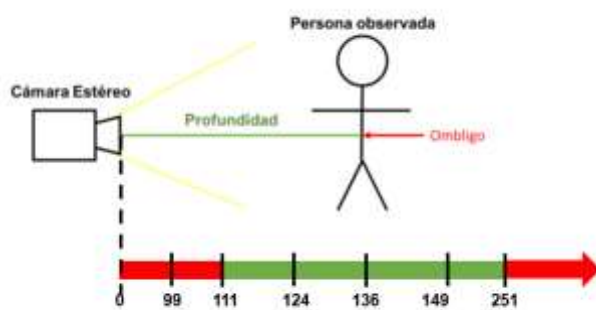
Prueba de detección con varias personas. a) Detección por mayor características b) Detección por prioridad de posición



Para medir la profundidad o distancia entre el cuerpo observado por las cámaras y las mismas cámaras, se toma como punto de referencia el ombligo de la persona y el centro de la cámara de Visión Estéreo como se observa en la **Figura 72** y la **Figura 73**.

### Figura 72

Representación gráfica de la medición de la profundidad.



### Figura 73

Medición vista desde el programa.





Los valores recopilados por la triangulación para distancias de 99 (**Tabla 19**), 111 (**Tabla 20**), 124 (**Tabla 21**), 136 (**Tabla 22**) y 149 (**Tabla 23**) cm durante 1 minuto de ejecución son:

**Tabla 19**

Valores calculados, valor real 99 cm.

Lista de valores calculados para una profundidad de 99 cm						
109	106	106	99	96	115	100
103	95	106	99	109	98	
Error absoluto individual (cm)						
10	7	7	0	-3	16	1
4	-4	7	0	10	-1	

El error absoluto de la lista con el valor real está dado por la Ecuación (36), el error relativo porcentual está dado por la Ecuación (37).

$$error_{absoluto} = \frac{\sum_{i=1}^n |Valor\ real - Valor\ calculado_i|}{n} \quad (36)$$

$$error_{absoluto} = \frac{10 + 7 + 7 + 0 + 3 + 16 + 1 + 4 + 4 + 7 + 0 + 10 + 1}{13}$$

$$error_{absoluto} = 5.3846\text{ cm}$$

$$error_{relativo} = \frac{error_{absoluto}}{Valor\ real} * 100\% \quad (37)$$

$$error_{relativo} = \frac{5.3846\text{ cm}}{99\text{ cm}} * 100\%$$

$$error_{relativo} = 5.439\%$$

**Tabla 20**

Valores calculados, valor real 111 cm.

Lista de valores calculados para una profundidad de 111 cm						
111	112	111	108	117	116	110
110	114	118	107	117	112	
Error absoluto individual (cm)						
0	1	0	-3	6	5	-1
-1	3	7	6	6	1	

$$error_{absoluto} = \frac{1 + 3 + 6 + 5 + 1 + 1 + 3 + 7 + 6 + 6 + 1}{13}$$

$$error_{absoluto} = 3.0769 \text{ cm}$$

$$error_{relativo} = \frac{3.0769 \text{ cm}}{111 \text{ cm}} * 100\%$$

$$error_{relativo} = 2.772\%$$

**Tabla 21**

Valores calculados, valor real 124 cm.

Lista de valores calculados para una profundidad de 124 cm

131	128	118	124	129	130	123
119	126	117	128	117	118	
Error absoluto individual (cm)						
7	4	-6	0	5	6	-1
-5	2	-7	4	-7	-6	

$$error_{absoluto} = \frac{7 + 4 + 6 + 5 + 6 + 1 + 5 + 2 + 7 + 4 + 7 + 6}{13}$$

$$error_{absoluto} = 4.6154 \text{ cm}$$

$$error_{relativo} = \frac{4.6154 \text{ cm}}{124 \text{ cm}} * 100\%$$

$$error_{relativo} = 3.7221\%$$

**Tabla 22**

Valores calculados, valor real 136 cm.

Lista de valores calculados para una profundidad de 136 cm

133	136	140	144	135	129	136
143	128	130	140	137	143	
Error absoluto individual (cm)						
-3	0	4	8	-1	-7	0
7	-8	-6	4	1	7	

$$error_{absoluto} = \frac{3 + 4 + 8 + 1 + 7 + 7 + 8 + 6 + 4 + 1 + 7}{13}$$

$$error_{absoluto} = 4.3077 \text{ cm}$$

$$error_{relativo} = \frac{4.3077 \text{ cm}}{136 \text{ cm}} * 100\%$$

$$error_{relativo} = 3.1674\%$$

**Tabla 23**

Valores calculados, valor real 149 cm.

Lista de valores calculados para una profundidad de 149 cm						
142	144	146	148	147	148	150
150	154	156	147	147	148	
Error absoluto individual (cm)						
7	5	3	1	2	1	-1
-1	-5	-7	2	2	1	

$$error_{absoluto} = \frac{7 + 5 + 3 + 1 + 2 + 1 + 1 + 1 + 5 + 7 + 2 + 2 + 1}{13}$$

$$error_{absoluto} = 2.9231 \text{ cm}$$

$$error_{relativo} = \frac{2.9231 \text{ cm}}{149 \text{ cm}} * 100\%$$

$$error_{relativo} = 1.9618\%$$

**Tabla 24**

Valores calculados, valor real 251 cm.

Lista de valores calculados para una profundidad de 251 cm						
260	247	265	260	247	246	255
260	256	262	261	258	261	
Error absoluto individual (cm)						
9	-4	14	9	-4	-5	4
9	5	11	10	7	10	

$$error_{absoluto} = \frac{9 + 4 + 14 + 9 + 4 + 5 + 4 + 9 + 5 + 11 + 10 + 7 + 10}{13}$$

$$error_{absoluto} = 7.7692 \text{ cm}$$

$$error_{relativo} = \frac{7.7692 \text{ cm}}{251 \text{ cm}} * 100\%$$

$$error_{relativo} = 3.0953\%$$

Para distancias menores e iguales a los 110 cm, la profundidad calculada posee un error relativo no aceptable del 5.44% y un error absoluto del 5.39cm debido a que el punto a medir pierde la referencia por los límites resolución de imagen, este error aumentaría a medida que se acerca al dispositivo de Cámara Estéreo. A partir de los 111 cm hasta los 250 cm la

referencia pudo ser observada dando errores relativos menores y aceptables desde el 1.96% hasta un error de 3.72%, y un error absoluto desde los 2.92 *cm* hasta los 4.62 *cm*. Desde los 251 *cm* la referencia puede confundirse con otros puntos del cuerpo de la persona detectada además de tener un error relativo del 3.10% y con un gran error absoluto de 7.77 *cm*.

### **Control Remoto**

La latencia o ping para la comunicación entre PyCharm y Unity es de 0.19ms como media según el resultado mostrado en la **Figura 74**.

#### **Figura 74**

*Prueba de comunicación entre PyCharm y Unity.*

```
C:\Users\Det-Pc\Desktop\Tesis\PythonCodes\PcCode01\venv\Scripts\python.exe C:\Users\Det-Pc
Universidad de las Fuerzas Armadas ESPE Sede Latacunga! como mensaje enviado
Se recibí? Universidad de las Fuerzas Armadas ESPE Sede Latacunga! como mensaje recibido
La latencia promedio de 100 mensajes enviados es de 0.19 ms

Process finished with exit code 0
```

La latencia o ping para la comunicación entre Unity y el robot TonyPi Pro es de 18.16ms como media según el resultado mostrado en la **Figura 75**.

#### **Figura 75**

*Prueba de comunicación entre Unity y TonyPi Pro.*

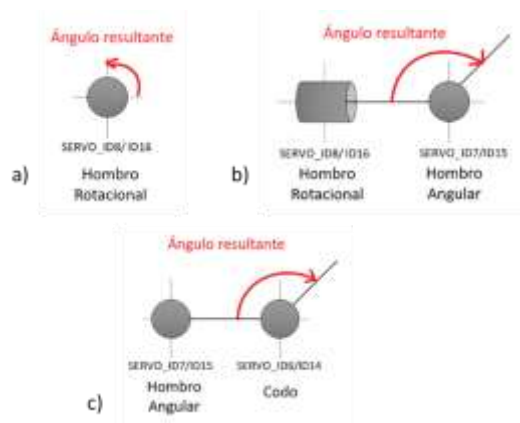
```
[08:41:47] Mensaje enviado: Universidad de las Fuerzas Armadas ESPE Sede Latacunga!
UnityEngine.MonoBehaviour:print (object)
[08:41:47] Mensaje recibido: Se recibió! Universidad de las Fuerzas Armadas ESPE Sede Latacunga!
UnityEngine.MonoBehaviour:print (object)
[08:41:47] Latencia promedio de 100 mensajes: 18.16 ms
UnityEngine.MonoBehaviour:print (object)
```

### **Conversión entre los GDL de una Persona a GDL de un Robot**

Los resultados de las pruebas para los siguientes Servomotores se obtuvieron de la medición del ángulo más corto, como se muestra en la **Figura 76** y la **Figura 77**.

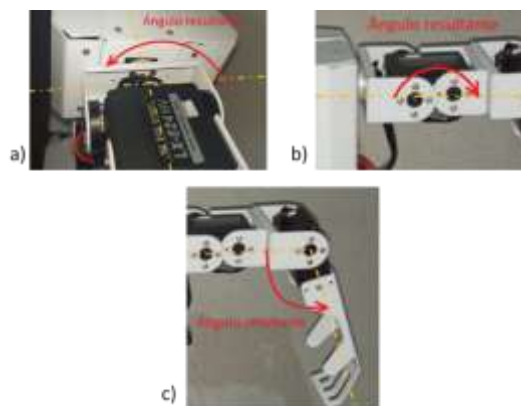
### Figura 76

Representación gráfica de cómo se mide el ángulo para cada articulación. a) Ángulo de los Servos de ID 8 y 16. b) Ángulo de los Servos de ID 7 y 15. c) Ángulo de los Servos de ID 6 y 14.



### Figura 77

Representación en el robot de cómo se mide el ángulo para cada articulación. a) Ángulo de los Servos de ID 8 y 16. b) Ángulo de los Servos de ID 7 y 15. c) Ángulo de los Servos de ID 6 y 14.



Por lo tanto, se tienen los siguientes resultados al momento de medir y comprobar:

Tabla 25

Error de resultado al convertir GDL.

Valor enviado	Ángulo medido (°)	Ángulo calculado (°)	Error (%)
<b>Servomotor ID 8 – Hombro del brazo izquierdo</b>			
813	-15	-15	0
782	-7	-7.56	7.41
407	83	82.44	0.68
63	165	165	0
<b>Servomotor ID 6 – Codo del brazo izquierdo</b>			
833	85	90	5.56
667	45	45.14	0.31
445	-15	-14.86	0.94
389	-30	-30	0
<b>Servomotor ID 16 – Hombro del brazo derecho</b>			
938	-165	-165	0
594	-83	-82.44	0.68
219	8	7.56	5.82
188	15	15	0
<b>Servomotor ID 15 – Hombro del brazo derecho</b>			
625	30	30	0
563	15	15.12	0.79
313	-45	-44.88	0.27
125	-90	-90	0
<b>Servomotor ID 14 – Codo del brazo derecho</b>			
611	30	30	0
556	15	15.14	0.92
334	-45	-44.86	0.31
167	-85	-90	5.56

Dando como resultado los siguientes errores para cada Servomotor medido:

Tabla 26

Error acumulado al convertir los GDL.

Servomotores	Error (%)
ID 8	2.02
ID 6	1.7
ID 16	1.63
ID 15	0.27
ID 14	1.7
<b>Total</b>	<b>7.32</b>

### ***Movimientos Gestuales y Espaciales del Robot***

Se tiene que entre la persona y PyCharm se obtiene un error máximo del 10% debido a su confidencialidad del 90%, entre PyCharm y Unity existe un error de casi 0% debido a la mínima latencia que existe en su comunicación, entre Unity y el robot se tiene un error acumulativo de casi 7.32% al convertir los datos enviados o individual de aproximadamente 1.46%. Por lo tanto, en todo el proyecto existe un error acumulativo del 17.32% e individual de 11.46%. La mayor parte del error se lleva la Visión Artificial ya que para trabajar con un mayor índice de confiabilidad se necesita de casos ideales de iluminación y otras características para que pueda detectar de forma eficiente a la persona.

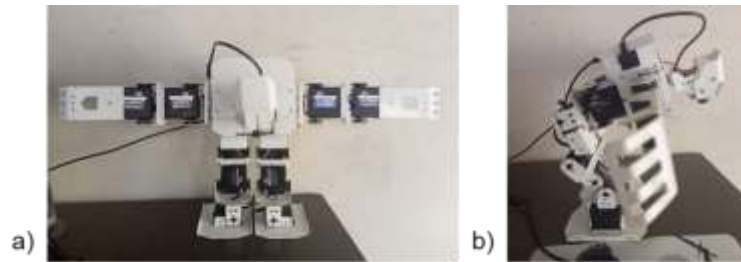
**Movimientos Espaciales.** Para los movimientos espaciales se realiza una parte del código con movimientos tanto de pierna como de cadera por parte del Robot. Evitando así la inestabilidad al tratar de mover cada uno de los Servomotores que lo conforman.

**Agachar al Robot.** En este caso se realizó un conjunto de movimientos para que el cuerpo baje a recoger el objeto sin que se muevan las manos o interfiera en ellas como se muestra en la **Figura 78**.

```
import hiwonder.Board as Board
Board.setBusServoPulse(1, 500, 1500)
Board.setBusServoPulse(2, 700, 1500)
Board.setBusServoPulse(3, 150, 1500)
Board.setBusServoPulse(4, 800, 1500)
Board.setBusServoPulse(5, 500, 1500)
Board.setBusServoPulse(9, 500, 1500)
Board.setBusServoPulse(10, 300, 1500)
Board.setBusServoPulse(11, 850, 1500)
Board.setBusServoPulse(12, 200, 1500)
Board.setBusServoPulse(13, 500, 1500)
Board.setBusServoPulse(17, 500, 1500)
Board.setBusServoPulse(18, 500, 1500)
```

## Figura 78

Robot agachado. a) Vista frontal b) Vista lateral



**Levantarse al Robot.** En este caso se realizó un conjunto de movimientos para que el cuerpo suba sin que se muevan las manos o interfiera en ellas como se muestra en la **Figura 79**.

```
import hiwonder.Board as Board
Board.setBusServoPulse(1, 500, 1000)
Board.setBusServoPulse(2, 388, 1000)
Board.setBusServoPulse(3, 500, 1000)
Board.setBusServoPulse(4, 550, 1000)
Board.setBusServoPulse(5, 500, 1000)
Board.setBusServoPulse(9, 500, 1000)
Board.setBusServoPulse(10, 612, 1000)
Board.setBusServoPulse(11, 500, 1000)
Board.setBusServoPulse(12, 450, 1000)
Board.setBusServoPulse(13, 500, 1000)
Board.setBusServoPulse(17, 500, 1000)
Board.setBusServoPulse(18, 500, 1000)
```

## Figura 79

Vista frontal del código para levantar al Robot.



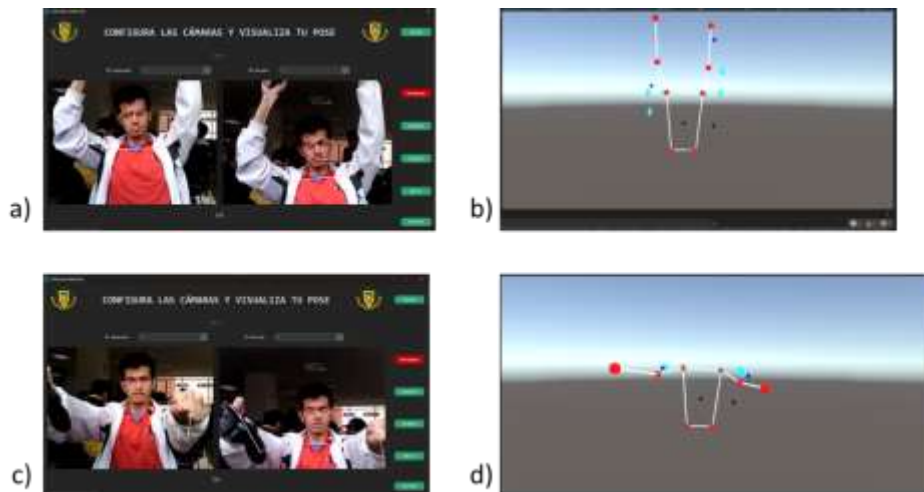
**Movimientos Gestuales.** Para las pruebas de movimiento gestual (**Figura 80**), se buscó que replique el movimiento de los brazos, primero alzándolos, luego estirándolos al



frente. El robot primero se ubica erróneamente a la posición de gesto que se esperaba; cuando se alzaron los brazos, uno bajó y luego subió; cuando se estiraron los brazos, uno se movió para el otro lado al que debería estar como se muestra en la **Figura 81**.

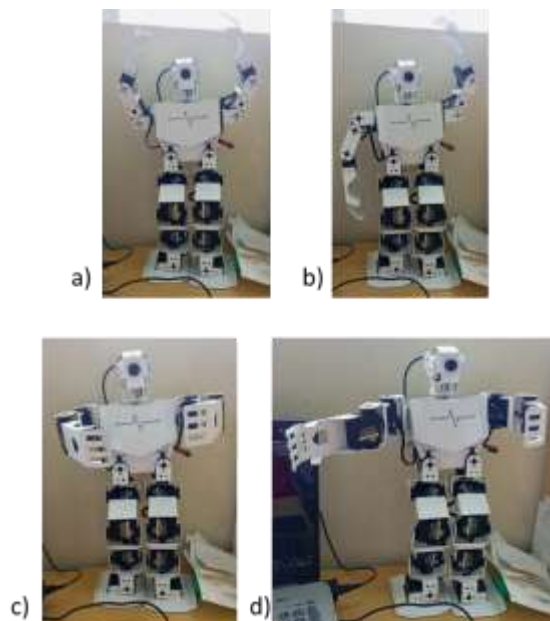
### Figura 80.

Resultado entre PyCharm y Unity. a) PyCharm con los brazos alzados. b) Unity con los brazos alzados. c) PyCharm con los brazos estirados. d) Unity con los brazos estirados.



### Figura 81

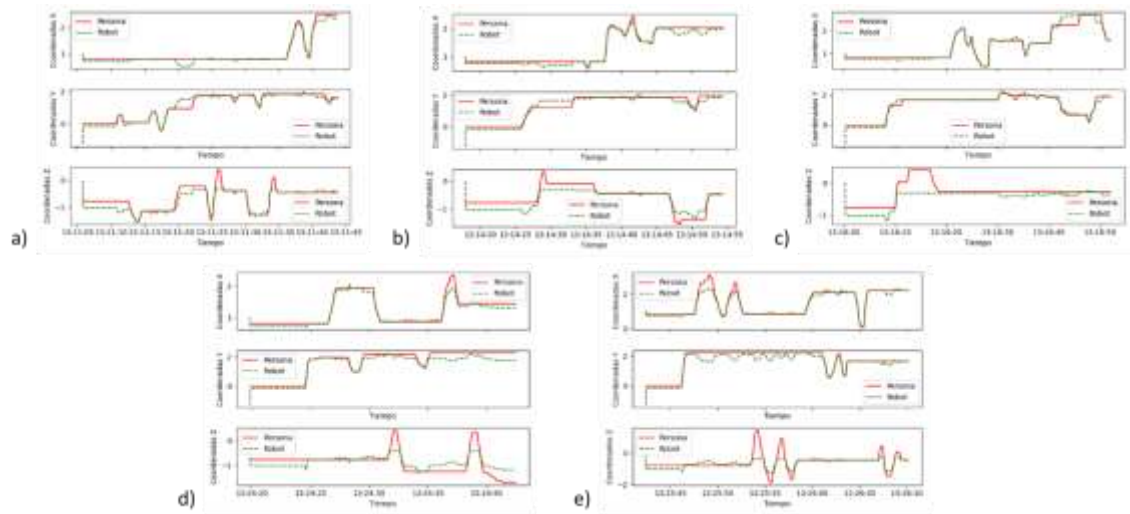
Resultados del robot TonyPi Pro. a) Correcto funcionamiento con los brazos estirados b) Incorrecto funcionamiento con los brazos alzados. c) Correcto funcionamiento con los brazos estirados d) Incorrecto funcionamiento con los brazos estirados.



Estos gestos observados en la **Figura 82** y la **Figura 83** en conjunto, da como resultado el seguimiento en las coordenadas X, Y y Z como se puede observar en la **Figura 82** y la **Figura 83** tanto para el robot como para la persona detectada.

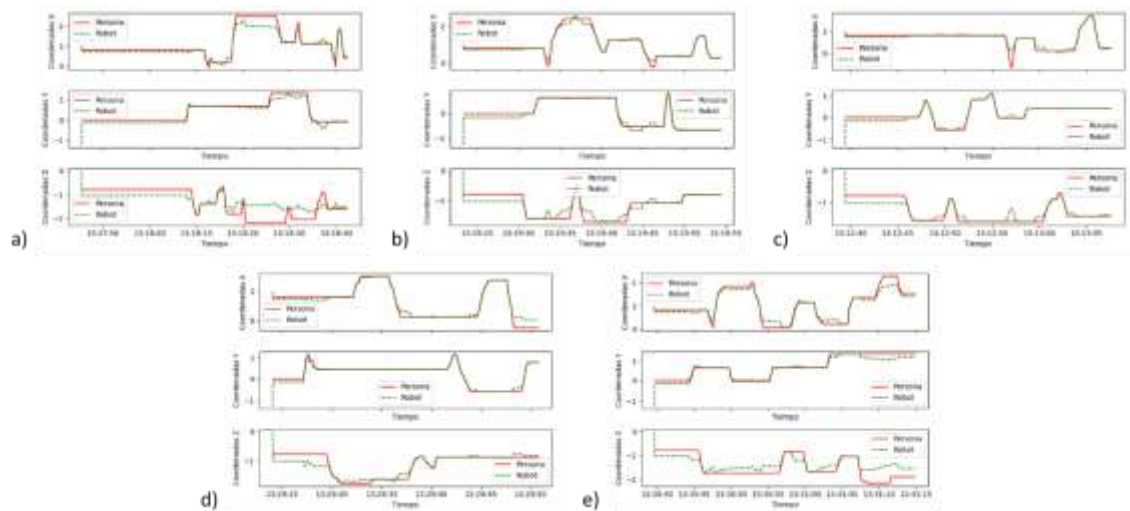
### Figura 82

*Comparación de coordenadas para la muñeca izquierda. a) Primera iteración b) Segunda iteración c) Tercera iteración d) Cuarta iteración e) Quinta iteración*



### Figura 83

*Comparación de coordenadas para la muñeca derecha. a) Primera iteración b) Segunda iteración c) Tercera iteración d) Cuarta iteración e) Quinta iteración*



**Interacción Dentro del Entorno.** Para los movimientos gestuales se busca que interactúe con el entorno levantando un objeto, en este caso una caja de cartón. Se tiene la

detección del gesto de abrir los brazos como se muestra en la **Figura 84** y el de cerrar los brazos como se muestra en la **Figura 85**.

### Figura 84

Gesto para abrir los brazos del robot. a) PyCharm b) Unity



### Figura 85

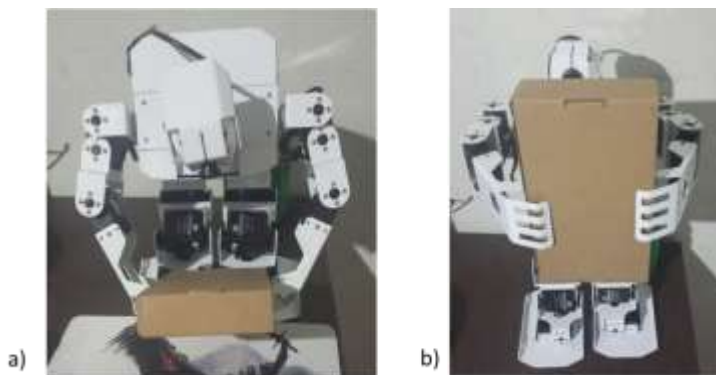
Gesto para cerrar los brazos del robot. a) PyCharm b) Unity



El robot se mueve según los eventos realizados por el operador con el mando, para luego con los gestos lograr agarrar el objeto o soltarlo como se muestra en la **Figura 86**.

### Figura 86

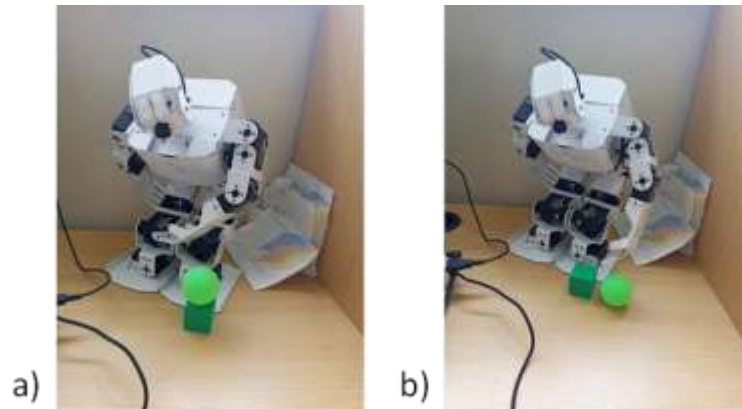
Robot agarrando una caja de cartón. a) Agachado b) Levantado



El robot puede interactuar con el entorno empujando objetos que se encuentren en su rango de acción, como se muestra en la **Figura 87**.

### Figura 87

Robot empujando un objeto. a) Antes de empujarlo b) Después de empujarlo



### Validación de la Hipótesis

Para verificar las distintas pruebas utilizando el método de validación probabilístico chi cuadrado, se busca determinar si la hipótesis planteada en el estudio es respaldada o rechazada. Esto involucra evaluar tanto la hipótesis alternativa H1 como la hipótesis nula H0.

**H1:** ¿Se podrá controlar remotamente un robot humanoide que permita interactuar y manipular objetos dentro de un entorno, mediante visión estéreo y teleoperación, en el laboratorio de Mecatrónica?

**H0:** No se podrá controlar remotamente un robot humanoide que permita interactuar y manipular objetos dentro de un entorno, mediante visión estéreo y teleoperación, en el laboratorio de Mecatrónica.

Para validar la hipótesis, se busca interactuar y manipular distintos objetos siendo una esfera pequeña, un cubo pequeño, una caja de cartón, una caja de plástico y un paquete pequeño de aperitivo. El robot deberá mover al objeto desde un punto A hasta un punto B obteniendo el tiempo que demora y de forma acumulativa si se pudo o no realizar la actividad, como se muestra en la **Figura 88**.

### Figura 88

*Pruebas para interactuar y manipular objetos. Siendo 1 al 4 la esfera pequeña, el cubo pequeño, la caja de cartón, la caja de plástico respectivamente. a) Punto A o inicio de la prueba. b) Punto medio de manipulación del objeto. c) Resultado capturado con respecto al punto B de llegada. d) Punto medio de interacción con el objeto.*



Para obtener los valores de la **Tabla 27**, se realizó la interacción entre el robot y 5 objetos diferentes, moviéndolos desde un punto A hasta un punto B, repitiendo 5 veces la actividad con cada uno de los objetos. La primera columna de resultados indica la cantidad de veces que el robot pudo llevar el objeto al destino (punto B). La segunda columna de resultados indica la cantidad de veces que el robot no pudo llevar el objeto al destino. La columna de

totales es la sumatoria de las iteraciones o veces que se realizó la prueba para el objeto correspondiente. La fila de totales es la sumatoria de las iteraciones que si llegaron al destino y no llegaron al destino correspondientemente. La última columna de resultados muestra el tiempo medio en minutos que se demoró el robot en llevar al objeto desde el punto A al punto B.

**Tabla 27**

*Tabla de resultados obtenidos por cada objeto, en 5 iteraciones, por interacción.*

	Si Llegó Al Destino	No Llegó Al Destino	Total	Tiempo (min)
<b>Objeto 1</b>	1	4	5	23:15
<b>Objeto 2</b>	3	2	5	18:34
<b>Objeto 3</b>	4	1	5	14:57
<b>Objeto 4</b>	4	1	5	16:38
<b>Objeto 5</b>	4	1	5	18:47
<b>Total</b>	16	9	25	

Para la **Tabla 27**, las frecuencias teóricas están dadas por la Ecuación (38), entonces se tiene que:

$$ft = \frac{(Total\ de\ columna) * (Total\ de\ filas)}{Total} \quad (38)$$

$$ft_{Objeto_{iSi}} = \frac{(16)(5)}{25} = 2.16$$

$$ft_{Objeto_{iNo}} = \frac{(9)(5)}{25} = 1.8$$

Se determina el grado de libertad tomando en cuenta la Ecuación (39):

$$V = (\#de\ filas - 1) * (\#de\ columnas - 1) \quad (39)$$

$$V = (5 - 1)(2 - 1) = 4$$

Se calcula el chi cuadrado con la Ecuación (40), teniendo:

$$x^2 = \sum \frac{(f - ft)^2}{ft} \quad (40)$$



$$x^2 = \frac{(1 - 2.16)^2}{2.16} + \frac{(4 - 1.8)^2}{1.8} + \frac{(3 - 2.16)^2}{2.16} + \frac{(2 - 1.8)^2}{1.8} + \frac{(4 - 2.16)^2}{2.16} + \frac{(1 - 1.8)^2}{1.8} + \frac{(4 - 2.16)^2}{2.16} \\ + \frac{(1 - 1.8)^2}{1.8} + \frac{(4 - 2.16)^2}{2.16} + \frac{(1 - 1.8)^2}{1.8}$$

$$x_{calculada}^2 = 9.4296$$

Se encuentra el valor de chi cuadrado crítico mediante la distribución de chi cuadrado de la **Tabla 28** con el grado de libertad de 4 y un nivel de significancia del 0.1.

**Tabla 28**

*Distribución Chi Cuadrado.*

v/p	0,001	0,0025	0,005	0,01	0,025	0,05	0,1	0,15	0,2	0,25	0,3	0,35	0,4	0,45	0,5
1	10,8274	9,1404	7,8794	6,6349	5,0239	3,8415	2,7055	2,0722	1,6424	1,3233	1,0742	0,8735	0,7083	0,5707	0,4549
2	13,8150	11,9827	10,5965	9,2104	7,3778	5,9915	4,6052	3,7942	3,2189	2,7726	2,4079	2,0996	1,8326	1,5970	1,3863
3	16,2660	14,3202	12,8381	11,3449	9,3484	7,8147	6,2514	5,3170	4,6416	4,1083	3,6649	3,2831	2,9462	2,6430	2,3660
4	18,4662	16,4238	14,8602	13,2767	11,1433	9,4877	7,7794	6,7449	5,9886	5,3853	4,8784	4,4377	4,0446	3,6871	3,3567
5	20,5147	18,3854	16,7496	15,0863	12,8325	11,0705	9,2363	8,1152	7,2893	6,6257	6,0644	5,5731	5,1319	4,7278	4,3515
6	22,4575	20,2491	18,5475	16,8119	14,4494	12,5916	10,6446	9,4461	8,5581	7,8408	7,2311	6,6948	6,2108	5,7652	5,3481

Entonces, para  $V = 4$  y  $p = 0.1$ , se tiene que:

$$x_{critico}^2 = 7.7794$$

De los valores de chi encontrados, se los compara para establecer cuál de las hipótesis cumple. Para que la hipótesis alternativa (H0) cumpla, el valor del chi cuadrado calculado debe ser mayor al chi cuadrado crítico; caso contrario, la hipótesis nula (H1) cumple.

$$x_{calculada}^2 > x_{critico}^2$$

$$9.4296 > 7.7794$$

Para obtener los valores de la **Tabla 29**, se realizó la manipulación entre el robot y 5 objetos diferentes (los mismos que se usó para la interacción), agarrándolos, moviéndolos desde un punto A hasta un punto B para terminar soltándolos, repitiendo 5 veces la actividad con cada uno de los objetos. La primera columna de resultados indica la cantidad de veces que el robot pudo llevar el objeto al destino (punto B). La segunda columna de resultados indica la cantidad de veces que el robot no pudo llevar el objeto al destino. La columna de totales es la sumatoria de las iteraciones o veces que se realizó la prueba para el objeto correspondiente.

La fila de totales es la sumatoria de las iteraciones que si llegaron al destino y no llegaron al destino correspondientemente. La última columna de resultados muestra el tiempo medio en minutos que se demoró el robot en llevar al objeto desde el punto A al punto B.

**Tabla 29**

*Tabla de resultados obtenidos por cada objeto, en 5 iteraciones, por manipulación.*

	Si llegó al destino	No llegó al destino	Total	Tiempo (min)
<b>Objeto 1</b>	1	4	5	16:51
<b>Objeto 2</b>	2	3	5	15:47
<b>Objeto 3</b>	5	0	5	10:24
<b>Objeto 4</b>	4	1	5	13:19
<b>Objeto 5</b>	3	2	5	13:75
<b>Total</b>	15	10	25	

De igual forma para la  $y$  con la Ecuación (38), las frecuencias teóricas son:

$$ft_{Objeto_{iSi}} = \frac{(15)(5)}{25} = 3$$

$$ft_{Objeto_{iNo}} = \frac{(10)(5)}{25} = 2$$

Se determina el grado de libertad tomando en cuenta la Ecuación (39):

$$V = (5 - 1)(2 - 1) = 4$$

Se calcula el chi cuadrado con la Ecuación (40), teniendo:

$$x^2 = \frac{(1-3)^2}{3} + \frac{(4-2)^2}{2} + \frac{(2-3)^2}{3} + \frac{(3-2)^2}{2} + \frac{(5-3)^2}{3} + \frac{(0-2)^2}{2} + \frac{(4-3)^2}{3} + \frac{(1-2)^2}{2} + \frac{(3-3)^2}{3} + \frac{(2-2)^2}{2}$$

$$x_{calculada}^2 = 8.3333$$

Entonces, para  $V = 4$  y  $p = 0.1$ , se tiene que:

$$x_{critico}^2 = 7.7794$$

De los valores de chi encontrados, se los compara para establecer cuál de las hipótesis cumple. Para que la hipótesis alternativa ( $H_0$ ) cumpla, el valor del chi cuadrado calculado debe ser mayor al chi cuadrado crítico; caso contrario, la hipótesis nula ( $H_1$ ) cumple.



$$x_{calculada}^2 > x_{critico}^2$$

$$8.3333 > 7.7794$$

Como ambas pruebas cumplen las condiciones de que el valor calculado sea mayor al crítico, se puede deducir que, si se puede controlar remotamente un robot humanoide que permita interactuar y manipular objetos dentro de un entorno, mediante Visión Estéreo y Teleoperación, en el laboratorio de Mecatrónica.

## Capítulo V

### Conclusiones y Recomendaciones

#### Conclusiones

- El modelo de cámara seleccionado para el dispositivo de Visión Estéreo es la Genius FaceCam 1000X, utilizando dos cámaras de las mismas características (modelo) y así mantener la paridad y perspectiva a  $63mm$  de separación entre ellas.
- El soporte de las cámaras para el dispositivo de Visión Estéreo, para las cargas aplicadas por estas cámaras y la manipulación del operador, tiene un desplazamiento máximo de  $0.38mm$  con un coeficiente de seguridad de  $1.03$  unidades.
- El robot TonyPi Pro posee una Unidad de Procesamiento de modelo Raspberry Pi 4B que permite la comunicación entre este y varios dispositivos de computadoras o celulares inteligentes.
- Las librerías creadas pretenden eliminar o disminuir los códigos repetitivos para que, al revisar o editar todo el código, este sea más legible y no confunda entre segmentos de código.
- La distancia entre la cámara y el punto de referencia de la persona observada para un mejor cálculo de la triangulación de la profundidad se encuentra entre los  $111cm$  y los  $250cm$  ya que el punto que se tomó como referencia para la medida entra en el rango de observación en la imagen.
- La latencia media, calculada a partir de enviar y recibir un mensaje específico 100 veces, en la comunicación entre PyCharm y Unity es de  $0.19ms$  debido a que se encuentran en el mismo dispositivo (computador).
- La latencia media, calculada a partir de enviar y recibir un mensaje específico 100 veces, en la comunicación entre Unity y el SO de la Raspberry es de  $18.16ms$  debido a que se encuentran en diferentes dispositivos (computador y robot respectivamente).

- El error acumulativo entre los Servomotores para el brazo izquierdo es de 3.7% dependiente al redondeo de los ángulos recibidos y límites, haciendo que, al calcular y enviar el valor del Servomotor correspondiente, el resultado sea diferente al ideal.
- Si se puede controlar remotamente un robot humanoide permitiendo interactuar y manipular objetos dentro de un entorno, mediante Visión Estéreo y Teleoperación en el Laboratorio de Mecatrónica.

## Recomendaciones

- Utilizar una cámara que ya de un mapeo de profundidad en tiempo real o una cámara de Visión Estéreo comercial, para obtener mejores resultados en la obtención de la coordenada Z de profundidad.
- Diseñar un soporte universal para las cámaras web o permitir que la distancia entre las cámaras pueda ser variable para mayores pruebas y resultados que favorezcan en el estudio del proyecto.
- Leer de forma minuciosa la documentación del robot para un buen uso, además de verificar la información con los foros que existen en internet para evitar errores o falta de información.
- Usar un solo tipo de nomenclatura para diferenciar entre variables de funciones y variables del programa, de esta forma es posible añadiendo un carácter más, el usar los mismos nombres para las constantes y variables.
- Para obtener valores precisos de profundidad para cada punto de CVZone, es crucial asegurarse de que estos puntos estén ubicados dentro del área de trabajo designada.
- Tener una CPU de mayor rendimiento en la computadora para agilizar la comunicación dentro del mismo dispositivo, lo que resultará en una transferencia más rápida de datos entre dispositivos o programas y evitará retrasos en el envío y recepción de información.
- Crear una red externa a la que se conecten el robot y la computadora, de esta forma sería posible el hacer uso de internet para consultas del proyecto como configuración y manejo de estos.

## Bibliografía

- Alberto, L. M. (2014). *Análisis de movimiento en Mecanismos Cinemática Directa e Inversa*. Bogotá DC, Colombia: Facultad de Ingeniería y Ciencias Básicas, Programa Ingeniería Industrial, Politécnico Grancolombiano.
- Andrezero. (2023). *Mercado Libre*. Obtenido de [https://www.mercadolibre.com.ec/camara-web-logitech-c920e-full-hd-30fps-color-negro/p/MEC18926944?pdp\\_filters=category:MEC1648#searchVariation=MEC18926944&position=21&search\\_layout=stack&type=product&tracking\\_id=74f574bb-14fa-495a-8bb7-5ce32c8c7320](https://www.mercadolibre.com.ec/camara-web-logitech-c920e-full-hd-30fps-color-negro/p/MEC18926944?pdp_filters=category:MEC1648#searchVariation=MEC18926944&position=21&search_layout=stack&type=product&tracking_id=74f574bb-14fa-495a-8bb7-5ce32c8c7320)
- Ariyanto, Y. (2015). *IMPLEMENTASI REMOTE DESKTOP KOMPUTER MENGGUNAKAN VIRTUAL NETWORK COMPUTING (VNC) SERVER DAN VNC VIEWER BERBASIS ANDROID*. Politeknik Negeri Malang.
- Association, A. N. (1999). *ANSI/RIA R15.07-1: Safety standard for industrial robots and robot systems - Part 1: General safety requirements*. Estados Unidos: American National Standards Institute/Robotic Industries Association (ANSI/RIA).
- Atom. (30 de Enero de 2018). *Introduction - OpenCV Tutorial C++*. Obtenido de <https://www.opencv-srf.com/p/introduction.html>
- Basulto Lantsova, A. (2020). *ResearchGate*. Obtenido de [https://www.researchgate.net/profile/Alejandro\\_Barranco\\_Gutierrez5/publication/349683067\\_Emparejamiento\\_de\\_puntos\\_en\\_triangulacion\\_multicamara\\_utilizando\\_DIC/links/603c5abe299bf1cc26fbd13b/Emparejamiento-de-puntos-en-triangulacion-multicamara-utilizando-D](https://www.researchgate.net/profile/Alejandro_Barranco_Gutierrez5/publication/349683067_Emparejamiento_de_puntos_en_triangulacion_multicamara_utilizando_DIC/links/603c5abe299bf1cc26fbd13b/Emparejamiento-de-puntos-en-triangulacion-multicamara-utilizando-D)
- BeagleBoard.org. (2022). *BeagleBone Black*. Obtenido de <https://www.beagleboard.org/boards/beaglebone-black>
- Bhatti, A. (2008). *ResearchGate*. Obtenido de [https://www.researchgate.net/publication/236455177\\_Stereo\\_Vision](https://www.researchgate.net/publication/236455177_Stereo_Vision)
- Bhatti, A. (2020). *ResearchGate*. Obtenido de [https://www.researchgate.net/figure/Stereo-vision-disparity\\_fig1\\_236455177](https://www.researchgate.net/figure/Stereo-vision-disparity_fig1_236455177)
- Chimansky, T. (2023). *Custom Tkinter*. Obtenido de <https://customtkinter.tomschimansky.com>
- Components, S. (2023). *Amazon*. Obtenido de [https://www.amazon.com/-/es/NUEVO-BEAGLEBONE-azabache-30-segundos-Componentes/dp/B00HSE2SDI/ref=sr\\_1\\_21?adgrpid=80482250806&hvadid=585362632545&hvdev=c&hvlocphy=9069556&hvnetw=g&hvqmt=b&hvrnd=15067444374204157964&hvtargid=kwd-50098007161&hydadcr=22306\\_13](https://www.amazon.com/-/es/NUEVO-BEAGLEBONE-azabache-30-segundos-Componentes/dp/B00HSE2SDI/ref=sr_1_21?adgrpid=80482250806&hvadid=585362632545&hvdev=c&hvlocphy=9069556&hvnetw=g&hvqmt=b&hvrnd=15067444374204157964&hvtargid=kwd-50098007161&hydadcr=22306_13)
- CVZone. (20 de Marzo de 2023). *CVZone - Pose Estimation*. Obtenido de <https://www.computervision.zone/courses/pose-estimation/>
- DJANGO. (2023). *Python Tutorial*. Obtenido de <https://www.pythontutorial.net/tkinter/tkinter-grid/>

- Electrónica, N. (2023). *Mercado Libre*. Obtenido de [https://articulo.mercadolibre.com.ec/MEC-517547244-camara-web-speedmind-full-hd1920x1080-microfonowindowps-10-\\_JM#position=53&search\\_layout=stack&type=item&tracking\\_id=74f574bb-14fa-495a-8bb7-5ce32c8c7320](https://articulo.mercadolibre.com.ec/MEC-517547244-camara-web-speedmind-full-hd1920x1080-microfonowindowps-10-_JM#position=53&search_layout=stack&type=item&tracking_id=74f574bb-14fa-495a-8bb7-5ce32c8c7320)
- Espinoza Castro, M. A., & Nasimba Nasimba, V. F. (2021). *Repositorio ESPE*. Obtenido de <https://repositorio.espe.edu.ec/bitstream/21000/27122/1/T-ESPE-017375.pdf>
- Falomir Llansola, Z. (2006). *Repositorio UJI*. Obtenido de [http://repositori.uji.es/xmlui/bitstream/handle/10234/27826/ICC\\_2006-10-03.pdf?sequence=1&isAllowed=y](http://repositori.uji.es/xmlui/bitstream/handle/10234/27826/ICC_2006-10-03.pdf?sequence=1&isAllowed=y)
- Foundation, P. S. (20 de Julio de 2023). *socket — Low-level networking interface*. Obtenido de <https://docs.python.org/3/library/socket.html>
- GeekPi. (2023). *Amazon*. Obtenido de [https://www.amazon.com/GeeekPi-Raspberry-Kit-inicio-alimentación/dp/B0B3M2HKN6/ref=sr\\_1\\_1\\_ssapa?adgrpid=83052647738&hvadid=585362632932&hvdev=c&hvlocphy=9069556&hvnetw=g&hvqmt=b&hvrnd=9042516198131848838&hvtargid=kwd-972466948035&hydadcr=22333\\_13333537&ke](https://www.amazon.com/GeeekPi-Raspberry-Kit-inicio-alimentación/dp/B0B3M2HKN6/ref=sr_1_1_ssapa?adgrpid=83052647738&hvadid=585362632932&hvdev=c&hvlocphy=9069556&hvnetw=g&hvqmt=b&hvrnd=9042516198131848838&hvtargid=kwd-972466948035&hydadcr=22333_13333537&ke)
- GUÉRIN, B.-A. (Abril de 2018). *eBook*. Obtenido de <https://books.google.es/books?hl=es&lr=&id=y361soCvNvsC&oi=fnd&pg=PA172&dq=c%23&ots=wBirZiY300&sig=Jw-9s1LwsoR4HT4FFu85bJtMego#v=onepage&q=c%23&f=false>
- Hiwonder. (2023). *Hiwonder - Home*. Obtenido de <https://www.hiwonder.com>
- Hunt, J. (2019). *Advanced Guide to Python 3 Programming*. Springer. Obtenido de <https://realpython.com/python-sockets/>
- Inga Ortiz, R. S., & Zoria Ortega, J. C. (2018). *Repositorio ESPE*. Obtenido de <https://repositorio.espe.edu.ec/bitstream/21000/14922/1/T-ESPE-040293.pdf>
- intel. (2023). *intel*. Obtenido de <https://www.intel.la/content/www/xl/es/products/docs/processors/cpu-vs-gpu.html>
- JetBrains. (2023). *PyCharm*. Obtenido de <https://www.jetbrains.com/es-es/pycharm/>
- Makeronics. (2023). *Amazon*. Obtenido de [https://www.amazon.com/Makeronics-Developer-Acrylic-Wireless-Antenna/dp/B07ZKL2BP2/ref=sr\\_1\\_12\\_ssapa?adgrpid=146025314529&hvadid=633569899178&hvdev=c&hvlocphy=9069556&hvnetw=g&hvqmt=b&hvrnd=5413952281575125805&hvtargid=kwd-66340622225&hydadcr=2751\\_135418](https://www.amazon.com/Makeronics-Developer-Acrylic-Wireless-Antenna/dp/B07ZKL2BP2/ref=sr_1_12_ssapa?adgrpid=146025314529&hvadid=633569899178&hvdev=c&hvlocphy=9069556&hvnetw=g&hvqmt=b&hvrnd=5413952281575125805&hvtargid=kwd-66340622225&hydadcr=2751_135418)
- Martínez, Y., Fernández, R. R., Álvarez, D. A., García, M., & Martínez, E. (2014). *SCIELO*. Obtenido de [https://www.scielo.org.mx/scielo.php?script=sci\\_arttext&pid=S2007-40182014000300007](https://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S2007-40182014000300007)
- Meccano/Erector. (2023). *Meccano*. Obtenido de [https://www.meccano.com/en\\_gb/products/778988112878](https://www.meccano.com/en_gb/products/778988112878)
- Microsoft. (11 de Abril de 2022). *Unity Development with VS Code*. Obtenido de <https://code.visualstudio.com/docs/other/unity>

- Nandar Carvajal, J. V. (2022). *Repositorio Escuela Politécnica Nacional*. Obtenido de <https://bibdigital.epn.edu.ec/bitstream/15000/22544/1/CD%2011949.pdf>
- Nuño Ortega, E., & Basañez Villaluenga, L. (2004). *Universidad Politécnica de Catalunya*. Obtenido de <https://upcommons.upc.edu/bitstream/handle/2117/570/IOC-DT-P-200405.pdf?sequence=1>
- NVIDIA. (2023). *NVIDIA Jetson Nano*. Obtenido de <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/>
- Ñacato Estrella, R. D. (2014). *Repositorio ESPOCH*. Obtenido de <http://dspace.esepoch.edu.ec/bitstream/123456789/3622/1/108T0106.pdf>
- Ortiz Morales, G. X., Garrido Vázquez, J. N., Hernández Cadena, A., Jesús Magaña, J. Á., Gómez Zea, J. M., & León de la O, D. M. (Diciembre de 2020). *Acercamiento a la robótica: Robot humanoide*. Obtenido de [https://iydt.files.wordpress.com/2021/01/4\\_5\\_acercamiento-a-la-robotica-robot-humanoide\\_v.pdf](https://iydt.files.wordpress.com/2021/01/4_5_acercamiento-a-la-robotica-robot-humanoide_v.pdf)
- Padilla, M. (2018). *Catarina*. Obtenido de [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lmt/padilla\\_m\\_o/capitulo2.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lmt/padilla_m_o/capitulo2.pdf)
- PSF, & Foundation, P. S. (2023). *Python - About*. Obtenido de <https://www.python.org/about/>
- Python. (19 de Julio de 2023). *Tkinter message prompts*. Obtenido de <https://docs.python.org/3/library/tkinter.messagebox.html>
- Raspberry. (2023). *Raspberry Pi 4 Modelo B*. Obtenido de <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>
- Rodríguez Antón, J. (2015). *Blog Web*. Obtenido de <https://josefelixrodriguezantonweb.com/2015/09/26/la-vision-estereoscopica-avance-evolutivo-en-el-espacio/>
- Rodriguez, M. (2023). *INESEM - Business School*. Obtenido de <https://www.inesem.es/revistadigital/gestion-integrada/diferencia-robotica-grados-libertad-movilidad-3/#:~:text=Entre%20los%20grados%20de%20libertad,torno%20a%20su%20propio%20eje.>
- Sanchis Reig, A. (2022). *Repositorio Universidad de Alicante*. Obtenido de [https://rua.ua.es/dspace/bitstream/10045/124652/1/Interfaz\\_hombremaquina\\_mediante\\_estimacion\\_de\\_pose\\_Sanchis\\_Reig\\_Adrian.pdf](https://rua.ua.es/dspace/bitstream/10045/124652/1/Interfaz_hombremaquina_mediante_estimacion_de_pose_Sanchis_Reig_Adrian.pdf)
- Segundo Cruz, P. (2022). *Scribd*. Obtenido de <https://es.scribd.com/document/350842602/Grados-de-Libertad>
- Sidek, S. Z., Ramli, M. B., & Saad, S. N. (2020). *PHYTON GUI Programming with Tkinter*. Politeknik. Obtenido de <http://alexandre.benoit.83.free.fr/archives/1415/adumas/ISN/tkinter.pdf>
- Standardization, I. O. (1997). *ISO 9286: Robots and robotic devices - Industrial robots - Controllers*. Ginebra, Suiza: International Organization for Standardization (ISO).

Standardization, I. O. (1998). *ISO 9238: Robots and robotic devices - Industrial robots - Performance criteria and related test methods*. Ginebra, Suiza: International Organization for Standardization (ISO).

Standardization, I. O. (2011). *ISO 10218-1: Robots and robotic devices - Safety requirements for industrial robots - Part 1: Robots*. Ginebra, Suiza: International Organization for Standardization (ISO).

Technologies, U. (2023). *Unity*. Obtenido de <https://unity.com/es>

Tractus3D. (2020). *Filamento ABS*. Obtenido de <https://tractus3d.com/es/materials/abs/>

Tractus3D. (2020). *Filamento PLA*. Obtenido de <https://tractus3d.com/es/materials/pla/>

*Ubttech*. (2023). Obtenido de <http://robotsmexico.mx/Alpha/index.html#>

Vinicio. (2023). *Mercado Libre*. Obtenido de [https://articulo.mercadolibre.com.ec/MEC-519162195-camara-web-genius-face-cam-1000x-hd-720p-13mp-microfono-usb-\\_JM#position=31&search\\_layout=stack&type=item&tracking\\_id=74f574bb-14fa-495a-8bb7-5ce32c8c7320](https://articulo.mercadolibre.com.ec/MEC-519162195-camara-web-genius-face-cam-1000x-hd-720p-13mp-microfono-usb-_JM#position=31&search_layout=stack&type=item&tracking_id=74f574bb-14fa-495a-8bb7-5ce32c8c7320)

Y. Liu, C. L. (2021). *Stereo Vision-Based Object Tracking with Pose Estimation*. *Sensors*. doi:<https://doi.org/10.3390/s21051653>

Yudi. (2023). *Etsy*. Obtenido de <https://www.etsy.com/mx/listing/700395601/escultura-de-leon-de-madera-estatua>

Zone, C. V. (20 de Jul de 2023). *Github - CVZone*. Obtenido de <https://github.com/cvzone>



## Anexos