



UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
CARRERA DE ELECTRÓNICA Y AUTOMATIZACIÓN

TRABAJO DE TITULACIÓN

Propuesta tecnológica, previo a la obtención del título de:

INGENIERA EN ELECTRÓNICA Y AUTOMATIZACIÓN

**“DESARROLLO DE UN SISTEMA DE IMITACIÓN DE MOVIMIENTOS
BASADO EN VISIÓN ARTIFICIAL PARA EL CONTROL DE UN BRAZO
ROBÓTICO”**

AUTORES

MARCO VINICIO ESPINOZA ARROYO.

JOSUE ENRIQUE VERA MORALES.

PROFESOR TUTOR

ING. JUNIOR RAFAEL FIGUEROA OLMEDO, MGT.

LA LIBERTAD–ECUADOR

2024

AGRADECIMIENTO

Agradezco primero a Dios por darme vida y la oportunidad de poder estudiar esta carrera, por siempre iluminarme de conocimientos y experiencias necesarias en la universidad para poder ser un profesional.

A mi familia por apoyarme en cada uno de los proyectos en cada semestre, a mi padre Marco Espinoza y a mi madre Katia Arroyo, por a pesar de las circunstancias y adversidades siempre buscar una solución para apoyarme en mis estudios.

A mis hermanas por inspirarme a mejorar y apoyarme también a ser alguien mejor siempre.

A mi colega de tesis Josué Vera, por su constancia de trabajar en conjunto para que esta tesis se llevara a cargo con éxito.

A mis amistades por su apoyo y ayuda también en permitir que los componentes necesarios se obtuvieran sin mayor contratiempo y también por los ánimos que me han dado muchas veces.

A mi tutor, el Ing. Junior Figueroa, por su paciencia y conocimientos que nos brindó en cada una de las tutorías y a los demás docentes que he tenido a lo largo de la carrera universitaria.

Marco Vinicio Espinoza Arroyo

AGRADECIMIENTO

Agradezco a Dios por la vida y salud que me a otorgado para cumplir mis metas y poder culminar con éxito la etapa de mi carrera.

A mis padres, Grace Morales Tomalá y Juan Vera Tigrero y familiares quienes me han apoyado en cada momento. Gracias por creer en mí y brindarme fuerzas para seguir adelante.

A mis amistades, que han estado presente en los momentos más difíciles, ofreciendo su apoyo. quienes fueron dispensable para que no desistiera en este camino hacia convertirnos en profesionales.

Al ingeniero Junior Figueroa, mi tutor de tesis, por su valiosa orientación y experiencia, que me brindo durante el desarrollo de esta tesis. Su guía y paciencia fueron fundamentales para culminar este proceso académico.

Josué Enrique Vera Morales

DEDICATORIA

Esta tesis la dedico a mi padre Marco Espinoza y a mi madre Katia Arroyo que con su esfuerzo me han apoyado a lo largo de mis estudios. Gracias por su paciencia y su ejemplo de no rendirme, el esforzarme en mis deberes por hacer y ser un mejor hombre y tener mi profesión.

Marco Vinicio Espinoza Arroyo

DEDICATORIA

Mi tesis se la dedico a las siguientes personas:

A mi madre Grace Morales Tomalá, quien me dio la vida y ha sido mi fuente inspiración, fortaleza y guía, gracias a sus sabios consejos, que me han mantenido en el buen camino, sus sacrificios que me ha permitido culminar mis estudios, Por siempre creer en mí y ayudarme a cumplir mis metas.

A mi padre Juan Vera tigrero, por enseñarme el valor de la perseverancia y honesto. Tus sabios consejos que me han permitido afrontar los desafíos que da la vida.

A Venancio Quiroz, quien me apoyo durante mi fase final de mis estudios convirtiéndose en un padre y amigo en el que se puede confiar.

A mis hermanos Mike, Luis y Brithany quienes han estado en los buenos y malos momentos, demostrando el significado de ser familia. A pesar de nuestros defectos y desacuerdo, siempre permaneceremos unidos.

A mis abuelos José, Julio, Justina quienes me brindaron su apoyo y me tratan como si fuera su hijo. En especial, a mi abuela Rosa, que, aunque ya no se encuentra entre nosotros, me enseñó a ser bondadoso y el valor de siempre ayudar a los demás.

Y a mis amigos Laura Vásconez, Carlos mora y Bryan Sánchez, que a pesar de los desacuerdos que pudimos tener, me han brindaron su apoyo y amistad durante este proceso. Compañeros a los que puedo llamar hermanos, cuya presencia en este camino ha sido un regalo inolvidable en nuestra formación como profesionales.

Josué Enrique Vera Morales

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del trabajo de integración curricular denominado: “DESARROLLO DE UN SISTEMA DE IMITACIÓN DE MOVIMIENTOS BASADO EN VISIÓN ARTIFICIAL PARA EL CONTROL DE UN BRAZO ROBÓTICO”, elaborado por los estudiantes Espinoza Arroyo Marco Vinicio y Vera Morales Josué Enrique, de la Carrera de Electrónica y Automatización de la Facultad de Sistemas y Telecomunicaciones de la Universidad Estatal Península de Santa Elena, me permito declarar que luego de haber orientado, estudiado y revisado, lo apruebo en todas sus partes y autorizo a los estudiantes para que inicien los trámites legales correspondientes.

La Libertad, 29 de noviembre del 2024

A handwritten signature in black ink, appearing to read 'Junior Figueroa', is written over a horizontal line. The signature is stylized and somewhat cursive.

Ing. Junior Figueroa Olmedo, Msc.

DOCENTE TUTOR

TRIBUNAL DE SUSTENTACIÓN



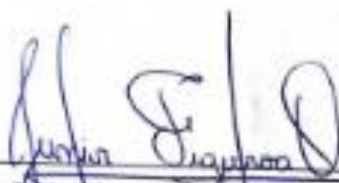
Ing. Ronald Rovira Jurado, Ph.D.

DIRECTOR DE LA CARRERA DE
ELECTRÓNICA Y AUTOMATIZACIÓN



Ing. Edison Coral Salinas, MSc.

DOCENTE ESPECIALISTA



Ing. Junior Figueroa Oviedo, MSc.

DOCENTE TUTOR



Ing. Luis Chuquimarca Jiménez, Mgt.

DOCENTE GUÍA UIC



Ing. Corina Gonzabay-De La A, Mgt.

SECRETARIA

DECLARACIÓN

El contenido del presente trabajo de titulación es de nuestra entera responsabilidad, el patrimonio intelectual del mismo le pertenece a la Universidad Estatal Península de Santa Elena.



Espinoza Arroyo Marco Vinicio

AUTOR



Vera Morales Josué Enrique

AUTOR

RESUMEN

El presente trabajo de titulación se enfoca en el desarrollo de un sistema de imitación de movimientos basado en visión artificial para el control de un brazo robótico.

El proyecto tiene como objetivo el diseñar el algoritmo de control que permita enviar las órdenes a los actuadores de las articulaciones del brazo robótico para ejecutar el proceso de imitación.

La creación e implementación de estos algoritmos de control está basado en principios matemáticos y en el uso de un software de alto nivel, especial para este tipo de tarea. El sistema de imitación permitirá al usuario en situaciones donde su integridad física presente un riesgo significativo.

El objetivo principal es procesar mediante visión artificial las imágenes adquiridas por la cámara Kinect para extraer información relevante tales como el ángulo o posición que permitirán posteriormente detectar la posición de las articulaciones del brazo robótico referente al brazo humano del usuario.

Palabras claves: brazo robótico, imitación, cámara Kinect, articulaciones.

ABSTRACT

This thesis focuses on the development of a motion imitation system based on artificial vision for the control of a robotic arm.

The project aims to design a control algorithm that allows commands to be sent to the actuators of the joints of the robotic arm to execute the imitation process.

The creation and implementation of these control algorithms is based on mathematical principles and the use of high-level software, especially for this type of task. The imitation system will allow the user in situations where their physical integrity presents a significant risk.

The main objective is to process the images acquired by the Kinect camera through artificial vision to extract relevant information such as the angle or position that will later detect the position of the joints of the robotic arm relative to the user's human arm.

Keywords: robotic arm, imitation, Kinect camera, joints.

ÍNDICE DE CONTENIDO

CAPÍTULO I.....	10
1.1 ANTECEDENTES	10
1.2 DESCRIPCIÓN DEL PROYECTO	11
1.3 OBJETIVOS DEL PROYECTO	13
1.3.1 <i>Objetivo General</i>	13
1.3.2 <i>Objetivo Específico</i>	13
1.4 JUSTIFICACIÓN.....	14
1.5 ALCANCE DEL PROYECTO	15
1.6 METODOLOGÍA.....	17
CAPITULO II.....	18
2.1 MARCO CONTEXTUAL	18
2.2 MARCO CONCEPTUAL	19
2.2.1 <i>Brazo humano</i>	19
2.2.2 <i>Brazos robóticos</i>	21
2.2.2.1 Elementos de un brazo robótico.....	21
2.2.2.2 Tipos de brazos robóticos	22
2.2.2.3 Grados de libertad de un robot.....	24
2.2.2.4 Cinemática de un brazo robótico	25
2.2.3 <i>Visión artificial</i>	29
2.2.3.1 Procesamiento de imágenes.....	30
2.2.3.2 Tecnologías utilizadas en cámaras.....	32
2.2.3.3 Lenguajes de programación utilizados en visión artificial.....	33
2.2.4 <i>Sistemas de comunicación vinculados a brazos robóticos</i>	34
2.2.5 <i>Tarjetas controladoras</i>	35
2.3 MARCO TEÓRICO	37

CAPITULO III	39
3.1. COMPONENTES DE LA PROPUESTA	39
3.1.1. COMPONENTES FÍSICOS	39
3.1.1.1. Cámara Kinect Xbox 360° modelo: 414.....	39
3.1.1.2. Arduino UNO	41
3.1.1.3. Brazo robótico X-Arm 1S.....	43
3.1.2. COMPONENTES LÓGICOS	49
3.1.2.1. Entorno de programación LabVIEW	49
3.1.2.2. Entorno de desarrollo integrado Arduino IDE.....	54
3.2. DISEÑO DE LA PROPUESTA.....	57
3.2.1.1. Conexión de la cámara Kinect	59
3.2.1.2. Conexión de dispositivos con el brazo robótico.....	59
3.3. PRUEBAS Y RESULTADOS	85
3.3.1. Pruebas en calibración de cámara Kinect (sensor).....	85
3.3.2. Pruebas en grados formados por las articulaciones de los brazos	88
CAPITULO IV.....	99
4.1. CONCLUSIONES	99
4.2. RECOMENDACIONES	100
BIBLIOGRAFÍA	102
ANEXOS.....	111

ÍNDICE DE FIGURAS

Figura 1. Representación del sistema de imitación [Fuente: Autor].	11
Figura 2 Diagrama de bloques general del sistema [Fuente: Autor].	12
Figura 3. Partes del brazo humano [Fuente: Autor].	20
Figura 4. Articulaciones divididas en el brazo humano [Fuente: Autor].	20
Figura 5. Partes de un brazo robot antropomórfico [13].	21
Figura 6. Ilustración de un brazo robótico [20].	23
Figura 7. Ilustración de robots lineales [21].	23
Figura 8. Ilustración y representación de un brazo robot cilíndrico [21].	24
Figura 9. Ilustración y representación de un brazo robótico esférico [21].	24
Figura 10. Eslabón - articulación de un brazo robótico [Fuente: Autor].	25
Figura 11. Cinemática en brazo robot [24].	25
Figura 12 Cinemática de brazo robot referente al movimiento de una persona [26].	26
Figura 13. Cinemática método Denavit Hartenberg [30].	27
Figura 14 Diagramas de bloques del proceso de imágenes a través de visión artificial [40].	31
Figura 15. Cámara estéreo o estereoscópica [46].	33
Figura 16. Placas Arduino [61].	36
Figura 17. Placa ESP32 [63].	36
Figura 18. Raspberry Pi [65].	37
Figura 19. Cámara Kinect 360° [Fuente: Autor].	40
Figura 20. Arduino UNO [70].	42
Figura 21. Características de placa Arduino UNO [71].	42

Figura 22. Partes del brazo robótico [72].	44
Figura 23. Dimensiones del brazo robótico [72].	44
Figura 24. Distribución de los servos en el brazo robótico [72].	46
Figura 25. Servo LX-15D [72].	46
Figura 26. Servo LX-225 [72].	47
Figura 27. Controlador para servo LSC – 6 [72].	48
Figura 28. Visión de controlador en el brazo robótico [72].	49
Figura 29. Formato de HMI del sistema de imitación [Fuente: Autor].	50
Figura 30. Librería Kinect en VI Package Manager - LabVIEW [Fuente: Autor].	51
Figura 31. Puntos referenciales en articulaciones del cuerpo humano [73].	53
Figura 32. Ventana inicial del Arduino IDE [Fuente: Autor].	54
Figura 33. Monitoreo serial en Entorno Arduino IDE [Fuente: Autor].	54
Figura 34. Apartados en Arduino IDE [74].	56
Figura 35. Gestor de biblioteca en el programa Arduino IDE [Fuente: Autor].	56
Figura 36. Librerías usadas en proyecto [Fuente: Autor].	57
Figura 37. Diseño e implementación del Hardware en el sistema de imitación [Fuente: Autor].	58
Figura 38. Conexión entre cámara Kinect y ordenador [Fuente: Autor].	59
Figura 39. Conexión del brazo robótico [Fuente: Autor].	60
Figura 40. Controlador instalado en el brazo robótico [Fuente: Autor].	61
Figura 41. Conexión de tarjeta LSC-6 con servos [Fuente: Autor].	61
Figura 42. Conexión de tarjeta LSC-6 a controlador Arduino UNO [Fuente: Autor].	62
Figura 43. Brazo robótico X-Arm 1S [72].	62

Figura 44. Articulaciones en brazo robótico [Fuente: Autor].	63
Figura 45. Limitaciones en cada articulación del brazo robótico [Fuente: Autor].	63
Figura 46. Parámetro H_{10} - Método Denavit Hartenberg [Fuente: Autor].	63
Figura 47. Parámetro H_{21} - Método Denavit Hartenberg [Fuente: Autor].	64
Figura 48. Parámetro H_{32} - Método Denavit Hartenberg [Fuente: Autor].	64
Figura 49. Parámetro H_{43} - Método Denavit Hartenberg [Fuente: Autor].	64
Figura 50. Parámetro H_{54} - Método Denavit Hartenberg [Fuente: Autor].	64
Figura 51. Diagrama de conexión de la cámara Kinect [Fuente: Autor].	69
Figura 52. Diagrama de flujo para conexión entre la cámara Kinect y el programa LabVIEW [Fuente: Autor].	70
Figura 53. Inicio y configuración de cámara Kinect en Lab VIEW [Fuente: Autor].	71
Figura 54. Bloque Close-Skeleton and Video en LabVIEW [Fuente: Autor].	72
Figura 55. Filtro de datos obtenidos en LabVIEW [Fuente: Autor].	74
Figura 56. Ángulos positivo - negativo en relación manecillas del reloj [Fuente: Autor].	74
Figura 57. Esqueleto en 2D [Fuente: Autor].	75
Figura 58. Referencia de coordenadas con respecto persona [Fuente: Autor].	75
Figura 59. Procedimiento matemático para obtención de coordenadas y ángulos en LabVIEW [Fuente: Autor].	76
Figura 60. Datos de ángulos en entradas de articulaciones realizados en LabVIEW [Fuente: Autor].	76
Figura 61. Diagrama de flujo para obtener las coordenadas y ángulo [Fuente: Autor].	77
Figura 62. Uso de la función map () en el código en Arduino IDE [Fuente: Autor].	79
Figura 63. Recurso VISA y Kinect [Fuente: Autor].	80

Figura 64. Modos de control configurado en LabVIEW [Fuente: Autor].	80
Figura 65. Control de movimientos en muñeca, codo, hombros y gripper en brazo robótico [Fuente: Autor].	81
Figura 66. Lectura de ángulos del brazo derecho [Fuente: Autor].	82
Figura 67. Lectura de ángulos del brazo izquierdo [Fuente: Autor].	82
Figura 68. Diagrama del proceso [Fuente: Autor].	83
Figura 69. Ventana principal de interfaz del sistema de imitación en LabVIEW [Fuente: Autor].	84
Figura 70. Ventana de visualización de coordenadas en interfaz del sistema de imitación en LabVIEW [Fuente: Autor].	84
Figura 71. Ventana de monitoreo de interfaz del sistema de imitación en LabVIEW [Fuente: Autor].	85
Figura 72. Prueba luminosidad - Baja.	86
Figura 73. Prueba luminosidad - Alta.	86
Figura 74. Usuario a distancia menor de 2 metros.	87
Figura 75. Usuario a distancia mayor igual a 2 metros.	87
Figura 76. Usuario a distancia mayor 3 metros.	88
Figura 77. Pruebas con hombro derecho a 90° [Fuente: Autor].	89
Figura 78. Pruebas con hombro izquierdo a 90° [Fuente: Autor].	89
Figura 79. Pruebas con hombro derecho a 125° [Fuente: Autor].	90
Figura 80. Pruebas con hombro izquierdo a 125° [Fuente: Autor].	90
Figura 81. Pruebas con hombro derecho a 150° [Fuente: Autor].	91
Figura 82. Pruebas con hombro izquierdo a 150° [Fuente: Autor].	91
Figura 83. Pruebas con hombro derecho a 180° [Fuente: Autor].	92

Figura 84. Pruebas con hombro izquierdo a 180° [Fuente: Autor].	92
Figura 85. Pruebas con codo derecho a 90° [Fuente: Autor].	93
Figura 86. Pruebas con codo izquierdo a 90° [Fuente: Autor].	94
Figura 87. Pruebas con codo derecho a 125° [Fuente: Autor].	94
Figura 88. Pruebas con codo izquierdo a 125° [Fuente: Autor].	95
Figura 89. Pruebas con codo derecho a 150° [Fuente: Autor].	95
Figura 90. Pruebas con codo izquierdo a 150° [Fuente: Autor].	96
Figura 91. Pruebas con codo derecho a 180° [Fuente: Autor].	96
Figura 92. Pruebas con codo izquierdo a 180° [Fuente: Autor].	97
Figura 93. Pruebas con muñeca a 90° [Fuente: Autor].	98

ÍNDICE DE TABLAS

Tabla 1. Atributos y parámetros adicionales de cámara Kinect 360°.	41
Tabla 2. Atributos y parámetros de controlador Arduino UNO.	43
Tabla 3. Locación y ángulos de los servos en el brazo robótico.	45
Tabla 4. Atributos y parámetros del Servo LX-15D.	47
Tabla 5. Atributos y parámetros del Servo LX-225.	48
Tabla 6. Bloques para configuración de cámara Kinect en LabVIEW [Fuente: Autor].	53
Tabla 7. Los 20 puntos referenciales en articulaciones del cuerpo humano.	73
Tabla 8. Pruebas de distancia y visualización con la cámara Kinect.	87
Tabla 9. Pruebas realizadas en los hombros a 90°.	88
Tabla 10. Pruebas realizadas en los hombros a 125°.	89
Tabla 11. Pruebas realizadas en los hombros a 150°.	91
Tabla 12. Pruebas realizadas en los hombros a 180°.	92
Tabla 13. Pruebas realizadas en los codos a 90°.	93
Tabla 14. Pruebas realizadas en los codos a 125°.	94
Tabla 15. Pruebas realizadas en los codos a 150°.	95
Tabla 16. Pruebas realizadas en los codos a 180°.	96
Tabla 17. Pruebas realizadas en la muñeca a 90°.	98

LISTA DE ANEXOS

ANEXO A - Código de simulación Arduino.....	98
---	----

CAPÍTULO I

1.1 ANTECEDENTES

La robótica es una materia aun no explorada al máximo, proviene del término robot; el cual, se acuña en los años veinte del siglo pasado, la robótica industrial nace en los cincuenta y sólo en los setenta comienzan a impartirse [1], se convierte en la actualidad como un recurso para facilitar el aprendizaje y desarrollo de habilidades generales como la socialización, la creatividad y la iniciativa que permitan al ente responder de manera eficaz a los entornos cambiantes del mundo y en específico, en el ambiente virtual.

Según la Revista de Derecho Privado: “El uso de la robótica se vuelve aún más relevante para procesos de producción, montaje y procesamiento de materiales, líneas de montaje, en la producción de bienes” [2].

En general, la producción de tecnologías para la implementación de brazos robóticos se ha convertido en herramientas para profesionales y estudiantes, debido a que permite realizar tareas complejas, peligrosas y repetitivas de manera más sencilla y eficiente [3].

La programación es una parte esencial para el control de robots, incluyendo los brazos robóticos, con lo cual se facilita el aprendizaje y comunicación con base a conocimientos básicos en robótica y métodos de diseño [4]. Se pueden programar para realizar innumerables tareas en una amplia variedad de entornos.

En los últimos años, los brazos robóticos están destinados a la implementación tanto industrial, comercial, medico, laboral y educativo, se utilizan en fábricas para automatizar tareas repetitivas, aplicación de pintura a equipos o partes; ensamblar, seleccionar o clasificar productos de una cinta transportadora de distribución, en un almacén para atender pedidos de consumidores; o recolección de frutos maduros en el campo y separarlos en contenedores de almacenamiento [5].

En la actualidad, se ha logrado un progreso significativo en el reconocimiento de voz y menos en la interpretación visual en la parte digital. La visión artificial por computadora gana una relevancia particular en los últimos años, puesto que es una técnica y una disciplina en creciente auge con multitud de aplicaciones, como inspección automática, reconocimiento

de objetos, mediciones, robótica etc. [6]. La misma, cubre técnicas para adquirir, procesar, analizar y comprender imágenes del mundo real para generar información numérica o simbólica que puede ser procesada por computadoras u ordenadores, gracias a campos tan diversos como la geometría, la estadística, la física y otras disciplinas [7].

Entre los diversos progresos realizados mediante programación en robots usando imitación, también llamado aprendizaje con demostración, tenemos especialmente en el desarrollo de robots humanoides [8], brazos robóticos para desmantelamiento de bombas, brazos robóticos para laboratorios y para la parte educativa.

1.2 DESCRIPCIÓN DEL PROYECTO

En el presente proyecto se realizará el desarrollo y control de un brazo robótico para un sistema de imitación de movimientos basado en visión artificial (véase figura 1).

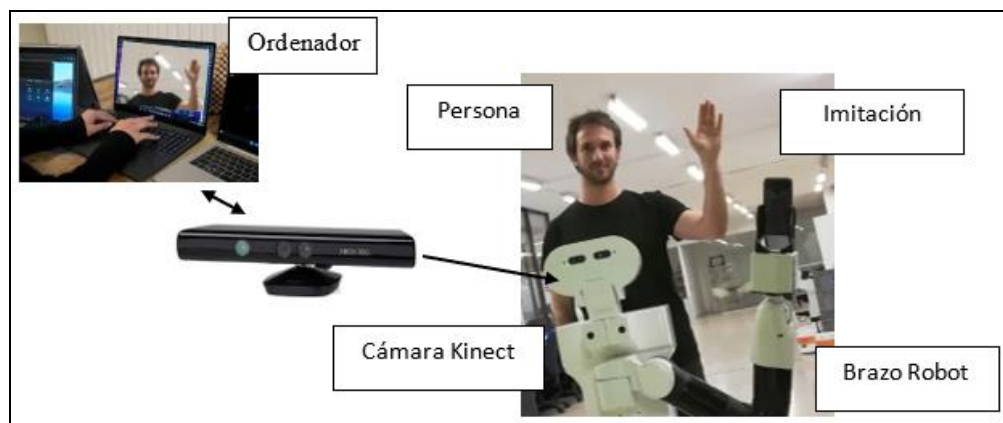


Figura 1. Representación del sistema de imitación [Fuente: Autor].

A través de la obtención de datos y mediante procesamiento de imágenes por visión artificial, se extraerá la información visual (vídeos) de la escena mediante una cámara Kinect. La captura de los videos se iniciará y terminará manualmente, así como el enfoque al espectador. El sistema de “Imitación” relacionará la representación de la información con las imágenes captadas y de esta forma, se dotará al sistema robótico de los mismos movimientos generados por el usuario.

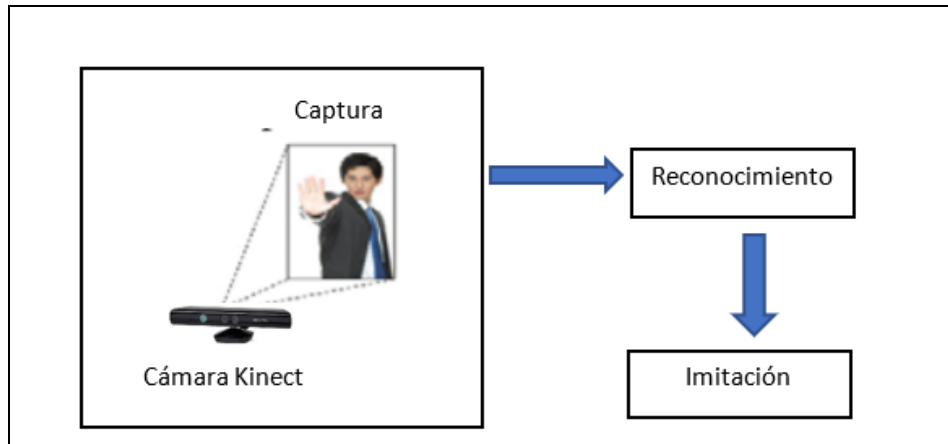


Figura 2 Diagrama de bloques general del sistema [Fuente: Autor].

Una parte investigativa está relacionada con la “imitación” de un sentido del ser humano: la vista. Mediante este sentido, se espera que el sistema dote a la computadora y a la cámara, no sólo de la capacidad de "captar" imágenes, sino también de reconocer; es decir, "reconocer" lo percibido (véase figura 2).

Debido a lo mencionado anteriormente, el uso de la robótica en conjunto con la visión artificial, permitirán aprovechar el procesamiento de órdenes a ejecutar; tal como, el imitar el movimiento del brazo de una persona hacia el brazo robot vinculado al sistema de imitación, dichas ordenes serán recibidas mediante una cámara Kinect. Los datos recopilados por el sensor Kinect, se almacenarán en el ordenador; en el cual se procesarán las imágenes que permitirán posteriormente enviar las órdenes a las articulaciones del brazo robótico para generar el movimiento deseado.

Los datos, variables e imágenes obtenidas de todo el sistema pueden ser visualizados mediante una interfaz de visión artificial, que permitirán interactuar con los sensores y actuadores implementados en el brazo robótico. Todo este estudio estará basado en tecnologías de uso actual en términos de automatización y visión artificial, referente a líneas de investigación concretas de la carrera.

1.3 OBJETIVOS DEL PROYECTO

1.3.1 Objetivo General

- Desarrollar un sistema de imitación de movimientos basado en visión artificial para el control de un brazo robótico.

1.3.2 Objetivo Específico

- Investigar las temáticas correspondientes a la visión artificial la cual será aplicada en el sistema de imitación para el correcto procesamiento de imágenes.
- Dimensionar los equipos eléctricos – electrónicos que se utilizarán en el sistema para evitar mal funcionamiento de dicho sistema.
- Implementar el sistema del brazo robótico y de la cámara Kinect para pruebas iniciales sobre los componentes del sistema.
- Seleccionar los softwares y entornos de programación adecuados que posteriormente se emplearán para la configuración del brazo robótico.
- Procesar mediante visión artificial las imágenes adquiridas por la cámara Kinect para extraer información relevante que permitirán posteriormente detectar la posición de las articulaciones del brazo robótico.
- Diseñar el algoritmo de control que permita enviar las órdenes a los actuadores de las articulaciones del brazo robótico para ejecutar el proceso de imitación.
- Ejecutar las respectivas pruebas de verificación para ajustar los parámetros acordes a las necesidades del entorno donde se localice el sistema robótico.
- Establecer la técnica de visión artificial que se usará para el procesamiento de las imágenes para extraer los parámetros necesarios que serán utilizados en el código de programación.

1.4 JUSTIFICACIÓN

Aunque los brazos robóticos de imitación por visión artificial pueden lograr una precisión notable, a veces pueden tener limitaciones en la exactitud de replicar los movimientos humanos. Esto puede deberse a diversos factores, como la calidad de la visión artificial utilizada para capturar los movimientos, la resolución y precisión de los sensores, o las limitaciones inherentes al propio diseño y construcción del brazo robótico, suelen enfrentar dificultades para adaptarse a movimientos o configuraciones no previamente entrenados. Si el sistema no ha sido entrenado para reconocer ciertos movimientos o escenarios, puede haber dificultades para interpretar correctamente las intenciones del operador humano y replicarlos de manera efectiva.

Un brazo robótico por imitación puede replicar el movimiento de un brazo humano para realizar cualquier tipo de tarea que suponga un riesgo significativo para la salud y la integridad humana. Este proyecto ofrece una herramienta para reconocer el movimiento de las articulaciones de un brazo humano y su posterior réplica sobre un brazo robótico, posibilitando así una mayor variedad de tareas en las que un brazo autónomo no podría interpretar y actuar ante diversas situaciones y en donde un ser humano podría gracias a su experticia dar una solución óptima [9].

La importancia de este proyecto se enfoca en como el brazo robótico, con distintos grados de libertad (de 5 grados), podrá realizar movimientos de imitación utilizando como base la visión artificial para así poder ejecutar tareas que sean complicadas o de riesgos para las personas [10].

1.5 ALCANCE DEL PROYECTO

El presente trabajo se realiza con propósito didáctico, la elección del tema referente a diseño, construcción y programación de un sistema de imitación tiene bases en los conocimientos y afinidades de quienes lo realizan. La problemática que se pretende satisfacer es la necesidad de vincular el brazo robótico a los movimientos del brazo derecho de una persona a través de visión artificial y que posterior este tipo de sistema pueda ser empleado en la academia y en la investigación.

Para el desarrollo de este proyecto se va a utilizar hardware y software especializado. Para la programación y codificación del sistema de imitación se espera utilizar un software de alto nivel, como por ejemplo LabVIEW, junto a librerías que permitan adquirir datos de ciertas variables de interés, posteriormente procesar esta información mediante un algoritmo de control y finalmente enviar señales de control a los diferentes actuadores.

Como parte de la implementación de este proyecto se adquirirá un computador portátil que debe contar con una memoria mínima 4 de RAM, un procesador COREi5 o Ryzen 3, Windows 10 u 11. Estas características técnicas permitirán posteriormente instalar el software de alto nivel para el procesamiento de las imágenes y las interfaces requeridas en el proceso de control y supervisión.

Como ya se ha mencionado hasta el momento el sistema propuesto contará con un brazo robótico antropomórfico de 5 grados de libertad con una garra al final de sus eslabones como elemento de manipulación. Cabe recalcar que se va a adquirir un brazo tipo comercial y a escala, puesto que la propuesta de este proyecto se centra en la programación de la lógica de control para la adquisición de imágenes referentes a los movimientos de un brazo humano para su posterior réplica en un brazo humano. Este brazo deberá contar con una tarjeta controladora que le permita establecer la conexión alámbrica o inalámbrica con el computador portátil.

También se necesitará de una cámara 3D, como por ejemplo una cámara Kinect, que sirva de periférico de entrada para la adquisición de imágenes con una resolución mínima de 640×480 pixeles y rango de profundidad mínimo de 1.2 – 3.5 metros. Dicha cámara deberá ser compatible con el software a usar.

El retardo en el envío y recepción de la información se deberá porque los diferentes elementos como la cámara, el controlador del robot, el procesador del computador maneja diferentes velocidades de procesamiento de la información, además dependerá de los sensores acoplados en el brazo robótico para saber si ha girado ciertos grados de orientación en las articulaciones.

Los diferentes equipos tienen microcontroladores o microprocesadores que manejan velocidades diferentes de transmisión de datos, o tal vez las librerías de los controladores de procesamiento de imágenes tengan o tomen tiempo más de lo esperado.

También cabe resaltar que la precisión del sistema dependerá del entorno en donde se esté utilizando la cámara junto al brazo robótico debido a factores lumínicos o de temperatura, podría darse que dado la estabilidad lumínica influirá en la calidad de procesamiento teniendo en cuenta que, a mayor luz o ausencia de esta, existirá perturbaciones en la captación del brazo humano, se puede llegar a tener dificultades para capturar y replicar con precisión los movimientos humanos. La presencia de obstáculos, interferencias o cambios rápidos en el entorno puede afectar el rendimiento y la eficiencia del sistema.

Debido a que es un brazo robot a escala se van a ir desgastando los engranajes y se reflejarán como error en los grados de giros, al ser este un prototipo se generarán oscilaciones o vibraciones en la generación de movimientos; por tanto, los movimientos reales del brazo humano no llegarían a estar en fase o sincronizados con los movimientos del brazo robótico.

1.6 METODOLOGÍA

A continuación, se indican las fases de la metodología del proyecto:

- **Fase I Revisión bibliográfica:** Para el desarrollo de este proyecto, se realizará las respectivas investigaciones en libros, revistas o artículos relacionados al manejo y programación de brazos robóticos, visión artificial, métodos de automatización, el uso de microcontroladores y sensores.
- **Fase II Selección de equipos eléctricos, electrónicos y electromecánicos:** Se procederá a seleccionar los dispositivos, sensores, actuadores, cámara, entre otros, que cumplan con los requerimientos mínimos establecidos por el sistema.
- **Fase III Verificación del funcionamiento y comunicación de los diferentes equipos:** De forma individual, en el caso del cámara primero se iniciará obteniendo imágenes estáticas, se harán uso de algoritmos sencillos para extraer información, en el caso del brazo robótico se enviarán señales a las articulaciones para verificar que los motores acoplados sean funcionales.
- **Fase IV Proceso de programación y control del sistema de imitación:** Se procede a realizar la programación del sistema de imitación en un programa de alto nivel con sus respectivas librerías en un ordenador con características mínima de arranque.
- **Fase V Diseño de una interfaz de usuario para visualizar datos de interés:** Mediante el proceso de simulación se visualizarán en el software LabVIEW, los datos relevantes de los sensores y actuadores que se recibirán en tiempo real. Se incluirán imágenes, valores y grados en los que las articulaciones estén en movimiento.
- **Fase VI Desarrollo de pruebas experimentales:** se procederá a hacer pruebas con el sistema ya conectado entre sí para conseguir una mayor estabilidad en el brazo robótico y evitar vibraciones u oscilaciones en el mismo.

CAPITULO II

2.1 MARCO CONTEXTUAL

Los sistemas desarrollados en imitación a través de visión artificial para el manejo de brazos robóticos constituyen un área de la robótica que busca argumentar que los movimientos realizados por personas puedan ser ejecutados por manipuladores robóticos mediante el empleo de procesamiento de imágenes con visión artificial.

En la actualidad, es muy común ver varios modelos de brazos robóticos los cuales son utilizados para aplicaciones específicas en diferentes campos de la industria, sociedad e investigación como por ejemplo industrias farmacéuticas, medicina y en la educación, los cuales comparten una característica particular puesto que todos ellos son controlados mediante la imitación de movimientos sincronizados de un ser humano.

El sistema de imitación de movimiento basado en visión artificial para el control de un brazo robótico, a través de una cámara Kinect, contará con una resolución y actualización de imágenes de 30 FPS, además de un rango de profundidad de 1,2 – 3,5 metros.

El sistema consistirá en la imitación de los movimientos de una persona en un brazo robótico de 5 grados de libertad, este sistema de imitación será basado en visión artificial realizado en una interfaz gráfica desarrollado en el programa LabVIEW, el procesamiento de imágenes será a través de una cámara Kinect 360. Gracias a la codificación del sistema por imitación, se logrará el accionamiento de los servomotores integrados al brazo robótico para posterior movimiento sincronizado al movimiento del brazo humano.

La cámara Kinect recolectará imágenes, las mismas que serán procesadas para que el robot pueda realizar la imitación y procesar tareas de riesgo para la salud e integridad del humano que desee aplicar. Debido a estas tareas, se realizará el estudio y la programación del sistema, se mantendrá el monitoreo constante de las variables y parámetros de los actuadores, que serán visualizados en un software de alto nivel (LabVIEW).

Los rangos de seguridad permisibles del brazo robótico serán mínimos de un radio de 1 metro, puesto que los movimientos de este pueden dañar tanto al mecanismo como a la persona o personas en su alrededor; así mismo, la cámara Kinect por precaución no debe

colocarse en este radio mínimo del brazo robótico, puesto que queda expuesta a que haya roces o golpes de este a la cámara.

Este tipo de sistema abarca en la actualidad una amplia variedad de tareas en las cuales empresas no podrían realizar mediante brazos autónomos y en la parte educativa siendo esta más resaltable para su provecho en instrucción en nuevos ingenieros.

Este anteproyecto va a contribuir a los estudiantes de la carrera referente a electrónica o campos semejantes a esta, para generar en posteriores proyectos una ayuda en la que se utilice algún tipo de imitación basado en visión artificial.

2.2 MARCO CONCEPTUAL

En este capítulo, se hará una descripción de todos los elementos que compondrán el sistema de imitación de un brazo humano como lo es el brazo robótico, los elementos y tipos de brazos robóticos actuales, los grados de libertad y la parte cinemática que tienen estos. A su vez, se explicará cómo el procesamiento de imágenes será de ayuda en la visión artificial mediante una cámara, todo esto realizado en un lenguaje de programación de alto nivel que permitirá a una tarjeta controladora realizar el movimiento del manipulador.

2.2.1 Brazo humano

El brazo de un ser humano tiene una serie de movimientos básicos (véase figura 3) que se realizan en diversas tareas de la vida diaria: movimiento del hombro, flexión de antebrazo, rotación de muñeca y agarres [11].

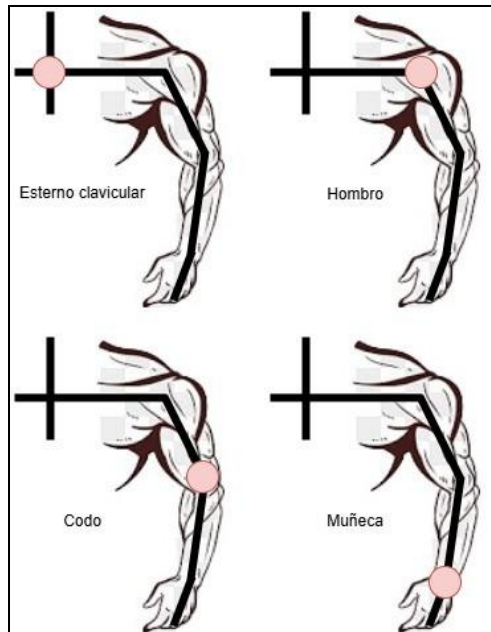


Figura 3. Partes del brazo humano [Fuente: Autor].

Cada articulación está definida por un centro de articulación, que se considera un punto unido a dos cuerpos adyacentes. En particular, la articulación del hombro (véase figura 4) se considera una simple articulación esférica que conserva la funcionalidad del movimiento del hombro, pero no su configuración real [12]. El antebrazo se considera un cuerpo rígido, por lo que se debe considerar el movimiento de pronación y supinación alrededor del codo. Las manos están construidas como cuerpos rígidos.

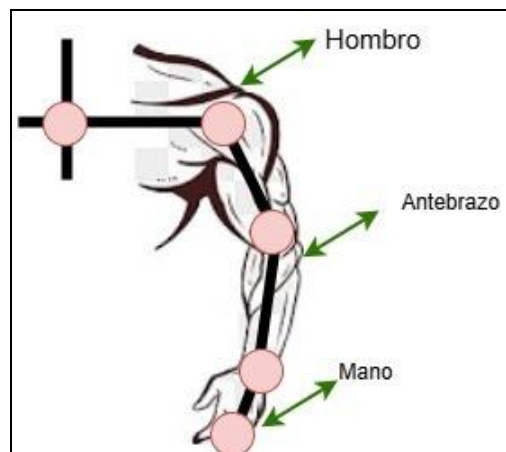


Figura 4. Articulaciones divididas en el brazo humano [Fuente: Autor].

2.2.2 Brazos robóticos

Son robots que tienen una estructura parecida al brazo de una persona, el cual se divide en diversas partes (véase figura 5) como los son hombro, codo y la muñeca el cual es complementado por una herramienta adicional tipo pinza o garra que sirven para enganchar o maniobrar cosas estos pueden ser controlados de manera manual o programados.

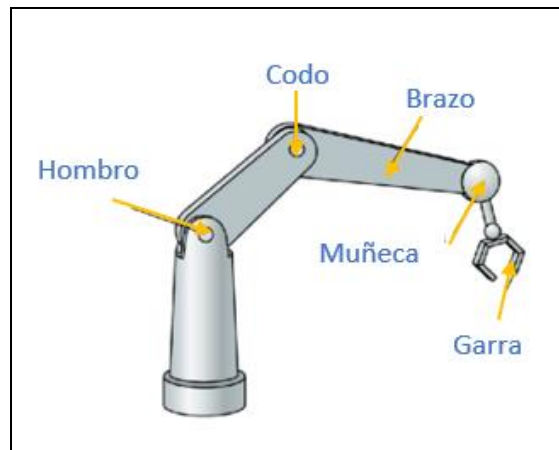


Figura 5. Partes de un brazo robot antropomórfico [13].

Estos son usados en diversas áreas teniendo más relevancia en los área médica y procesos automotrices. En la parte automotriz estos mecanismos son usados para el ensamblaje, pintado. En el área médica los brazos robóticos son usados por su alta precisión al realizar distintos procesos [14].

2.2.2.1 Elementos de un brazo robótico

Los brazos robóticos más comunes están compuestos por una serie de elementos que están interconectados los cuales le permite ejecutar distintas tareas, entre los elementos más comunes encontramos:

- **Estructuras mecánicas:** Mecánicamente, cualquier tipo de robot consta de articulaciones que se encuentran conectadas entre sí que permiten movimientos relativos cada cierto eslabón continuos. La parte física de la mayoría de los autómatas son muy similares ya que siguen la anatomía del brazo humano por lo tanto para poder

identificar los diferentes elementos se utilizan términos como hombro, brazo, codo, antebrazo, muñeca y mano [15].

- **Articulaciones:** Son los puntos donde se conectan cada parte del manipulador entre estas se encuentran las rotacionales (realizan giros en un solo eje) y las prismáticas (realizan desplazamientos lineales), estos movimientos pueden ser ejecutados por pistones o por métodos de desplazamientos que puede ejecutar a través de componentes mecánicos o eléctricos [16].
- **Actuadores:** Son dispositivos que suministran la potencia y el movimiento necesario para que las articulaciones puedan realizar los movimientos, los más usados son los motores DC (corriente continua) o servomotores que se encargan de realizar pequeños controles en cada articulación [17].
- **Sensores:** También conocidos como captadores, permiten a los manipuladores robóticos interactuar con su entorno para la recolección de información, teniendo diferentes sensores que sirven para medir la posición y orientación, sensores que permite medir la fuerza a aplicarse entre otras [18].
- **Controlador:** Es el cerebro que controla al manipulador robótico estos son los encargados de reunir la información emitidos por los sensores. Existen diferentes tipos de controladores que van desde microcontroladores o computadoras que emplean algoritmos [19].

2.2.2.2 Tipos de brazos robóticos

Dentro de los distintos brazos robóticos adaptados a diferentes áreas tenemos los siguientes:

- **De tipo articulado:** Los brazos robóticos articulado (véase figura 6) poseen una estructura que están compuestos por varios fragmentos acoplados a articulaciones giratorias, de las cuales se obtiene un grado de libertad por cada segmento (estos reflejan al menos tres grados de libertad) para tener una mayor flexibilidad.

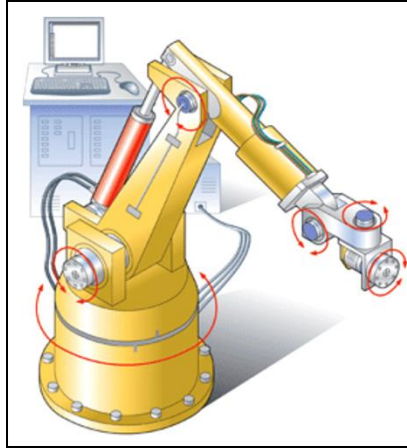


Figura 6. Ilustración de un brazo robótico [20].

- **De tipo cartesiano:** También son conocidos como robots lineales, estos pueden realizar movimiento en los ejes cartesianos x , y , z (véase figura 7) usando motores y correas las cuales le permite tener rapidez y exactitud al moverse. A los brazos robóticos cartesianos también se les puede adaptar funciones de visión artificial para mejorar su rendimiento, siendo usados en la elaboración de láminas de circuitos impreso y en los grabados en 3 dimensiones (3D).

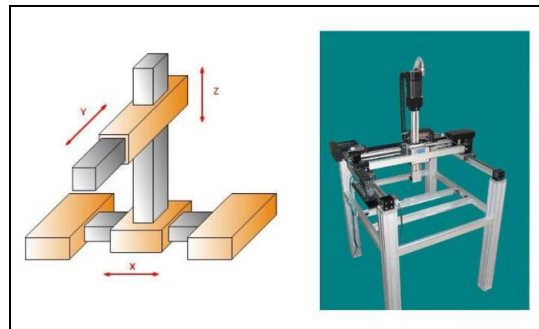


Figura 7. Ilustración de robots lineales [21].

- **De tipo cilíndrico:** Los brazos cilíndricos son similares a los brazos articulados (véase figura 8), pero estos cuentan con una parte rotatoria en la base y dos partes prismáticas las cuales se emplean para acoplar el radio y la elevación en el espacio de trabajo. Estas clases de robots son usados en áreas donde se demanden movimientos de rotación y desplazamiento tales como el traslado de materiales de un almacén o en áreas de fabricación.

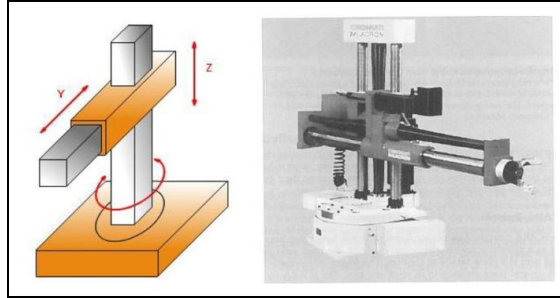


Figura 8. Ilustración y representación de un brazo robot cilíndrico [21].

- **De tipo polar:** También conocido como brazos robóticos esférico las cuales usan el sistema de coordenadas polares tridimensionales (radio, ángulo, altura). El diseño de estos puede variar según sea su necesidad, sin embargo, estos cuentan con una base rotatoria (véase figura 9) el cual permite realizar movimiento en el plano horizontal y una extremidad extensiva que puede acoplar su longitud y una parte encargada de realizar movimientos en el plano vertical.

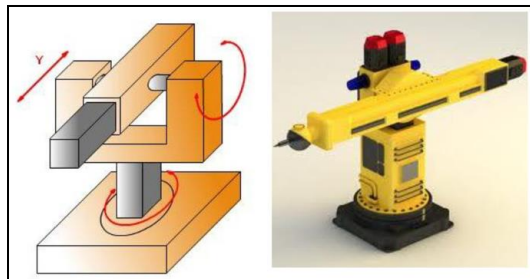


Figura 9. Ilustración y representación de un brazo robótico esférico [21].

2.2.2.3 Grados de libertad de un robot

Los grados de libertad son cada movimiento independiente vinculados en relación con la anterior, eslabones (véase figura 10). Se determina la estructura mecánica y el tipo del robot

mediante tres tipos de articulaciones (lineales, esféricas y cilíndricas), proporcionan una configuración específica para los tres ejes principales [22].

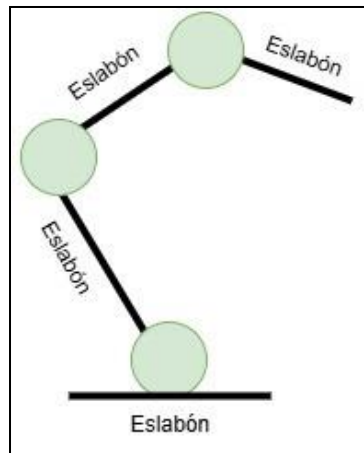


Figura 10. *Eslabón - articulación de un brazo robótico [Fuente: Autor].*

2.2.2.4 Cinemática de un brazo robótico

La cinemática de un robot (véase figura 11) estudia su movimiento con respecto a un marco de referencia. Por lo tanto, la cinemática está interesada en la descripción analítica de los cambios en el movimiento espacial del robot a lo largo del tiempo, especialmente en la relación entre la posición final y la orientación del robot y sus valores de coordenadas conjuntas en dos perspectivas [23].

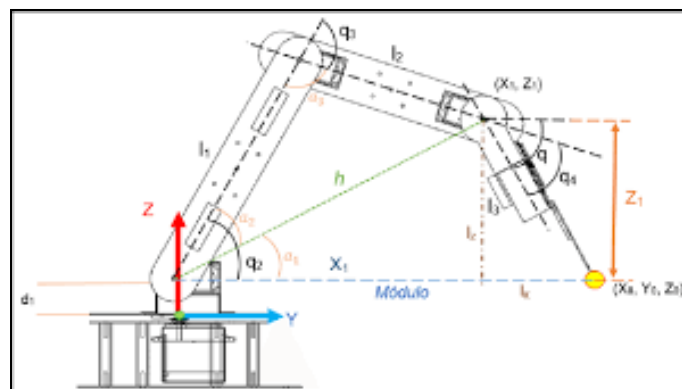


Figura 11. *Cinemática en brazo robot [24].*

Una herramienta indispensable para describir la geometría del espacio del manipulador es una representación homogénea de coordenadas. El concepto de una representación de coordenadas homogéneas en el espacio euclidiano tridimensional es de utilidad para el desarrollo de transformaciones de matrices, incluidas las transformaciones de rotación, traslación, escala y perspectiva. En general, una matriz de transformación homogénea es una matriz que transforma un vector de posición expresado en coordenadas homogéneas de un sistema de coordenadas a otro [25].

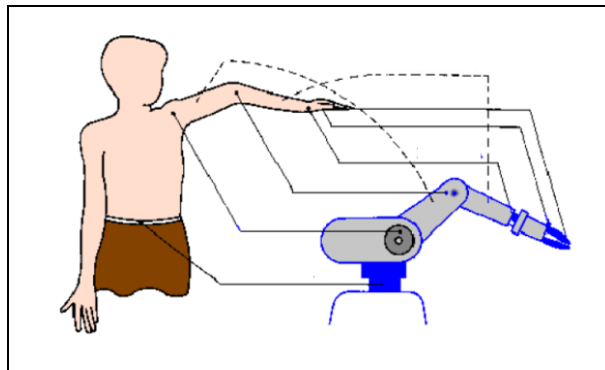


Figura 12 Cinemática de brazo robot referente al movimiento de una persona [26].

En cinemática de robots; incluyendo a los brazos robóticos (véase figura 12), hay dos importantes [27]:

- **Cinemática directa**

La solución al problema de la cinemática directa es encontrar el valor de la posición final del manipulador del robot en función de los valores de los ángulos de las articulaciones, es decir valores de traslación o rotación de juntas [28].

Aunque la cinemática del robot se puede solucionar geoméricamente, el método de la representación sistemática de Denavit Hartenberg (véase figura 13) ofrece la ventaja de conocer tanto la posición final de manipulador como la posición de cada una de sus articulaciones [29].

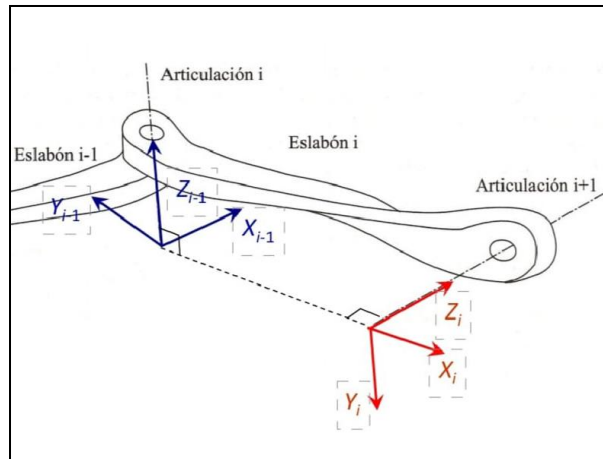


Figura 13. Cinemática método Denavit Hartenberg [30].

Se utiliza fundamentalmente el álgebra vectorial y matricial para representar y describir la localización de un objeto en el espacio tridimensional con respecto a un sistema de referencia fijo.

Dado que se puede pensar en el robot como una cadena cinemática de objetos rígidos o eslabones conectados por eslabones, es posible establecer un marco de referencia fijo en la base del robot y la posición de cada eslabón con respecto a este marco de referencia.

El método sistemático de Denavit Hartenberg posee pautas o pasos a seguir, los cuales se mencionarán a continuación [31]:

1. **Paso 1.** Identificación de los sistemas de coordenadas asociados a cada articulación, cumpliendo las siguientes reglas:
 - El eje Z_i coincide con el eje de rotación de la junta (si es rotacional) o la dirección del movimiento (si es prismática).
 - El origen se posiciona donde se interseca Z_i con el eje X_{i-1} (o lo más cerca posible si no se intersecan).
 - El eje X_i es perpendicular a Z_i y apunta hacia el siguiente eslabón.
 - El eje Y_i se define para completar un sistema de coordenadas ortogonal (según la regla de la mano derecha).

2. **Paso 2.** Definir los parámetros de Denavit Hartenberg para cada par de eslabones consecutivos ($i - 1$ e i), define los siguientes parámetros DH:

- Ángulo de torsión (α_{i-1}): Es el ángulo (en radianes) entre los ejes Z_{i-1} y Z_i , medido alrededor de X_{i-1} .
- Distancia entre ejes (a_{i-1}): Es la distancia (longitud) entre Z_i y Z_{i-1} a lo largo de X_{i-1} .
- Desplazamiento de la junta (d_i): Es la distancia entre los ejes X_{i-1} y X_i , medida a lo largo de Z_i . Es variable si la junta es prismática.
- Ángulo de la junta θ_i Es el ángulo entre los ejes X_{i-1} y X_i , medido alrededor de Z_i . Es variable si la junta es rotacional.

3. **Paso 3.** Construir las matrices de transformación, utilizando los parámetros DH para construir la matriz de transformación homogénea A_i que relaciona el sistema de coordenadas del eslabón $i - 1$ con el eslabón i . Dicha matriz está definida como:

$$A_i = \begin{bmatrix} \cos(\theta_i) & -\cos(\alpha_i)\sin(\theta_i) & \sin(\alpha_i)\sin(\theta_i) & a_i \cos(\theta_i) \\ \sin(\theta_i) & \cos(\alpha_i)\cos(\theta_i) & -\sin(\alpha_i)\cos(\theta_i) & a_i \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ecuación 1. Matriz de transformación homogénea.

4. **Paso 4.** Multiplicar las matrices para obtener la cinemática directa, esto permitirá calcular la posición y orientación del efector final multiplicando las matrices de transformación de todos los eslabones:

$$T = A_1 \cdot A_2 \cdot A_3 \cdots A_n$$

Donde T es la matriz homogénea que describe la posición y orientación del efector final en el sistema base.

5. **Paso 5.** Interpretar los resultados, por tanto, la matriz homogénea final T tiene la forma:

$$T = \begin{bmatrix} R_{11} & R_{12} & R_{13} & R_x \\ R_{21} & R_{22} & R_{23} & R_y \\ R_{31} & R_{32} & R_{33} & R_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ecuación 2. Matriz homogénea final T.

6. **Paso 6.** Ajustar para configuraciones específicas si el robot tiene juntas prismáticas o rotaciones, reemplaza d_1 o θ_1 con sus respectivos valores variables según su estado.

- **Cinemática inversa**

La cinemática inversa implica encontrar los vectores de ángulos articulares, este problema es difícil de resolver en términos de orientación y posición del efector final debido a las ecuaciones no lineales y las múltiples soluciones involucradas [32].

Dadas las dimensiones de las partes del manipulador (cuerpo, antebrazo, mano, etc.) y la posición y orientación deseada del efector final (indicada por vectores), se intenta obtener los valores de los ángulos que forman en las articulaciones. para poder moverse en la orientación deseada y posicionar el robot en un punto específico del espacio con alta precisión [33].

2.2.3 Visión artificial

La visión artificial es un campo de la inteligencia artificial donde se aplican técnicas que permiten la obtención, procesamiento y estudio de cualquier investigación a través de imágenes digitales [34].

La visión artificial tiene la misma función que la visión humana con la diferencia que esta se encarga de capacitar a una maquina donde se emplean cámaras, información y algoritmos en vez de retinas y nervios visuales, para lograr esto se usan programas como lo son LabVIEW, Matlab, Python, machine learning entre otros lo cuales usan datos mediante algoritmos que le permiten al ordenador diferenciar una imagen de otra.

Para tener una buena comunicación entre el brazo robótico y el entorno donde se encuentra es necesario la creación de sistemas avanzados de inteligencia artificial como lo es la visión artificial [35].

2.2.3.1 Procesamiento de imágenes

Debido a la complejidad y la cantidad de cálculos a realizar, el análisis y procesamiento de imágenes se realiza con la ayuda de computadoras. Por tanto, aunque las fórmulas matemáticas necesarias para ello tienen siglos de antigüedad, la posibilidad real de utilizarlo en la práctica diaria sólo ha sido posible en las últimas décadas gracias al desarrollo de la tecnología de hardware [36].

El procesamiento y análisis de imágenes se desarrolló para resolver los tres mayores problemas asociados con las imágenes [37]:

- La digitalización y codificación de imágenes son útiles para la transmisión, visualización y almacenamiento de imágenes.
- Mejora y actualización de imágenes para explicar mejor su contenido.
- Descripción y segmentación de imágenes para aplicaciones de robótica o visión por computadora.

El procesamiento de imágenes viene de la mano en la mayoría de las veces con la visión artificial, el cual, es un método que ha ido evolucionando con el tiempo debido al desarrollo de la tecnología y la ciencia, comenzó como un programa que permitía "ver" la estructura del bloque y enviarlo al ordenador con la ayuda de una cámara, hasta convertirse en una ciencia propia, pudiendo obtener una representación adecuada de los objetos. La imagen se procesa hasta filtrarla para crear una versión modificada de la imagen, se segmenta la imagen para identificar el objeto en estudio, se extraen las características de interés observadas en el objeto y finalmente ajusta la imagen [38].

El manejo de la visión artificial (véase figura 14) mejora la eficiencia de un brazo robótico ya que ofrece una amplia funcionalidad las cuales solo son proporcionadas por la cámara la cual ofrece niveles alto de autonomía y brindar información suficiente para la adquisición de decisiones [39].

El algoritmo que sigue la visión artificial y el procesamiento de imágenes se muestra a continuación [35]:

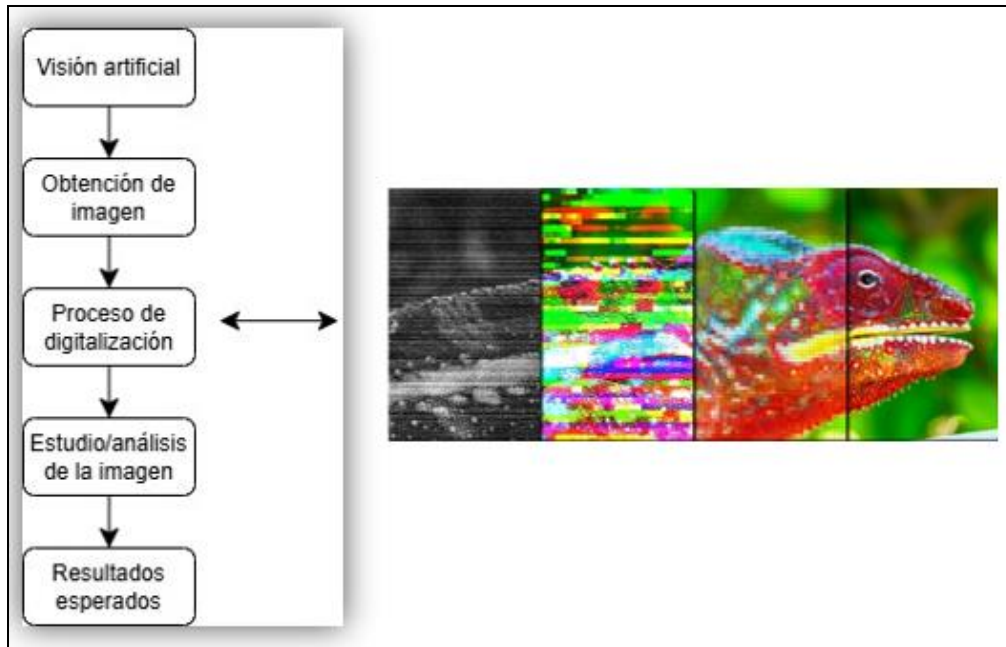


Figura 14 Diagramas de bloques del proceso de imágenes a través de visión artificial [40].

Las técnicas de análisis de imágenes digitales se incluyen en el campo conocido como procesamiento de imágenes digitales. Las herramientas están organizadas según el nivel de procesamiento requerido para analizar la información en imágenes digitales. Una breve descripción de ellos es la siguiente [41]:

- 1) **Pretratamiento:** Adapta la información de la imagen y realizará un mejor análisis en los siguientes pasos. Ejemplos de procesamiento son operaciones con brillo y contraste.
- 2) **Segmentación:** Divide la imagen en regiones que representan la información necesaria para resolver el problema.
- 3) **Determinación y clasificación de objetos:** Identifica y clasifica objetos contenidos en imágenes.
- 4) **Análisis de imágenes:** Obtiene información sobre el contenido de la imagen.

A continuación, se explicará algunas de las definiciones que se encuentran implicadas dentro del procesamiento digital de imágenes [42]:

- **Visión por computadora:** Implica la recopilación, procesamiento, clasificación y reconocimiento de imágenes digitales.

- **Píxeles:** Elementos básicos de las imágenes (elementos de imagen).
- **Imagen:** Disposición bidimensional de píxeles con diferentes intensidades de luz (escala de grises).
- **Color:** El color es una combinación de tres colores primarios (RGB en inglés): rojo, azul y verde.

2.2.3.2 Tecnologías utilizadas en cámaras

Entre las tecnologías utilizadas en cámaras para el uso de imitación se pueden describir las siguientes:

Cámaras 3D: La función básica de una cámara 3D es imitar lo que ve el ojo humano. Es importante recordar que los ojos humanos están separados por aproximadamente dos pulgadas y media, por lo que los ojos ven diferentes perspectivas, que el cerebro interpreta de manera diferente. Usando una cámara 3D, el cerebro combina las dos imágenes para crear una sensación de profundidad o tres dimensiones [43].

Cámara 360°: La resolución de las imágenes con cámaras de 360° es mucho menor que las tomadas de forma tradicional. El procesamiento aplicado a las imágenes en 360° corresponde a la reconstrucción en perspectiva, es decir, se realiza la transformación de la imagen en proyecciones oblicuas, a partir de las cuales se emparejan los puntos; lo anterior se aplica de manera análoga a las cámaras panorámicas [44].

Cámara estéreo: Para obtener una imagen estéreo, se toman imágenes de dos cámaras (véase figura 15) y se encuentra una correspondencia entre los puntos de las dos cámaras. En base a la diferencia en las posiciones de estos puntos en las dos imágenes, la profundidad se puede obtener por triangulación [45].



Figura 15. Cámara estéreo o estereoscópica [46].

2.2.3.3 Lenguajes de programación utilizados en visión artificial

Dentro de los lenguajes de programación que usan los softwares de alto nivel se encuentran los siguientes:

Python: Es un intérprete o consola que permite probar ciertas características del lenguaje sin construir módulos para él. Sus capacidades son muy similares a las que proporciona el lenguaje Matlab para determinadas tareas [47]. Python posee para visión artificial las siguientes librerías:

- Pytorch, el cual admite aplicaciones de aprendizaje profundo. Esta capacidad de respuesta y conveniencia híbrida hace que PyTorch sea muy útil para desarrollar redes neuronales profundas. Tiene una amplia gama y es adecuado para diversas aplicaciones. Así como Python se usa para programar, PyTorch es un excelente prólogo para el aprendizaje profundo y una herramienta disponible para aplicaciones del mundo real [48].
- Scikit, esta biblioteca está integrada en SciPy, un ecosistema abierto de software de matemáticas, ciencia e ingeniería desarrollado en Python [49].
- Keras, es una biblioteca de redes neuronales escrita en Python que puede ejecutarse en Tensorflow y Theano. Compatibilidad con redes neuronales convolucionales y redes neuronales recurrentes (combinaciones de estas), admite esquemas de conexión

arbitrarios (incluido el entrenamiento de múltiples entradas y múltiples salidas),
funcionable con CPU y GPU compatible con versión de Python 2.7 en adelante [50].

C++: Generalmente, los programas en C suelen estar escritos en diferentes archivos, el compilador combina correctamente estos archivos y los convierte en códigos de objeto para producir programas ejecutables. El lenguaje C consta de una pequeña cantidad de instrucciones, pero proporciona amplias bibliotecas y herramientas que los programas pueden importar cuando sea necesario [51]. El lenguaje C++ y C posee para visión artificial las siguientes librerías:

- OpenCV, la cual es una biblioteca de visión artificial gratuita para el procesamiento de imágenes que se puede utilizar en una variedad de aplicaciones comerciales y de investigación entre ellas la detección de movimiento o control de procesos mediante reconocimiento de objetos [52].
- TensorFlow, también usada en Python, su significado proviene de la estructura de datos principal que conforma la biblioteca, el "tensor", y se deriva de las operaciones que realizan las redes neuronales en matrices multidimensionales de datos [53].

Matlab: Sistema interactivo cuyos componentes de datos son matrices adimensionales. Permite resolver muchos problemas técnicos de computación, especialmente aquellos relacionados con formulaciones de matrices y vectores, utilizando bibliotecas especializadas y código central [54].

Java: Su sintaxis es muy similar a C++ y C, lo que facilita la migración para los desarrolladores que ya están familiarizados con estos lenguajes. A pesar de la adopción de la sintaxis, se eliminaron todas las características conflictivas de estos lenguajes, como el acceso directo a las direcciones de memoria a través de punteros o la gestión manual de memoria dinámica [55].

2.2.4 Sistemas de comunicación vinculados a brazos robóticos

Entre los sistemas de comunicación con los cuales se pueden vincular los brazos robóticos se tienen los siguientes:

USB: Universal Serial Bus, es una de las tecnologías más comunes y utilizadas desde sus inicios, desde su origen, dota a un PC de un bus externo de alta velocidad que ofreciera características idóneas; fácil conexión y desconexión [56].

Wifi: La transmisión WIFI es un medio compartido, y cualquier dispositivo conectado al medio puede usar la señal de transmisión, los mensajes de comunicación compartidas por este medio pueden ser interceptados o bloqueados dentro del rango de la señal [57].

Bluetooth: Se define como un modelo completo (tanto de hardware como de software) para comunicaciones inalámbricas de baja potencia utilizando señales de radio en la banda de frecuencia ISM; sus siglas significan Industrial Scientific Medical, con una aproximación de 2,4 GHz [58]. La tecnología hace posible conectar diferentes dispositivos mediante conexiones de radiofrecuencia de corto alcance.

2.2.5 Tarjetas controladoras

Una tarjeta controladora suele ser una tarjeta que realiza el control del sistema porque en este sistema se encuentra el microcontrolador que se encarga de realizar todas las tareas necesarias a seguir a lo largo de la tarea o proyecto a realizarse [59]. Entre estas tarjetas controladoras encontramos:

Arduino: Es una plataforma de prototipo electrónico, que consiste principalmente en placas de microcontroladores. En el entorno de desarrollo, la programación del lenguaje admite la entrada y salida de datos y señales [60]. Entre las placas Arduino se pueden mencionar algunas como en las mostradas en la figura 16.

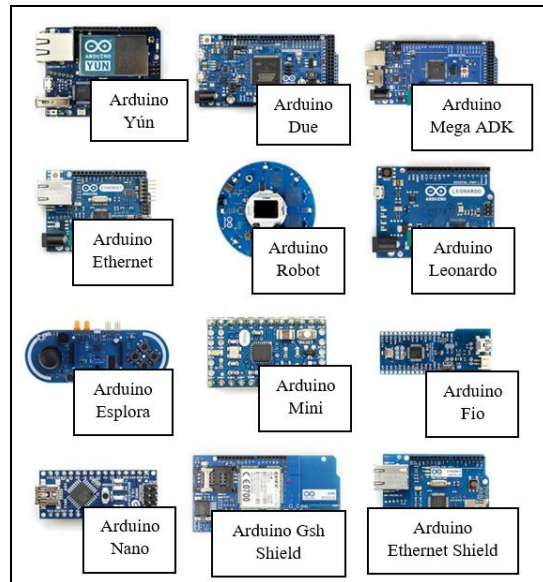


Figura 16. Placas Arduino [61].

Esp32: El módulo ESP32 (véase figura 17) es un módulo basado en MCU (unidad de microcontrolador) de 32 bits que contiene un dispositivo de comunicación inalámbrica con una frecuencia de 2,4 GHz, lo que permite la conexión a redes WI-FI y Bluetooth [62].

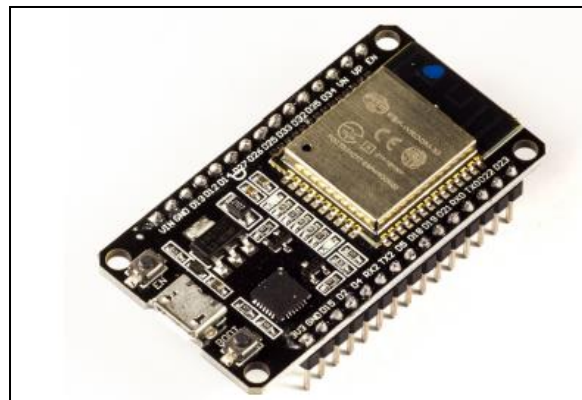


Figura 17. Placa ESP32 [63].

Raspberry PI: Raspberry Pi (véase figura 18) tal como Arduino, permite programar de manera muy flexible. Se puede hacer usando varias herramientas y diferentes lenguajes de programación (Python, Java, C o C++) o desde la consola u ordenador propio [64].



Figura 18. Raspberry Pi [65].

2.3 MARCO TEÓRICO

Los siguientes proyectos están relacionados con el desarrollo de un sistema de imitación de movimientos basado en visión artificial para el control de un brazo robótico, el cual está a disposición para proporcionar orientación en investigación y desarrollo de solución técnica.

En el artículo publicado por Sobrado Malpartida y por Eddie Ángel de la Pontificia Universidad Católica del Perú, con el tema “SISTEMA DE VISIÓN ARTIFICIAL PARA EL RECONOCIMIENTO Y MANIPULACIÓN DE OBJETOS UTILIZANDO UN BRAZO ROBOT”; implementa la tecnología de clasificación y aprendizaje automático del sistema de visión artificial, y cuando la pieza específica es identificada y posicionada, se envía una señal de comando al robot manipulador para que la recoja y la coloque en la posición predeterminada por el operador [66].

El artículo científico realizado por Nope Rodríguez Sandra, Loaiza-Correa Humberto y por Caicedo-Bravo Eduardo, lleva por título “IMITACIÓN DE GESTOS POR BRAZOS ROBÓTICOS: UNA PROPUESTA PARA EVALUAR SU CALIDAD.”; publicado en la revista científica DYNA-Ingeniería e Industria, tiene como objetivo el abordar la propuesta de una evaluación cuantitativa con 4 movimientos imitados por un brazo robot simulado; especificando una métrica simple para cada forma geométrica [67].

El trabajo publicado en el “Congreso Tecnología Aprendizaje y Enseñanza de la Electrónica” TAAE2016 en la ciudad de Zaragoza, con tema “DESARROLLO DE UN BRAZO ROBÓTICO E IMPLEMENTACIÓN DE UNA ESTRATEGIA DE CONTROL POR

RECONOCIMIENTO DE GESTOS MEDIANTE LEAP MOTION”; el propósito de este trabajo realizado por Artal Jesús Sergio y por Montañés Luis, es desarrollar un prototipo de brazo robótico articulado e implementar un método de control mediante reconocimiento de gestos (Leap Motion Sensor) utilizando movimientos naturales del antebrazo y la mano [68].

En la Universidad Piloto De Colombia, Jhon Alexander Niño Peña, Pablo Enrique Parra Gaona y por Sebastián Felipe Rodríguez Cortes, mediante su tema de proyecto de grado “BRAZO ROBÓTICO MANIPULADOR DE RESIDUOS PELIGROSOS ENVASADOS”, como objetivo el diseño y construcción de un brazo robótico que imite el movimiento de un brazo humano para desechar desechos peligrosos empaquetados el cual llegó como conclusión a una estrategia de procesamiento de imágenes capaz de rastrear la trayectoria el cual, el antebrazo humano proporcionó una referencia al sistema de control posterior [69].

CAPITULO III

3.1. COMPONENTES DE LA PROPUESTA

En esta sección se describirán características técnicas y de funcionamiento tanto del hardware y software del sistema de imitación mediante visión artificial, también se describirá la lógica de programación de los algoritmos usados, así como los cálculos matemáticos implicados.

3.1.1. COMPONENTES FÍSICOS

Se mencionará en este apartado de manera explícita los componentes que son parte del sistema de imitación, el modelo de la cámara Kinect, la tarjeta controladora para la comunicación serial mediante Arduino, así como la conectividad y composición del brazo robótico X-Arm 1S, sus tipos de servos de cada articulación y su tarjeta controladora.

3.1.1.1. Cámara Kinect Xbox 360° modelo: 414

Desarrollada originalmente para la consola de juegos Xbox 360, la cámara Kinect 360 ha trascendido su uso original en videojuegos para convertirse en una herramienta versátil para proyectos de investigación y desarrollo tecnológico. El dispositivo combina un sensor de profundidad basado en luz infrarroja y una cámara RGB para capturar imágenes en 3D y detectar movimiento con alta precisión. Su capacidad para detectar instantáneamente la posición y los movimientos del cuerpo humano ha facilitado su integración en aplicaciones de robótica, en particular sistemas de visión por computadora y control de brazos robóticos. Esta cámara Kinect posee un costo relativamente bajo y es compatible con lenguajes de programación como Python y C lo hacen ideal para tareas como reconocimiento de gestos, seguimiento de objetos y reconstrucción 3D.

La cámara Kinect del tipo 360° (véase figura 19), de la marca de Xbox por Microsoft, tiene un servomotor de inclinación monitoreado el cual le permite ajustar el ángulo de la cámara hasta 27°, un sensor de profundidad de captación de 1,2 – 3,5 metros, micrófonos múltiples

y un procesador personalizado que ejecuta software propio el cual permite la captura de movimiento de cuerpo completo en 3D, reconocimiento facial y reconocimiento de voz.

Este tipo de cámara, la cual mide 30 cm de ancho, tiene un cable que se bifurca en dos, una para la conexión a la corriente y otro para conectar mediante USB a una consola u ordenador en el cual se pueda enlazar para su posterior visualización de la persona.



Figura 19. Cámara Kinect 360° [Fuente: Autor].

El sensor de profundidad 3D consta de un proyector de infrarrojos y un sensor CMOS monocromático. Los sensores infrarrojos pueden capturar datos de vídeo 3D en cualquier condición de iluminación. También funciona con resolución VGA (640 x 480), profundidad de 16 bits, 30 fotogramas por segundo y ofrece 2048 niveles de sensibilidad. Los proyectores de profundidad que utilizan sensores infrarrojos (retícula izquierda) y sensores CMOS monocromáticos (retícula derecha) calculan la distancia en función del tiempo que tarda la luz en reflejarse.

Las cámaras RGB que contiene, se utilizan para capturar resolución espacial, es decir, imágenes en color, y transmitir datos con una frecuencia de actualización de 30 fotogramas por segundo. En la tabla 1, se muestran otras características técnicas que tiene este modelo de cámara Kinect:

Atributo	Parámetro
Campo de visión horizontal	57°
Campo de visión vertical	43°
Rango de inclinación física	± 27°
Data Streams	Flujo de datos
Audio	16-bit a 16 kHz
Bit de color	640 × 480 32-bit - 30fps
Bit de profundidad	320 × 240 a 16 bits - 30fps

Tabla 1. Atributos y parámetros adicionales de cámara Kinect 360°.

El reconocimiento de imagen con esta cámara depende específicamente de dos elementos, el proyector y la cámara de infrarrojos VGA. Para comprender la distancia de cada píxel en la imagen de profundidad, el emisor de infrarrojos emite una serie de puntos. Luego, la cámara infrarroja localiza la constelación y Kinect calcula la diferencia (la diferencia entre la posición del punto proyectado y su posición en la proyección) para cada píxel, esto se llama cámara de luz estructurada. A partir de estos datos, Kinect realiza una serie de filtros para calcular qué es y qué no es una persona. El sistema utilizará pautas como "una persona tiene una cabeza, dos piernas y dos brazos" para distinguirla de otros elementos que podrían encontrarse en un sofá o en un campo deportivo.

3.1.1.2. Arduino UNO

La placa Arduino UNO (véase figura 20) es una placa de desarrollo ampliamente utilizada para proyectos de electrónica y robótica debido a su versatilidad, asequibilidad y facilidad de uso. Basado en el microcontrolador ATmega328P que permite el control y gestión de diversas señales mediante entradas y salidas digitales y analógicas, así como la capacidad de interactuar con sensores, actuadores y módulos externos, lo que lo hace ideal para implementar sistemas de control en brazos robóticos, lo que permite programar movimientos precisos y respuestas a estímulos ambientales.

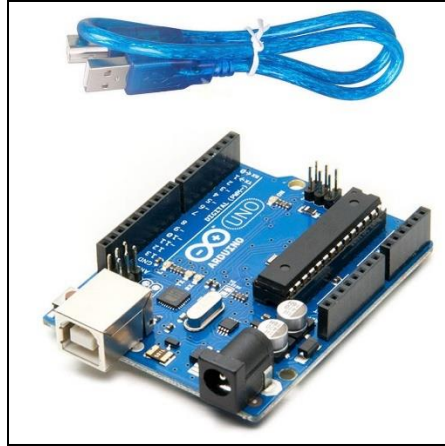


Figura 20. Arduino UNO [70].

Arduino UNO maneja un voltaje de operación de 5V y pines digitales (véase figura 21) de entrada/salida (I/O); 14 de los cuales 6 pueden ser usados como salidas PWM. Dentro de estos 14 pines, los pines usados que se tendrán en consideración para el proyecto serán los pines 10 y 11, los cuales, generalmente se configuran como salidas digitales que se utilizan para controlar señales específicas, como iniciar un actuador o enviar pulsos a un dispositivo externo.

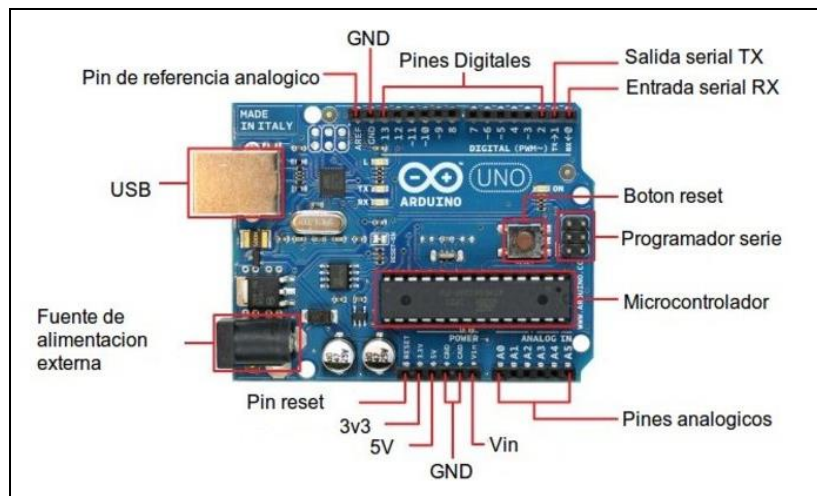


Figura 21. Características de placa Arduino UNO [71].

Por otro lado, los pines RX y TX son fundamentales para la comunicación serie, permitiendo transferir datos entre el Arduino y otros dispositivos como módulos inalámbricos, sensores avanzados u ordenadores que utilicen el protocolo UART. Además, el pin de alimentación

VCC proporciona la energía necesaria para el funcionamiento adecuado del sistema y los periféricos conectados y el pin de tierra GND, el cual asegura que el circuito tenga un punto común de referencia para el flujo de corriente.

En la tabla 2, se muestran más especificaciones técnicas de relevancia que tiene la tarjeta Arduino UNO:

Atributos	Parámetros
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (límites)	6-20V
Pines analógicos de entrada	6 (A0 a A5)
Corriente máxima por pin I/O	40 mA
Corriente máxima del pin 3.3V	50 mA
SRAM	2 KB
EEPROM	1 KB
Frecuencia de reloj	16 MHz
Puerto de comunicación	UART, SPI, I2C
Conectividad USB	USB tipo B
Conector de alimentación	Jack de 2.1 mm con pin positivo en el centro
Dimensiones	68.6 mm x 53.4 mm
Peso	Aproximadamente 25 g

Tabla 2. Atributos y parámetros de controlador Arduino UNO.

3.1.1.3. Brazo robótico X-Arm 1S

El brazo robótico X-Arm 1S, de tipo comercial, es de clase antropomórfico, cuenta con 5 grados de libertad, es un brazo robótico de metal completo con de 6 ejes o articulaciones. Este puede ser controlado de forma manual mediante comandos inalámbricos desde un joystick o mediante comandos enviados desde un ordenador. El robot también cuenta con una pinza o garra, una base giratoria y una placa controladora de servos la cual permitirá la adquisición de datos de los giros de las articulaciones (véase figura 22). También tiene ventosas para dar mejor estabilidad y evitar la caída de la estructura por movimientos fuertes ocasionados.

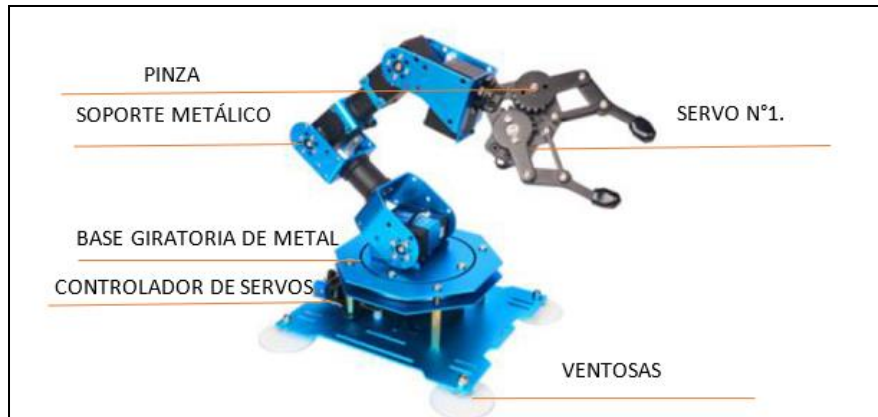


Figura 22. Partes del brazo robótico [72].

En la figura 23, se pueden visualizar las dimensiones de los diferentes componentes del brazo robótico en su posición extendida (home) y se pueden ver también las dimensiones de su garra y de la estructura, así como el dimensionamiento de la base donde se encuentra el Servo ID 6.

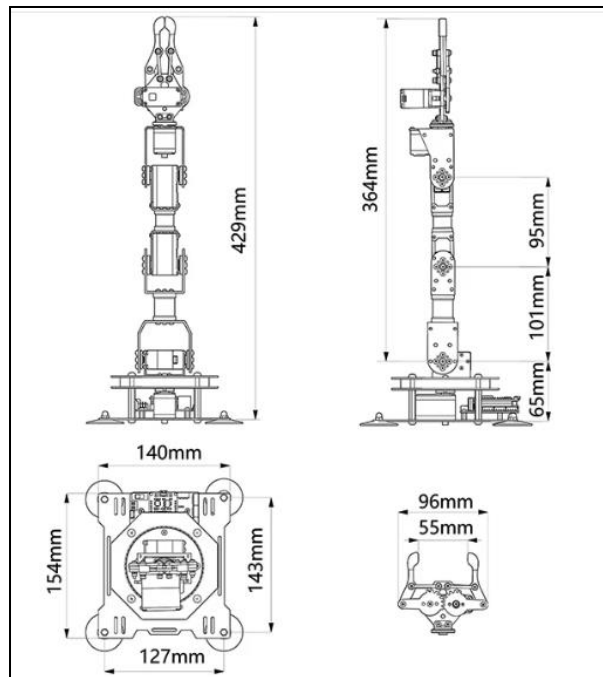


Figura 23. Dimensiones del brazo robótico [72].

A continuación, se va a explicar ciertas consideraciones de los diferentes elementos que tiene el brazo robótico.

Actuadores de articulaciones

Todas las articulaciones del brazo robótico están compuestas por servomotores, estos servomotores son de distintos tipos rango de rotación (previamente ya configurados en su programación), dichos rangos a su vez pueden ser modificados. En la tabla 3, se puede ver los ángulos de rotación de cada servo, el número de servo, el tipo y la locación de éstos en el brazo robótico.

Servo ID	Tipo	Locación	Ángulos
1	Servo para garra	Garra	-
2	LX - 15D	Muñeca	0-270°
3		Cuerpo	0-270°
4			0-270°
5	LX-225		0-270°
6	LX-15D	Plataforma	0-320°

Tabla 3. *Locación y ángulos de los servos en el brazo robótico.*

El brazo robótico X-Arm 1S cuenta con 6 servos ubicados a lo largo de su estructura, su conexión es de tipo bus, en la figura 24, se muestra la ubicación de los 6 servos instalados en el brazo robótico con sus respectivos ID.



Figura 24. Distribución de los servos en el brazo robótico [72].

Los servos 2, 3, 4 y 6 son del tipo LX-15D (véase figura 25) y están diseñados con engranajes de material de aleación con un conector tipo PH2.0-3P que funcionan con un voltaje de 6 V a 8,4 V. Estos servos poseen un grado de rotación de 0° a 270° ya preestablecido; sin embargo, estos se pueden ajustar de acuerdo con las necesidades del usuario.



Figura 25. Servo LX-15D [72].

En la tabla 4, se muestran otras especificaciones técnicas de relevancia del servo LX-15D.

Atributo	Parámetro
Tamaño	44,02 x 22,92 x 35,12 mm
Material de engranajes	Engranaje de aleación
Esfuerzo de torsión	15 kg.cm 6 V 17 kg.cm 7,4 V
Velocidad	0,23 s/60° 7,4 V
Control método	Comando serial UART
Comunicación tasa de baudios	115200
Retroalimentación de datos	Temperatura, voltaje, posición
Indicador	Luz RGB
Almacenar	Guardar la configuración del usuario al apagar
Sin carga actual	100 mA

Tabla 4. Atributos y parámetros del Servo LX-15D.

El servo 5 es del tipo LX-225 (véase figura 26) y tiene engranajes de material de aleación con un conector tipo PH2.0-3P que funciona con un voltaje de 6 V a 8,4 V. Estos poseen un grado de rotación de 0° a 270° ya preestablecido; sin embargo, este se puede ajustar de acuerdo con las necesidades del usuario.



Figura 26. Servo LX-225 [72].

En la tabla 5, se muestran otras especificaciones técnicas de relevancia del servo LX-225.

Atributo	Parámetro
Tamaño	40 x 20,14 x 51,1 mm
Material de engranajes	Engranaje de aleación
Esfuerzo de torsión	25 kg.cm 7,4 V
Velocidad	0,22 s/60° 7,4 V
Control método	Comando serial UART
Comunicación tasa de baudios	115200
Retroalimentación de datos	Temperatura, voltaje, posición
Indicador	Luz RGB
Almacenar	Guardar la configuración del usuario al apagar
Sin carga actual	100 mA

Tabla 5. Atributos y parámetros del Servo LX-225.

Tarjeta controladora LSC-6

El brazo robótico X-Arm 1S viene equipado con una tarjeta controladora LSC-6 (véase figura 27) basada en Arduino Uno o una versión similar, la cual está encargada de leer los valores dados por los diferentes servomotores que corresponden a los ángulos de rotación del brazo robótico y también está encargada de enviar las señales a los servomotores para realizar los diferentes giros. Este tipo de controlador es común en brazos robóticos de gama baja o educativa, ya que permite un control flexible y de fácil programación. El firmware que viene instalado en el controlador de estos brazos está diseñado para comunicarse con un software externo o recibir comandos a través de interfaces como USB o UART (serial).

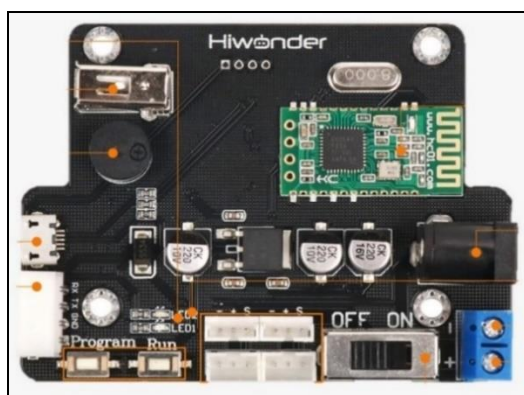


Figura 27. Controlador para servo LSC – 6 [72].

Este controlador integrado en la parte inferior de la base giratoria del brazo robótico X-Arm 1S (véase figura 28) se encargará de gestionar y regular el funcionamiento de todo el sistema. A esta tarjeta se debe conectar los diferentes servomotores de las articulaciones mediante un bus en serie (uno tras otro).

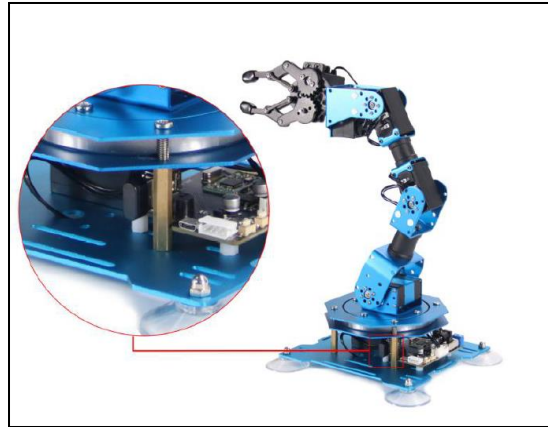


Figura 28. Visión de controlador en el brazo robótico [72].

3.1.2. COMPONENTES LÓGICOS

A continuación, se va a realizar una pequeña descripción de ciertos elementos de los entornos de programación utilizados en el desarrollo de este proyecto. Dentro de estos elementos se encuentra el entorno de programación LabVIEW y Arduino los cuales permitirán el gestionar los datos recibidos y enviados tanto de la cámara Kinect como del brazo robótico, en la obtención de ángulos del sistema de imitación por visión artificial.

3.1.2.1. Entorno de programación LabVIEW

LabVIEW es una plataforma de programación de alto nivel está diseñada para simplificar tareas complejas y proporciona una interfaz intuitiva para el desarrollo de aplicaciones de control e instrumentación, esta plataforma usa lenguaje de programación gráfico de nivel alto. Cuenta con herramientas para todo tipo de funciones, gracias a esto permite realizar interfaces de usuario presentables y funcionales.

Presenta dos pantallas interactivas (véase figura 29), la primera llamada Front Panel (Panel Frontal) donde el usuario puede interactuar con el programa, incluir botones, sliders, interruptores, etc. También esta parte permitirá la visualización de indicadores con gráficos, números, luces, los cuales serán parte del estado del sistema y permitirán controlar las acciones de la aplicación diseñada. La segunda parte llamada Block Diagram (Diagrama de Bloques), es donde se construirá la parte lógica del programa, también permite la conexión de VIs (Virtual Instruments), que son bloques de código para realizar tareas específicas y flujos de datos.

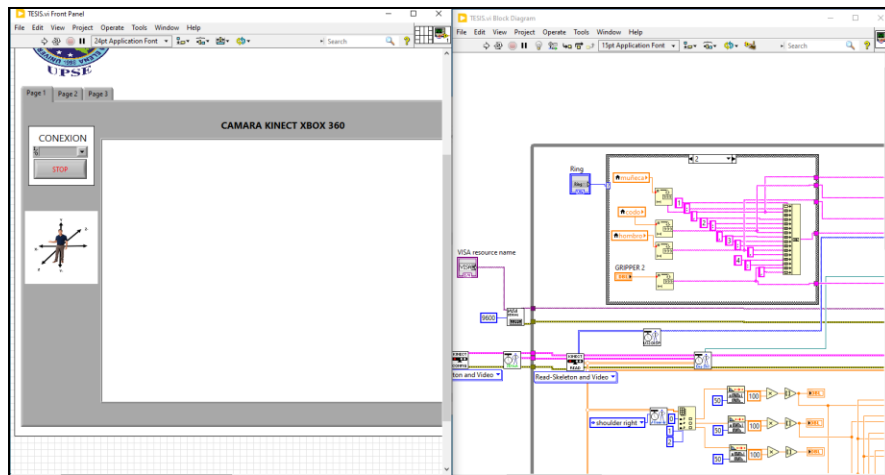


Figura 29. Formato de HMI del sistema de imitación [Fuente: Autor].

Cuando se instala el programa LabVIEW, se instalan a su vez diferentes Toolbox, los cuales son un conjunto específico de herramientas o funciones diseñadas para realizar tareas específicas en LabVIEW, las cuales se centran en proporcionar utilidades preconfiguradas para facilitar la programación en matemáticas avanzadas, procesamiento de datos o generación de gráficos, también incluyendo bibliotecas de bloques de funciones (VI) predefinidos.

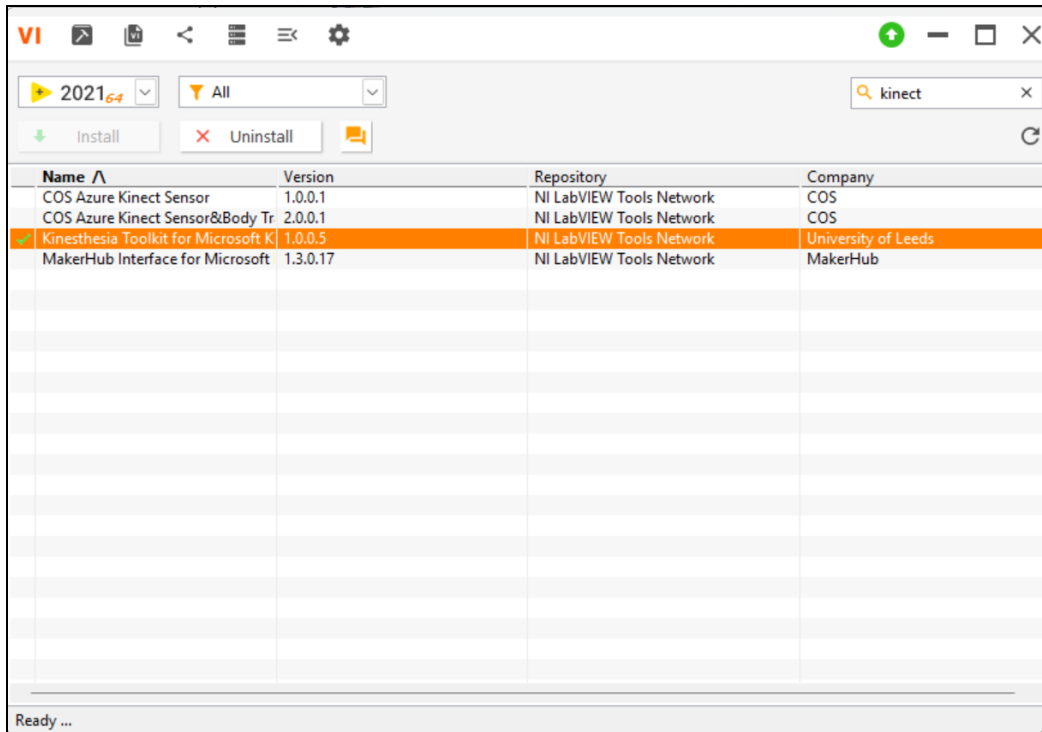















Figura 30. Librería Kinect en VI Package Manager - LabVIEW [Fuente: Autor].

El programa VI Package Manager mostrado en la figura 30, permite la instalación de librerías de terceros, en el caso de este proyecto se ha utilizado la librería SDK Kinect (véase figura 30). Esta biblioteca incluye varios bloques que permiten el enlace entre una cámara Kinect de Xbox 360 hacia el programa LabVIEW. En la tabla 6, se presenta una pequeña descripción de los bloques que se han empleado en este proyecto:

 Configure	El VI de configuración permite al desarrollador de LabVIEW elegir las opciones disponibles para los flujos de datos de la cámara Kinect, como son la resolución de la imagen de vídeo y de profundidad.
 Initialise	Este VI se utiliza para crear una instancia del sensor Kinect en la memoria de la computadora. Tanto la terminal de error como el clúster de referencia deben estar conectados al VI de configuración.
 Kinect Close	Este VI se utiliza para cerrar todas las referencias abiertas que se crean durante el funcionamiento del sensor Kinect.

 <p>Read</p>	<p>El bloque Read toma los flujos de datos de Kinect y los procesa y organiza en un formato en el que se puedan procesar o visualizar posteriormente. El VI de lectura debe ubicarse dentro de un bucle while de programas para permitir la visualización y el procesamiento continuos de los datos de Kinect.</p>
 <p>Initialise 3D Skeleton</p>	<p>Este bloque prepara el escenario para el trazado del esqueleto y es necesario antes de poder renderizar el esqueleto 3D. La referencia generada debe pasarse al VI Render 3D Skeleton.</p>
 <p>Display Colour Data (U32 to ...</p>	<p>El bloque convierte la matriz U32 sin procesar en un formato de imagen 2D que se puede visualizar mediante el control de imagen 2D de LabVIEW, permitiendo manipular los datos sin procesar antes de la renderización.</p>
 <p>Displacement and Distance ...</p>	<p>Este VI devuelve la distancia y el desplazamiento absoluto entre dos articulaciones o una articulación y una coordenada X, Y, Z personalizada. Para devolver la distancia o el desplazamiento entre dos articulaciones, la matriz de esqueleto y el número de las dos articulaciones deben conectarse a los terminales correctos y el valor booleano de coordenadas personalizadas debe establecerse en falso.</p>
 <p>2D Picture</p>	<p>Permite visualizar mediante una pantalla, imágenes en 2 dimensiones.</p>
 <p>3D Picture</p>	<p>Permite visualizar mediante una pantalla, imágenes en 3 dimensiones.</p>
 <p>Joint Coordinates</p>	<p>Este VI acepta un valor numérico entre 0 y 19 y devolverá las coordenadas X, Y, Z de la articulación seleccionada cuando la matriz de esqueleto esté conectada al terminal correspondiente.</p>
 <p>Median Filter</p>	<p>Aplica un filtro mediano de rango a X.</p>
 <p>acos</p>	<p>Calcula el arco coseno de x.</p>
 <p>Expression Node</p>	<p>Este nodo permite calcular expresiones que contengan una sola variable.</p>


 <p>Index Array</p>	<p>Devuelve el elemento o sub matriz de una matriz de n dimensiones en el índice.</p>

Tabla 6. Bloques para configuración de cámara Kinect en LabVIEW [Fuente: Autor].

Dentro de la librería SDK se tiene un apartado para la captación de las articulaciones del usuario, dada por el patrón de puntos referenciales proporcionado por la misma librería SDK de la cámara Kinect en LabVIEW, que permitirá obtener los movimientos generados por el usuario como puntos referenciales necesarias para la obtención de datos. A partir de esta información se organiza la identificación de las partes del cuerpo humano y se transforma en un esqueleto (véase figura 31). La librería viene precargada con más de 20 posiciones comunes del cuerpo humano, por lo que, si algún movimiento bloquea una parte del esqueleto de la cámara, Kinect automáticamente llenará los espacios y creará múltiples huesos, pero elige uno según la experiencia. El sistema hace todo esto de forma continua a 30 fotogramas por segundo [FPS], y hay que estar a unos dos metros de distancia para reconocerlo.

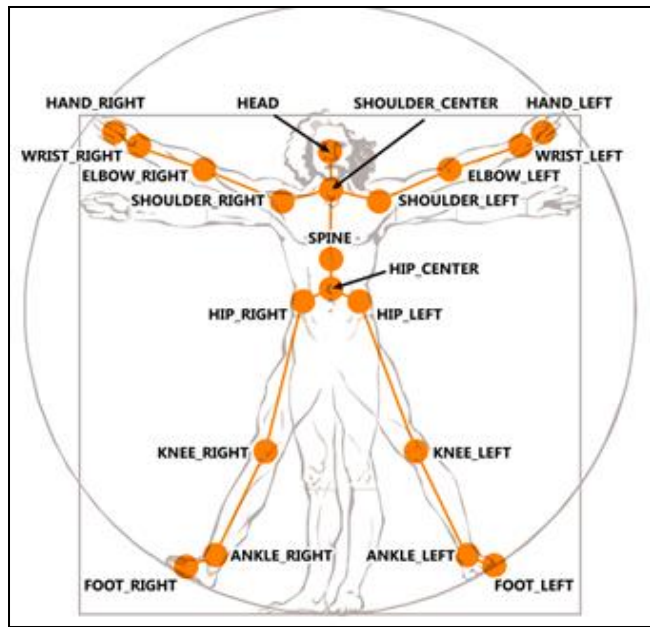


Figura 31. Puntos referenciales en articulaciones del cuerpo humano [73].

3.1.2.2. Entorno de desarrollo integrado Arduino IDE

Arduino tiene su propio entorno de desarrollo integrado (véase figura 32) para comunicarse; a través de varios métodos, entre ellas se encuentra los protocolos de comunicación serial (UART), I2C y SPI, los cuáles requieren un código precargado en el programa de Arduino que permite enviar y recibir datos.

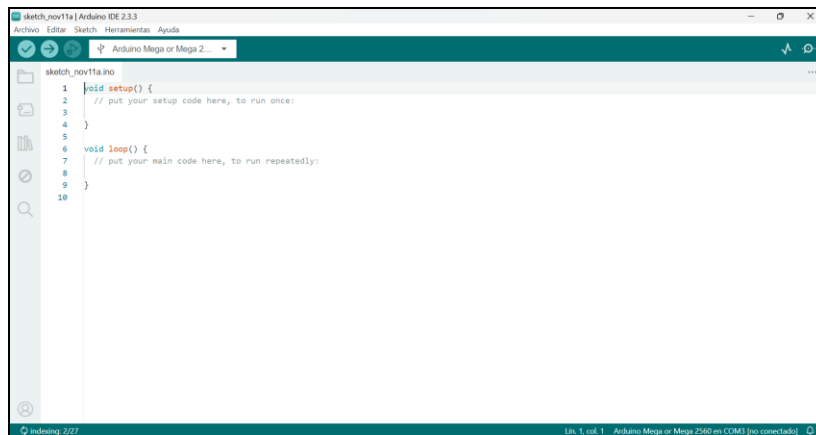


Figura 32. Ventana inicial del Arduino IDE [Fuente: Autor].

Tiene un apartado que permitirá la revisión en tiempo real de un monitoreo serial (véase figura 33), para la comunicación en serie entre la computadora y la placa Arduino correspondiente, este permitirá enviar y recibir datos que serán útiles para depuración o control en tiempo real.

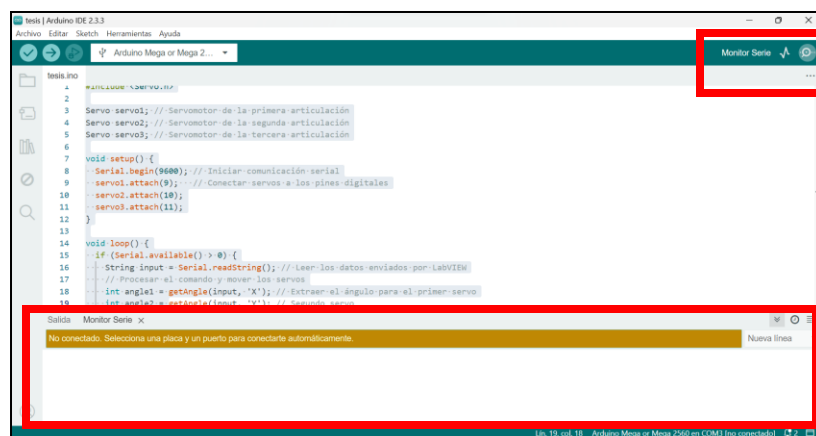


Figura 33. Monitoreo serial en Entorno Arduino IDE [Fuente: Autor].

La figura 34, muestra la interfaz del IDE de Arduino con las áreas y funciones principales etiquetadas para facilitar la comprensión, las cuales se dará la descripción de cada parte a la que apunta cada flecha a continuación:

- Barra de menú: contiene opciones de entorno importantes, como Archivo, Editar, Aplicaciones, Herramientas y Ayuda para que pueda realizar tareas como abrir y guardar bocetos, paletas y puertos, administrar bibliotecas y acceder a archivos.
- Accesos directos: con botones debajo de la barra de menú que le permiten realizar rápidamente acciones comunes como inspeccionar el código, cargarlo en el panel de desarrollo, crear un nuevo proyecto, abrir un archivo existente o guardar el archivo actual.
- Área de pestañas: muestra pestañas correspondientes a los distintos archivos que componen el proyecto lo que facilita la organización y navegación entre múltiples bocetos o módulos de código.
- Área de edición: es el lugar principal para escribir y editar el código original que incluye resaltado de sintaxis, identificación de errores y soporte de comentarios para mejorar la legibilidad del código.
- Área de estado: indica el progreso de la compilación, carga del código y la información de estado del proceso relacionada.
- Consola de salida: muestra mensajes generados durante la verificación, compilación y carga del boceto; incluidos errores, advertencias y confirmaciones de éxito que pueden ayudar con la depuración.
- Barra de información: proporciona información detallada sobre la placa seleccionada, el microcontrolador utilizado, los puertos y otra información relevante en el entorno de desarrollo.

El entorno de programación Arduino IDE también cuenta con un apartado que permite gestionar librerías (véase figura 35) éstas se podrán descargar según las necesidades del proyecto a programar. Las librerías son colecciones de archivos de código prescrito que proporcionan funciones específicas para interactuar con hardware o realizar tareas comunes. Estas simplifican el desarrollo al abstraer la lógica compleja detrás de ciertas operaciones.

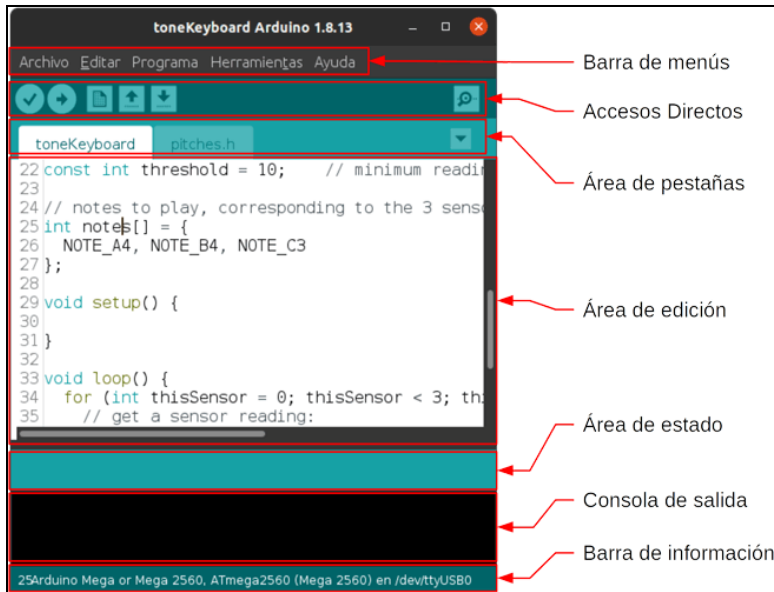


Figura 34. Apartados en Arduino IDE [74].

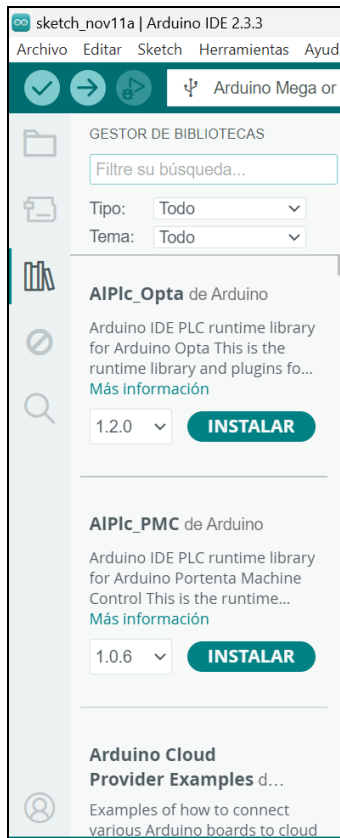


Figura 35. Gestor de biblioteca en el programa Arduino IDE [Fuente: Autor].

El fabricante del brazo robótico X-Arm 1S pone a disposición un conjunto de librerías (véase figura 36) diseñadas específicamente para facilitar el control preciso de los servomotores inteligentes integrados en el dispositivo. Las librerías proporcionadas no solo simplifican la programación y comunicación con los actuadores, sino que también permiten configurar parámetros claves, como los ángulos de rotación, velocidad y posición.

En este proyecto, se utilizaron herramientas de software para garantizar un control preciso y óptimo de los movimientos del brazo robótico, asegurando una integración fluida con otros módulos del sistema, como la interfaz de usuario y los dispositivos de entrada de datos.

Las librerías usadas en el proyecto son las siguientes:

```
#include "LobotServoController.h" // Librería del controlador de servos
#include <SoftwareSerial.h>       // Librería para SoftwareSerial
```

Figura 36. Librerías usadas en proyecto [Fuente: Autor].

La librería “LobotServoController.h” está diseñada para el control de servomotores el cual proporciona funciones específicas; tal como, definir posiciones, velocidad y ángulos de rotación. La segunda librería “SoftwareSerial.h” es un repositorio estándar de Arduino que permite crear puertos seriales adicionales en los pines digitales del microcontrolador, el cual es útil cuando el puerto serial hardware (conectado generalmente al puerto USB) ya está en uso, permitiendo la comunicación con otros dispositivos, en este caso el control de servos.

3.2. DISEÑO DE LA PROPUESTA

En este apartado, se presentan las características y especificaciones que tiene el sistema de imitación, indicando los pines exactos y los tipos de comunicación usados, además se indicarán mediante diagramas de flujo, la lógica de programación que se ha diseñado tanto la adquisición de las imágenes de la Kinect y el controlador que implementará la cinemática del brazo robótico X-Arm 1S, lo cual posteriormente se utilizó en la parte de programación en LabVIEW.

3.2.1. Diseño e implementación del Hardware

En este apartado, se dará detalle la arquitectura del sistema de imitación para controlar el brazo robótico X-Arm 1S utilizando un sensor Kinect, una placa Arduino y una computadora con software de visión artificial implementado en LabVIEW.

En la figura 37, se muestra como el sensor de la cámara Kinect detecta los movimientos del cuerpo humano. Esta cámara está conectada a la computadora mediante protocolo de comunicación USB 2.0, y utiliza el método de transferencia de datos Bulk Transfer dentro de este estándar para enviar información capturada. Los datos generados por la Kinect son procesados en la computadora para extraer coordenadas y ángulos de las articulaciones humanas mediante técnicas de visión artificial, utilizando una interfaz desarrollada en LabVIEW.

LabVIEW toma estos valores, los procesa y los analiza para enviarlos posteriormente a la placa Arduino, que está conectado mediante protocolo UART. El Arduino actúa como intermediario para controlar el movimiento del brazo robótico, transmitiendo los valores de la cinemática (ángulos calculados). De esta manera, el brazo robótico replica las articulaciones del cuerpo humano, permitiendo que los gestos de los brazos de una persona sean imitados en tiempo real.

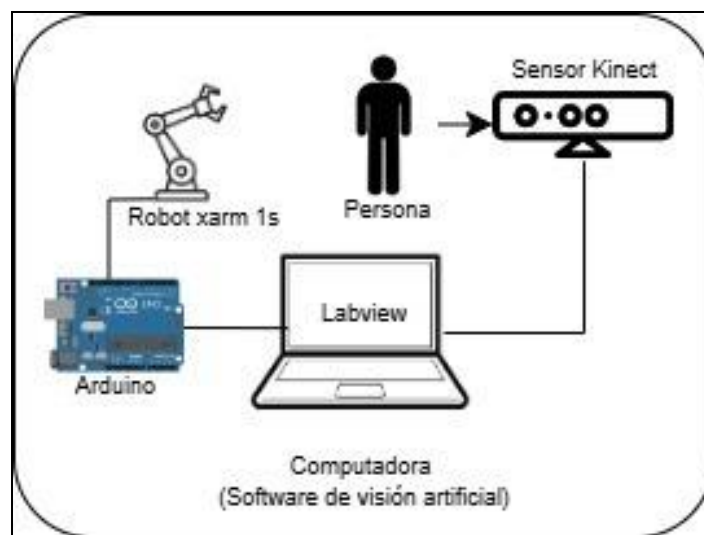


Figura 37. Diseño e implementación del Hardware en el sistema de imitación [Fuente: Autor].

3.2.1.1. Conexión de la cámara Kinect

La cámara Kinect posee una conexión eléctrica (véase figura 38) mediante una fuente de alimentación externa para integrarla con un ordenador utilizando un cable con conector USB. Posee un cable bifurcado el cual se divide en tres partes principales: el conector USB, el adaptador de 12 V y el conector Kinect. La fuente de alimentación proviene del conector USB (5 V, Datos, Datos, GND) y se conecta directamente al conector Kinect. Además, se utilizó un adaptador de corriente de 12 V y 1,5 A con su terminal positivo (12 V) y tierra (GND) conectados al conector Kinect para proporcionar energía a la cámara.

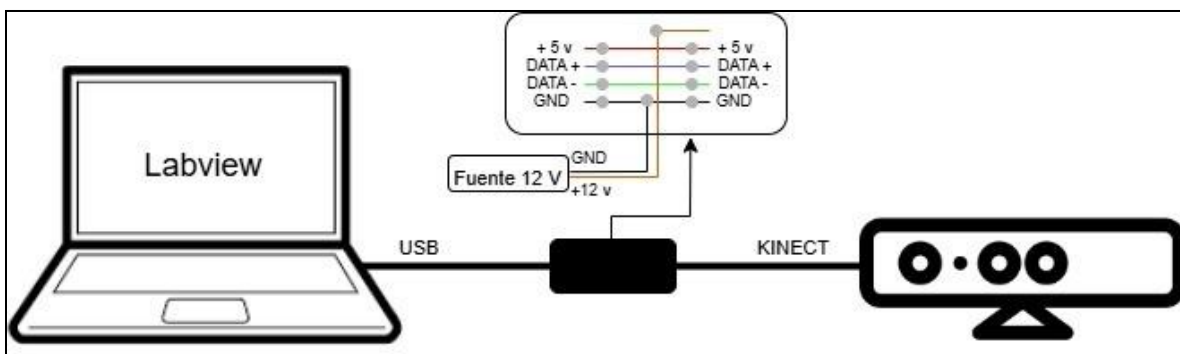


Figura 38. Conexión entre cámara Kinect y ordenador [Fuente: Autor].

3.2.1.2. Conexión de dispositivos con el brazo robótico

En la figura 39, se muestra los elementos que se conectan con el brazo robótico, el cual comienza desde una computadora ejecutando LabVIEW, que envía comandos y señales al brazo robótico mediante un Arduino UNO, la conexión entre estos dispositivos se establece a través de un puerto USB utilizando el protocolo UART. La tarjeta Arduino actúa como intermediario, transmitiendo señales a una tarjeta controladora de servomotores del brazo robótico, que están alimentados por una fuente externa, controlado mediante un interruptor. Este sistema posee una comunicación bidireccional.

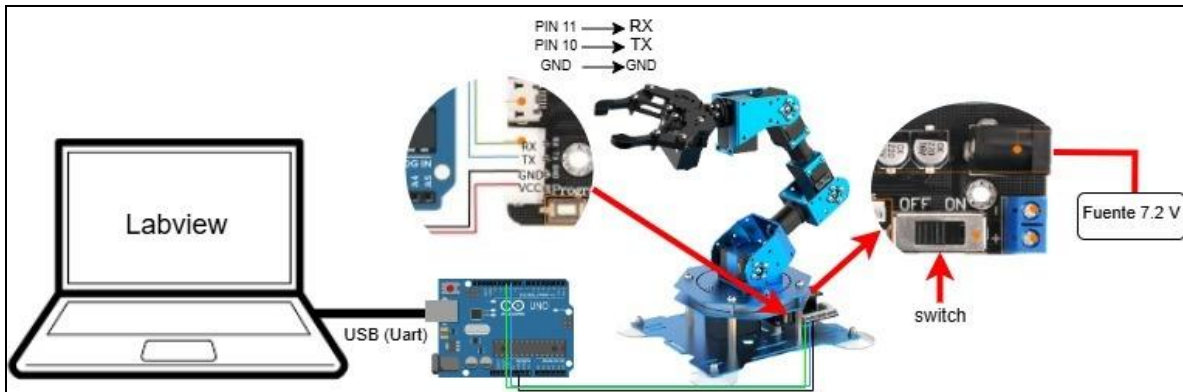


Figura 39. Conexión del brazo robótico [Fuente: Autor].

El brazo robótico X-Arm 1S tiene conectado 6 servomotores, los cuales están en tipo serie (uno tras otro) o tipo bus. Tiene integrada en su base, una tarjeta controladora LSC – 6 (véase figura 40), la cual tiene puertos para la conexión de los servomotores inteligentes, en este caso, al estar conectados en serie, el servo que va conectado a la tarjeta será el Servo ID6 LX-15D.

El controlador LSC-6 ubicada en la parte inferior del brazo robótico (véase figura 40), posee un puerto de comunicación serial, el cual permite establecer comunicación con otros dispositivos, como microcontroladores o computadoras, utilizando los pines RX (Recepción), TX (Transmisión), GND (Tierra) y VCC (Alimentación).

También esta tarjeta tiene un conector de alimentación de 7.2 V, el cual proporciona energía a la placa y a los dispositivos conectados a través de un conector especializado. La tensión de entrada de 7.2 V es común en aplicaciones robóticas, ya que es adecuada para alimentar motores y electrónica sin generar demasiada disipación de calor. También tiene un interruptor (switch), el cual permite encender o apagar el sistema manualmente.

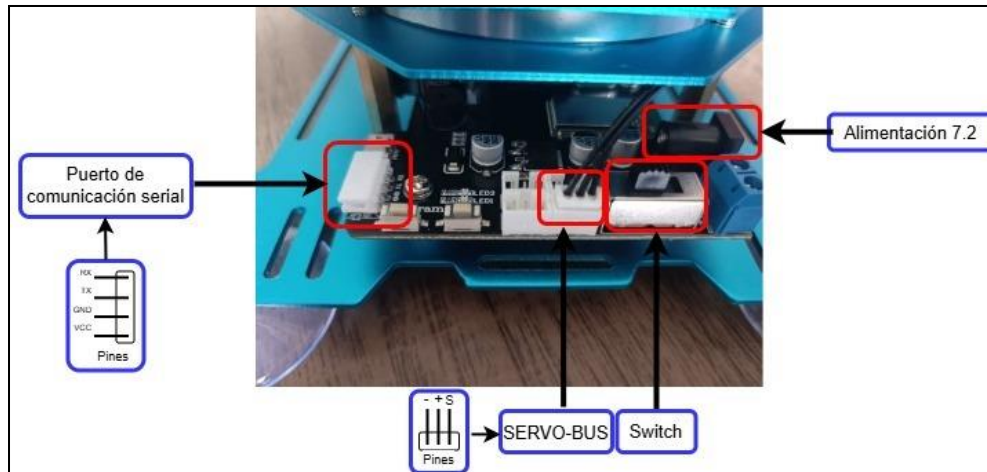


Figura 40. Controlador instalado en el brazo robótico [Fuente: Autor].

En la figura 41, en el centro inferior de la placa hay un conector de varios pines para controlar servomotores, se muestra conectado a tres servomotores mediante cables, representando señales de control, voltaje positivo y tierra respectivamente.

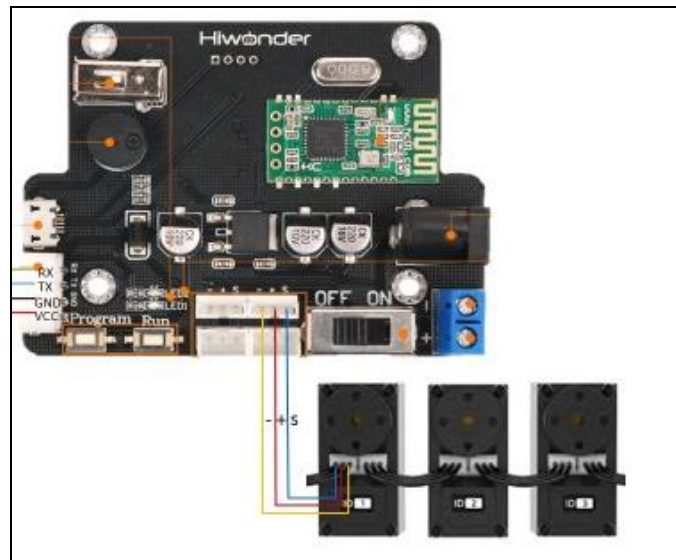


Figura 41. Conexión de tarjeta LSC-6 con servos [Fuente: Autor].

Dicha tarjeta, tiene también un puerto para la conexión RX, TX, VCC y GND para una tarjeta controladora externa, en este caso será para un Arduino UNO (véase figura 41).

En la figura 42, se muestra una interconexión entre una placa Arduino Uno y la tarjeta LSC-6. Esta tarjeta actúa como el núcleo del procesamiento lógico, ejecutando un programa

cargado previamente, el cual envía y recibe señales para controlar los periféricos conectados. Los pines utilizados en el Arduino son 10 y 11, estos pines están conectados a los pines de transmisión y recepción (TX y RX) del controlador LSC-6, permitiendo comunicación serial, el cual establece un canal bidireccional de datos entre ambas placas. El Arduino Uno envía instrucciones, mientras que la placa LSC-6 transmite estados o confirma ejecuciones.

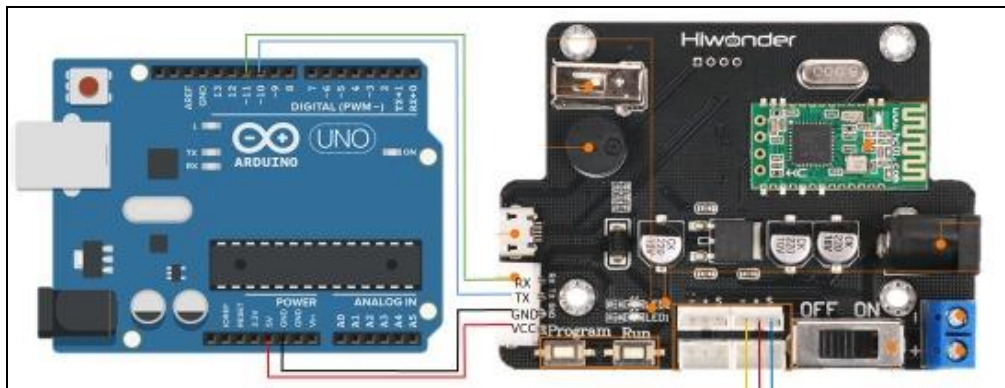


Figura 42. Conexión de tarjeta LSC-6 a controlador Arduino UNO [Fuente: Autor].

3.2.1.3. Modelo cinemático del brazo robótico X-Arm 1S

Puesto que el brazo robótico X-Arm 1S (véase figura 43) tiene grados de libertad, es factible el cálculo matemático por cinemática directa, aplicando el método de Denavit Hartenberg

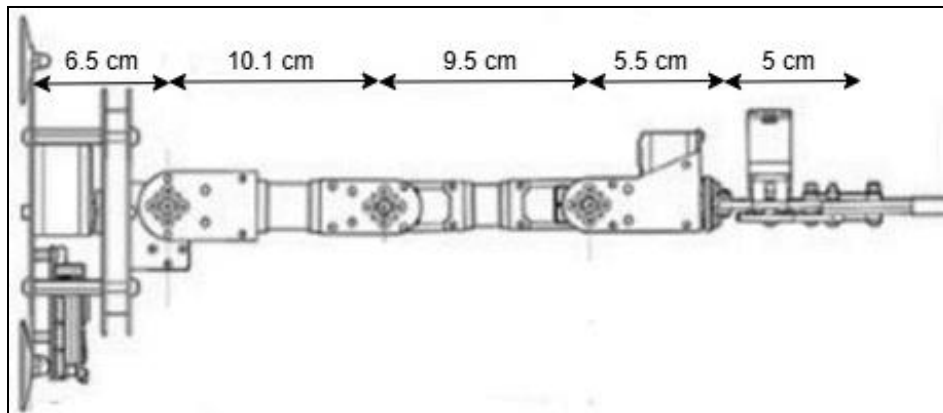


Figura 43. Brazo robótico X-Arm 1S [72].

La figura 44, se puede observar que el brazo robótico posee 6 grados de libertad las cuales son articulaciones giratorias, 5 de estas articulaciones conforman el cuerpo del brazo y 1 es usada para el gripper.

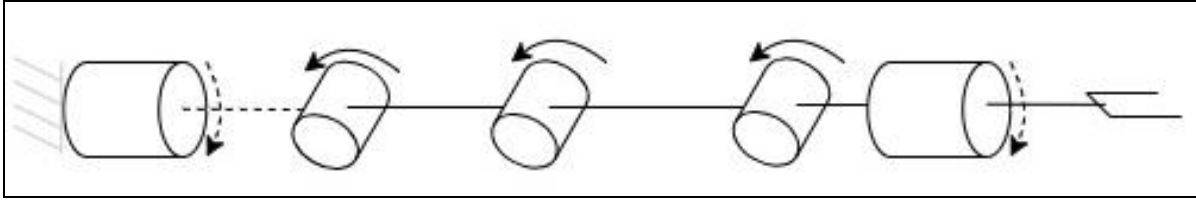


Figura 44. Articulaciones en brazo robótico [Fuente: Autor].

A continuación, se dará una explicación del cómo se encuentra representada las 6 articulaciones con su respectivo sentido de giro (véase figura 45), de las cuales se toma las siguientes consideraciones:

- No se puede desplazar en el eje Y.
- No se puede rotar con respecto al eje Y.
- Los ejes Z_i coinciden con los ejes de acción de las articulaciones (Rotación y Traslación en el eje z).
- Los ejes X_i deben de satisfacer $X_i \perp Z_i$ y $X_i \perp Z_{i-1}$ (Propiedad de doble perpendicularidad).

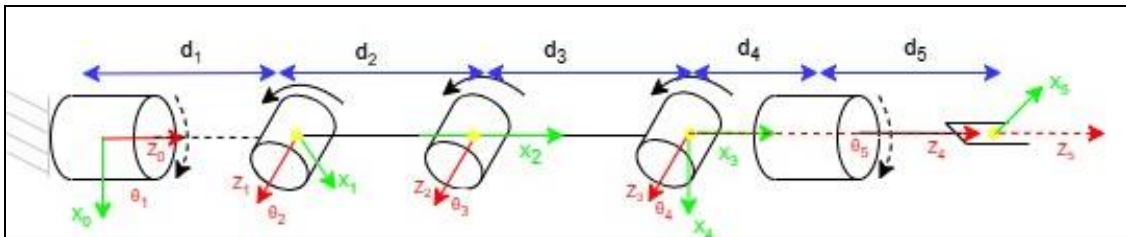


Figura 45. Limitaciones en cada articulación del brazo robótico [Fuente: Autor].

En la figura 45, se muestra el sistema de referencia en el cual se asignan parámetros, los cuales son usados para llenar la tabla de Denavit Hartenberg. Entre estos parámetros se tienen los siguientes:

Parámetros para el tramo H_1^0

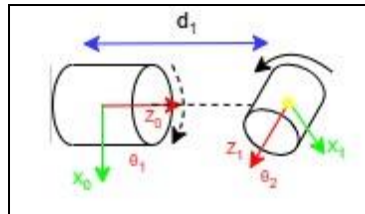


Figura 46. Parámetro H_1^0 - Método Denavit Hartenberg [Fuente: Autor].

Parámetros para el tramo H_2^1

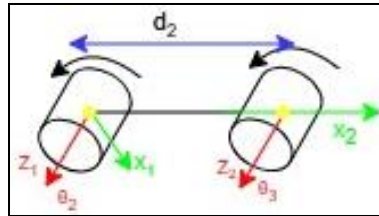


Figura 47. Parámetro H_2^1 - Método Denavit Hartenberg [Fuente: Autor].

Parámetros para el tramo H_3^2

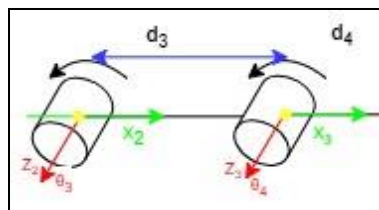


Figura 48. Parámetro H_3^2 - Método Denavit Hartenberg [Fuente: Autor].

Parámetros para el tramo H_4^3

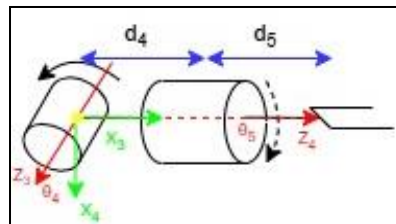


Figura 49. Parámetro H_4^3 - Método Denavit Hartenberg [Fuente: Autor].

Parámetros para el tramo H_5^4

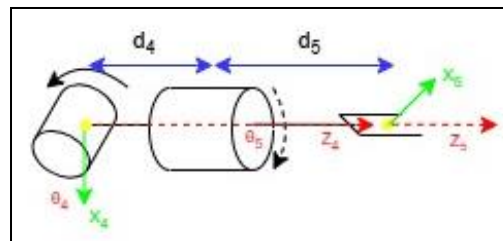


Figura 50. Parámetro H_5^4 - Método Denavit Hartenberg [Fuente: Autor].

Tabla de los parámetros

	Z_{i-1}		X_i	
	θ_i	d_i	a_i	α_i
H_1^0	θ_1^*	d_1	0	90
H_2^1	θ_2^*	0	d_2	0
H_3^2	θ_3^*	0	d_3	0
H_4^3	$\theta_4 + 90^*$	0	0	-90
H_5^4	θ_5	$d_4 + d_5$	0	0

	θ_x	d_x	a_z	α_z
H_1^0	θ_1^*	6.5	0	90
H_2^1	θ_2^*	0	10.1	0
H_3^2	θ_3^*	0	9.5	0
H_4^3	$\theta_4 + 90^*$	0	0	-90
H_5^4	θ_5	5.5 + 5	0	0

Matriz general de Denavit Hartenberg

$$H = \begin{bmatrix} \cos(\theta_i) & -\cos(\alpha_i)\sin(\theta_i) & \sin(\alpha_i)\sin(\theta_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\alpha_i)\cos(\theta_i) & -\sin(\alpha_i)\cos(\theta_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matriz de Transformación H_1^0

$$H_1^0 = \begin{bmatrix} \cos(\theta_1^*) & 0 & \sin(\theta_1^*) & 0 \\ \sin(\theta_1^*) & 0 & -\cos(\theta_1^*) & 0 \\ 0 & 1 & 0 & 6.5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matriz de Transformación H_2^1

$$H_2^1 = \begin{bmatrix} \cos(\theta_2^*) & -\sin(\theta_2^*) & 0 & 10.1\cos(\theta_2^*) \\ \sin(\theta_2^*) & \cos(\theta_2^*) & 0 & a_i\sin(\theta_2^*) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matriz de Transformación H_3^2

$$H = \begin{bmatrix} \cos(\theta_3^*) & -\sin(\theta_3^*) & 0 & 9.5\cos(\theta_3^*) \\ \sin(\theta_3^*) & \cos(\theta_3^*) & 0 & 9.5\sin(\theta_3^*) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matriz de Transformación H_4^3

$$H_4^3 = \begin{bmatrix} \cos(\theta_4 + 90^*) & 0 & -\sin(\theta_4 + 90^*) & 0 \\ \sin(\theta_4 + 90^*) & 0 & \cos(\theta_4 + 90^*) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matriz de Transformación H_5^4

$$H_5^4 = \begin{bmatrix} \cos(\theta_5^*) & -\sin(\theta_5^*) & 0 & 0 \\ \sin(\theta_5^*) & \cos(\theta_5^*) & 0 & 0 \\ 0 & \sin(\alpha_i) & 1 & 9.5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_5^0 = H_1^0 H_2^1 H_3^2 H_4^3 H_5^4$$

$$H = \begin{bmatrix} \cos(\theta_1^*) & 0 & \sin(\theta_1^*) & 0 \\ \sin(\theta_1^*) & 0 & -\cos(\theta_1^*) & 0 \\ 0 & 1 & 0 & 6.5 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos(\theta_2^*) & -\sin(\theta_2^*) & 0 & 10.1 \cos(\theta_2^*) \\ \sin(\theta_2^*) & \cos(\theta_2^*) & 0 & 10.1 \sin(\theta_2^*) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} *$$

$$\begin{bmatrix} \cos(\theta_3^*) & -\sin(\theta_3^*) & 0 & 9.5 \cos(\theta_3^*) \\ \sin(\theta_3^*) & \cos(\theta_3^*) & 0 & 9.5 \sin(\theta_3^*) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos(\theta_4 + 90^*) & 0 & -\sin(\theta_4 + 90^*) & 0 \\ \sin(\theta_4 + 90^*) & 0 & \cos(\theta_4 + 90^*) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} *$$

$$\begin{bmatrix} \cos(\theta_5^*) & -\sin(\theta_5^*) & 0 & 0 \\ \sin(\theta_5^*) & \cos(\theta_5^*) & 0 & 0 \\ 0 & \sin(\alpha_i) & 1 & 9.5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matriz final

$$H = \begin{bmatrix} R_{11} & R_{12} & R_{13} & R_x \\ R_{21} & R_{22} & R_{23} & R_y \\ R_{31} & R_{32} & R_{33} & R_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_{11} = \cos(90^\circ + \theta_4)\cos(\theta_2^* + \theta_3^*)\cos(\theta_1^*)\cos(\theta_5^*) - \sin(90^\circ + \theta_4)\sin(\theta_2^* + \theta_3^*)\cos(\theta_1^*)\cos(\theta_5^*) - \sin(\theta_1^*)\sin(\theta_5^*)$$

$$R_{12} = \sin(90^\circ + \theta_4)\sin(\theta_2^* + \theta_3^*)\cos(\theta_1^*)\sin(\theta_5^*) - \cos(90^\circ + \theta_4)\cos(\theta_2^* + \theta_3^*)\cos(\theta_1^*)\sin(\theta_5^*) - \cos(\theta_5^*)\sin(\theta_1^*)$$

$$R_{13} = -\sin(90^\circ + \theta_2^* + \theta_3^* + \theta_4)\cos(\theta_1^*)$$

$$R_x = 9.5\cos(\theta_1^*)\cos(\theta_2^* + \theta_3^*) - 9.5\cos(\theta_1^*)\sin(90^\circ + \theta_2^* + \theta_3^* + \theta_4) + 10.1\cos(\theta_1^*)\cos(\theta_2^*)$$

$$R_{21} = \cos(\theta_1^*)\sin(\theta_5^*) + \cos(90^\circ + \theta_4)\cos(\theta_2^* + \theta_3^*)\cos(\theta_5^*)\sin(\theta_1^*) - \sin(90^\circ + \theta_4)\sin(\theta_2^* + \theta_3^*)\cos(\theta_5^*)\sin(\theta_1^*)$$

$$R_{22} = \cos(\theta_1^*)\cos(\theta_5^*) - \cos(90^\circ + \theta_4)\cos(\theta_2^* + \theta_3^*)\sin(\theta_1^*)\sin(\theta_5^*) + \sin(90^\circ + \theta_4)\sin(\theta_2^* + \theta_3^*)\sin(\theta_1^*)\sin(\theta_5^*)$$

$$R_{23} = -\sin(90^\circ + \theta_2^* + \theta_3^* + \theta_4)\sin(\theta_1^*)$$

$$R_y = 9.5\sin(\theta_1^*)\cos(\theta_2^* + \theta_3^*) - 9.5\sin(\theta_1^*)\cos(\theta_2^* + \theta_3^* + \theta_4) + 10.1\sin(\theta_1^*)\cos(\theta_2^*)$$

$$R_{31} = \sin(90^\circ + \theta_2^* + \theta_3^* + \theta_4)\cos(\theta_5^*)$$

$$R_{32} = -\sin(90^\circ + \theta_2^* + \theta_3^* + \theta_4)\sin(\theta_5^*)$$

$$R_{33} = \cos(90^\circ + \theta_2^* + \theta_3^* + \theta_4)$$

$$R_z = 9.5\cos(90^\circ + \theta_2^* + \theta_3^* + \theta_4) + 9.5\sin(\theta_2^* + \theta_3^*) + 10.1\sin(\theta_2^*) + 6.5$$

3.2.2. Diseño e implementación del software

Se explicará la lógica de los algoritmos de configuración y programación de los diferentes componentes de la propuesta, por un lado, se indicará como se configuró la cámara Kinect

en el software LabVIEW para que detecte el posicionamiento de las articulaciones del cuerpo humano, posteriormente se explicará la lógica del controlador implementado en la tarjeta Arduino UNO; la cual, se encargará mediante la cinemática directa generar los movimientos del brazo robótico.

En la figura 51, se observa un diagrama de flujo el cual muestra un sistema diseñado para simular y controlar un brazo robótico utilizando tres modos principales: home, por Kinect y manual. Primero se inicia el proceso y se comprueba si la persona se encuentra dentro del alcance ($<2m$), de ser así se genera una imagen 3D del esqueleto y se obtienen las coordenadas y ángulos necesarios. Luego, en el módulo de control, el usuario puede optar por colocar la mano en la posición inicial, controlar manualmente el movimiento mediante un control deslizante o iniciar el control automático utilizando datos del dispositivo Kinect para imitar el movimiento detectado. Tras finalizar la operación, el sistema detiene la simulación y vuelve al estado inicial.

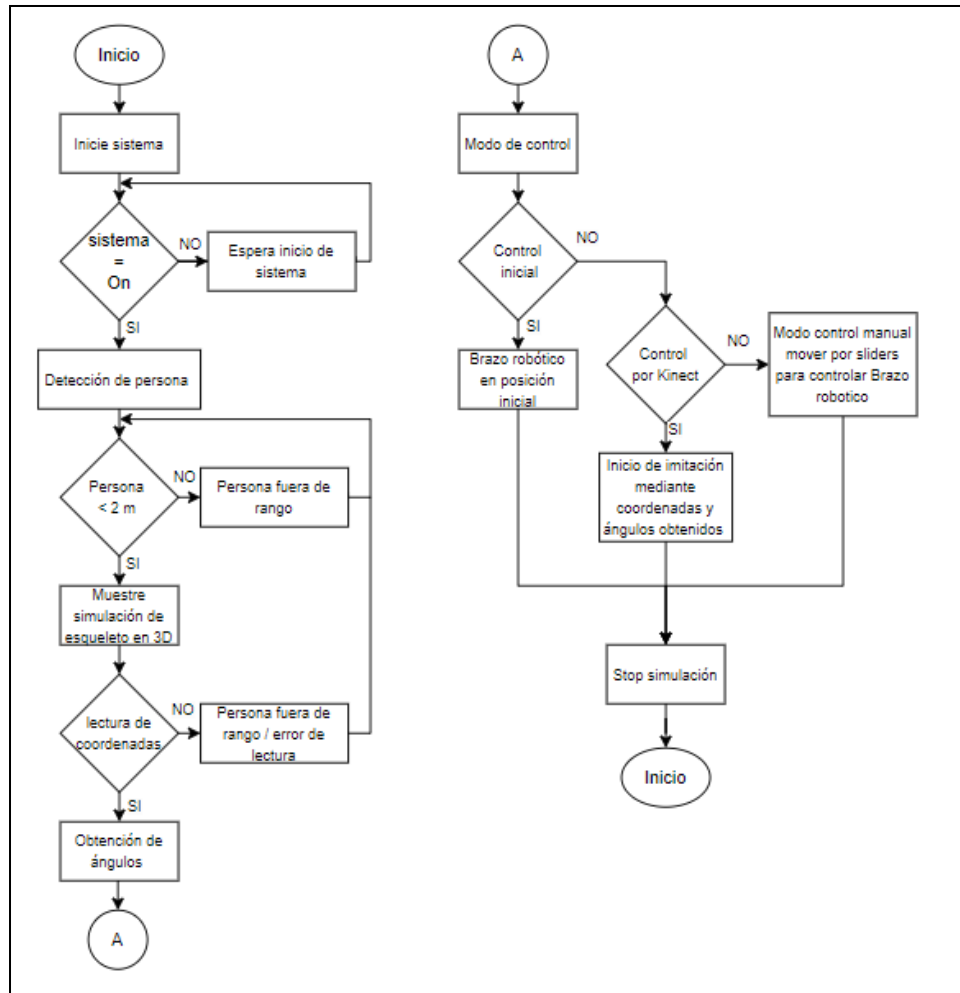


Figura 51. Diagrama de conexión de la cámara Kinect [Fuente: Autor].

3.2.2.1. Procesamiento de imágenes mediante Kinect y LabVIEW

En este apartado, se explicará sobre la comunicación de la cámara Kinect Xbox 360 con el programa LabVIEW y la obtención de coordenadas y ángulos.

En el diagrama de flujo de la figura 52, ilustra un proceso secuencial que comienza con la conexión de una cámara Kinect. Seguidamente, se verifica si el sistema está inicializado para luego detectar la presencia de una persona, si se detecta a una persona en una distancia menor a 2 metros, se proyectará un esqueleto en 2D en la pantalla. De lo contrario, el sistema espera a que la persona se acerque dentro del rango establecido.

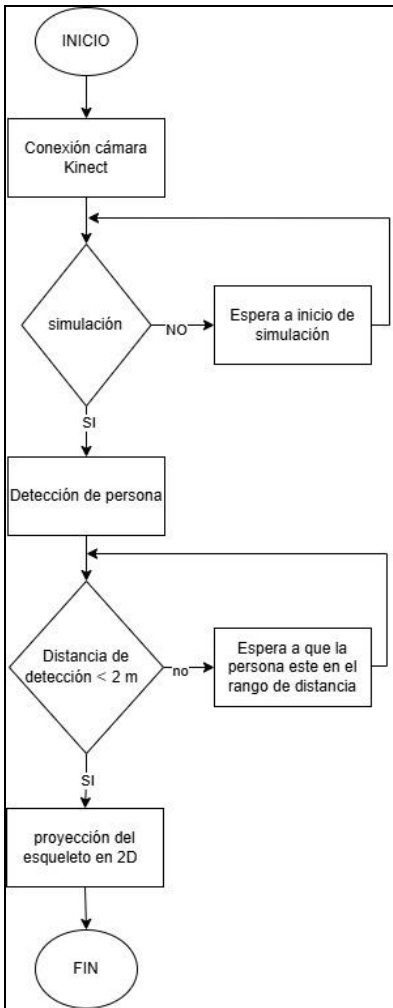


Figura 52. Diagrama de flujo para conexión entre la cámara Kinect y el programa LabVIEW [Fuente: Autor].

Comunicación de cámara Kinect con LabVIEW

La conexión de la cámara Kinect Xbox 360 con el programa LabVIEW se logra mediante el uso de varios VIs (Instrumentos Virtuales), los cuales permiten visualizar en una pantalla el reflejo del usuario, adicional también se proyecta una pantalla adicional con una figura en 3D de un esqueleto humano que simula los movimientos realizados por el usuario.

Dentro de los códigos usados en el programa LabVIEW para la comunicación de la cámara Kinect se tienen los siguientes extractos:

En la figura 53, se muestran los bloques que permiten inicializar el proceso de configuración y manejo de la cámara Kinect para la captura y procesamiento de datos en un sistema tridimensional. El bloque “KINECT INT” inicializa la cámara y establece la conexión de manera automática al puerto COM en el que la ha sido conectada. Conectado a este bloque, se encuentra acoplado el bloque de configuración “KINECT CONFIG”, que permite configurar la resolución de la cámara y los frames por segundo (FPS), dentro de estos parámetros se configuró a 640x480 y a 15 frames por segundo (FPS).

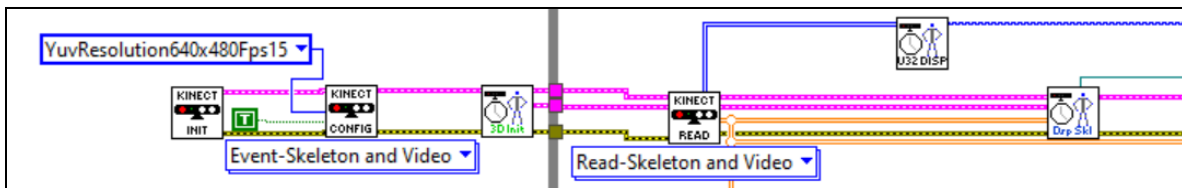


Figura 53. Inicio y configuración de cámara Kinect en LabVIEW [Fuente: Autor].

Seguido a esto, se encuentra conectado el bloque “Read – Skeleton and Video”, el cual se encuentra dentro de un bucle while, el bloque mencionado permite leer tanto el esqueleto de una persona como la imagen de video, capturando datos de profundidad mediante su sensor infrarrojo, permitiendo la representación de las coordenadas en un espacio tridimensional. También se conecta el bloque “Render 3D Skeleton”, el cual representa un sistema de seguimiento de movimiento en tiempo real, el cual permite visualizar en 3D el esqueleto humano, destacando la posición y el movimiento de las articulaciones, permitiendo observar de manera dinámica como se actualizan las posiciones de las articulaciones a medida que el usuario se mueve.

En la figura 54, se observa el bloque “Close –Skeleton and Video” que permite el cierre, tanto de la captura del esqueleto como la captura de video de la cámara Kinect. Siendo esto de forma manual, cuando ya no se necesiten más datos, o como parte de un proceso de reinicio.

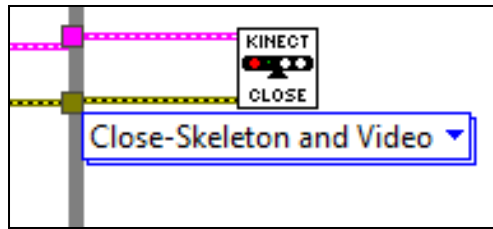


Figura 54. Bloque Close-Skeleton and Video en LabVIEW [Fuente: Autor].

Obtención de las coordenadas y ángulos dados por la Kinect

Después de enlazar LabVIEW con la cámara, el siguiente paso es extraer las coordenadas de los puntos de referencia que se muestran sobre las articulaciones del esqueleto en 3D. Para ejecutar esta acción, la cámara Kinect toma como punto de referencia el punto central del sensor infrarrojo para poder dar los datos, los mismos que vienen dados de la siguiente manera: los movimientos horizontales representan las coordenadas X, los movimientos verticales representan las coordenadas en el eje Y, y los movimientos en profundidad representan las coordenadas en el eje Z.

Una vez que los datos de las coordenadas son accedidos por el código implementado en el software LabVIEW estos pueden ser almacenados y visualizados. Estos datos son almacenados en una matriz que contienen 20 puntos referenciales, los mismos que representan las articulaciones que unen las diferentes partes del cuerpo humano. De este conjunto de puntos se seleccionaron solo aquellos relacionados con los brazos izquierdo y derecho. Estos puntos proporcionan valores en las 3 dimensiones (X, Y, Z).

En la tabla 7, se presenta 20 puntos de referencia correspondientes a las articulaciones del esqueleto humano en un sistema de 2D, utilizados para determinar coordenadas y ángulos específicos, cada punto representa una articulación, como “Hip Center”, “Spine”, “Shoulder Center”; ubicado en el lado izquierdo de la tabla. Así mismo, se destacan las articulaciones claves empleadas para los análisis marcados con colores y asociadas al usuario; ubicado en el lado derecho de la tabla. Por ejemplo, para la articulación en el sistema 2D en los puntos referenciales “Shoulder Left”, “Elbow Left” y “Wrist Left”, se utilizó la articulación del “Codo Izquierdo” (representada de color celeste), la cual corresponde al usuario.

Nº	Punto referencial de articulaciones de esqueleto 2D	Articulaciones del usuario para encuentro de coordenadas y ángulos en esqueleto 2D					
0	Hip Center						
1	Spine						
2	Shoulder Center					Hombro Derecho	Hombro Izquierdo
3	Head						
4	Shoulder Left	Muñeca izquierda	Codo izquierdo				
5	Elbow Left						
6	Wrist Left						
7	Hand Left						
8	Shoulder Right	Muñeca derecha	Codo Derecho	Hombro Derecho			
9	Elbow Right						
10	Wrist Right						
11	Hand Right						
12	Hip Left						
13	Knee Left						
14	Ankle Left						
15	Foot Left						
16	Hip Right						
17	Knee Right						
18	Ankle Right						
19	Foot Right						

Tabla 7. Los 20 puntos referenciales en articulaciones del cuerpo humano.

En la figura 55, muestra dentro de la codificación realizada en LabVIEW, la obtención de coordenada del hombro derecho de una persona detectada por la cámara Kinect. El bloque “shoulder right” proporciona un arreglo con las coordenadas X, Y, Z del hombro derecho. A continuación, se selecciona la primera coordenada (índice 0), que corresponde a la coordenada X. Este dato es enviado al bloque “filter median”, el cual es el encargado de dar el dato mediano de los 50 primeros datos obtenidos, el valor resultante se multiplica por 100 para convertirlos a centímetros. Este proceso se refleja en un indicador que muestra el valor obtenido tres veces por cada articulación

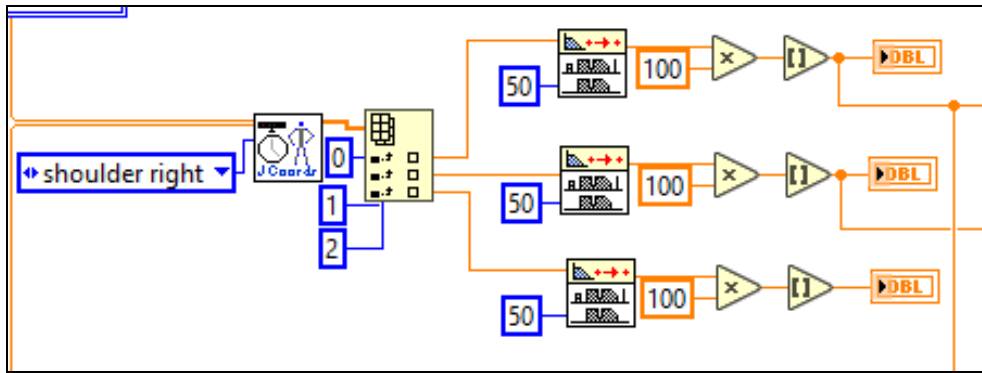


Figura 55. Filtro de datos obtenidos en LabVIEW [Fuente: Autor].

Los datos que pasan por el bloque “filter median” son filtrados para reducir errores, ya que los datos sin procesar tienden a dar imprecisiones. Este filtro permite estabilizar los valores obtenidos, debido a que dichos valores tienden a cambiar con el mínimo movimiento, los cuales pueden ser visualizados mediante un indicador asignado a cada coordenada de cada punto. Después de procesar los datos, estos serán usados para encontrar los ángulos, si el signo del ángulo es negativo su representación será a favor de las manecillas del reloj y si es positivo será en contra de las manecillas del reloj (véase figura 56).

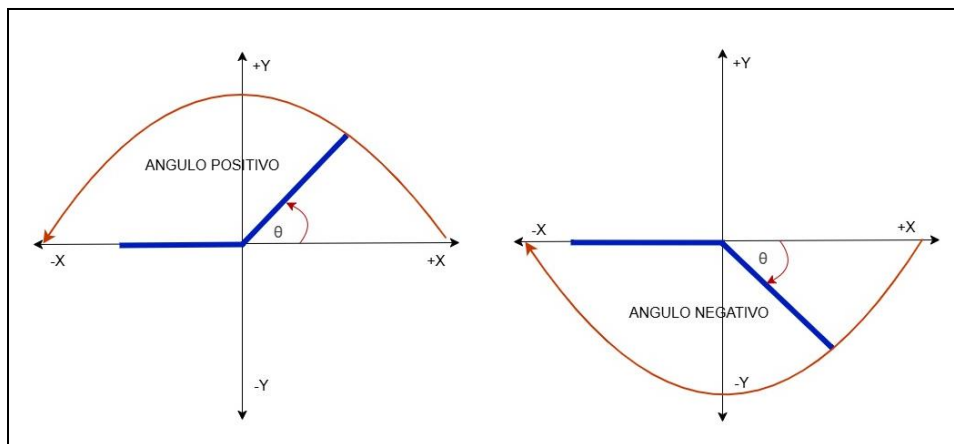


Figura 56. Ángulos positivo - negativo en relación manecillas del reloj [Fuente: Autor].

Como los ángulos que se desean saber son los que conforman los brazos, se requieren escoger tres puntos seguidos, ya que estos tendrán sus coordenadas con las cuáles se formarán vectores que permitirán encontrar dichos ángulos. En la figura 57, se observan las conexiones entre puntos desde la perspectiva de un esqueleto en 2D proyecto por la librería SDK de Kinect.

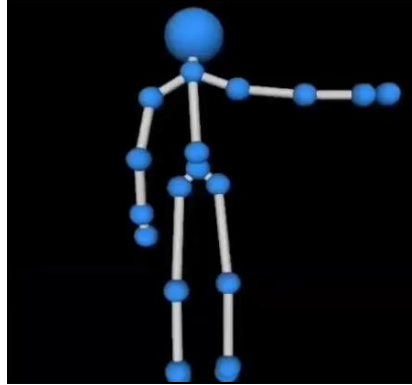


Figura 57. Esqueleto en 2D [Fuente: Autor].

Se usa un punto referencial de coordenadas (véase figura 58) para la toma de ángulos con respecto a la posición de una persona de forma frontal delante de la cámara Kinect, siendo estos: X sus posiciones laterales, Y su posición hacia arriba o abajo y Z la profundidad o distancia entre la persona y la cámara Kinect.

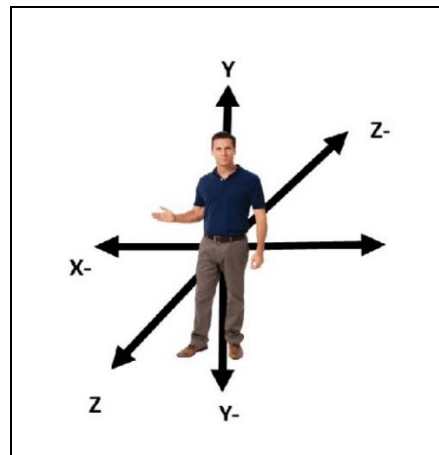


Figura 58. Referencia de coordenadas con respecto persona [Fuente: Autor].

En el proceso de la obtención de coordenadas del sistema de imitación en el programa LabVIEW, se utilizaron diversos bloques, los cuales se mencionarán a continuación, junto con la forma el cómo están entrelazados:

En la figura 59, se observa una serie de operaciones matemáticas aplicadas a un conjunto de datos para calcular los ángulos, los cuales se convierten de radianes a grados. Además, se calcula las distancias o los ángulos relativos entre puntos en un espacio. El resultado final de este proceso es la obtención de la cinemática resultante.

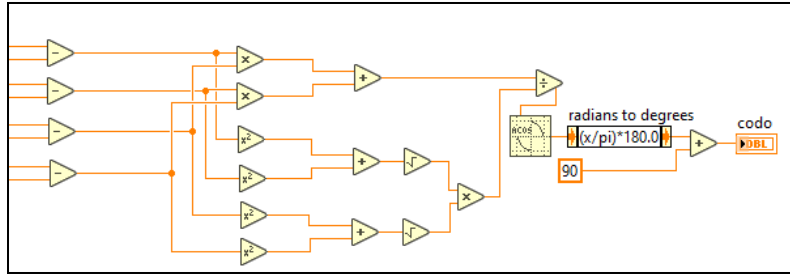


Figura 59. Procedimiento matemático para obtención de coordenadas y ángulos en LabVIEW [Fuente: Autor].

En la figura 60, se observa múltiples entradas rotuladas como “codo”, “hombro”, las cuales reciben datos de posición o ángulos correspondientes a diferentes articulaciones. Este fragmento de código forma parte del sistema que procesa la información cinemática, es decir, relacionada con el movimiento y la posición de las articulaciones.

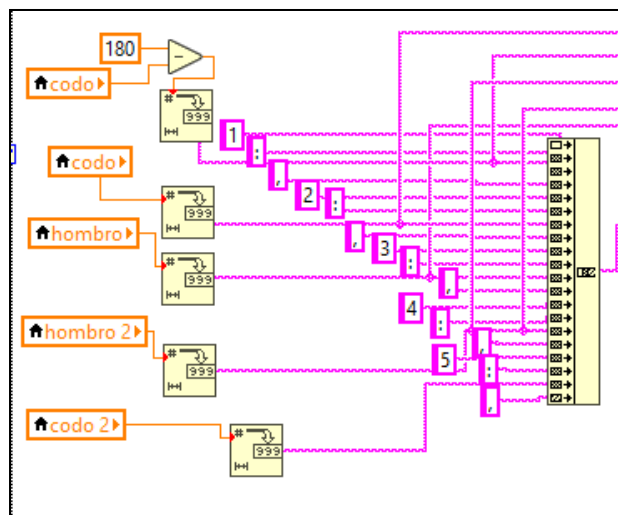


Figura 60. Datos de ángulos en entradas de articulaciones realizados en LabVIEW [Fuente: Autor].

A continuación, en la figura 61, se presenta un diagrama de flujo que toma como punto de referencia la proyección del esqueleto 2D para la lectura de las coordenadas obtenidas del sensor infrarrojo de la cámara. Estos datos llegan en una trama, la cual se separa en puntos, cada uno proporcionando tres 3 coordenadas. Los datos pasan por un filtro que calcula el valor medio de los primeros 50 datos obtenidos. Posteriormente, los datos son mostrados y almacenados, este proceso se realiza 9 veces, y los datos obtenidos son usados para calcular los ángulos que existe entre las articulaciones y mostrarlos.

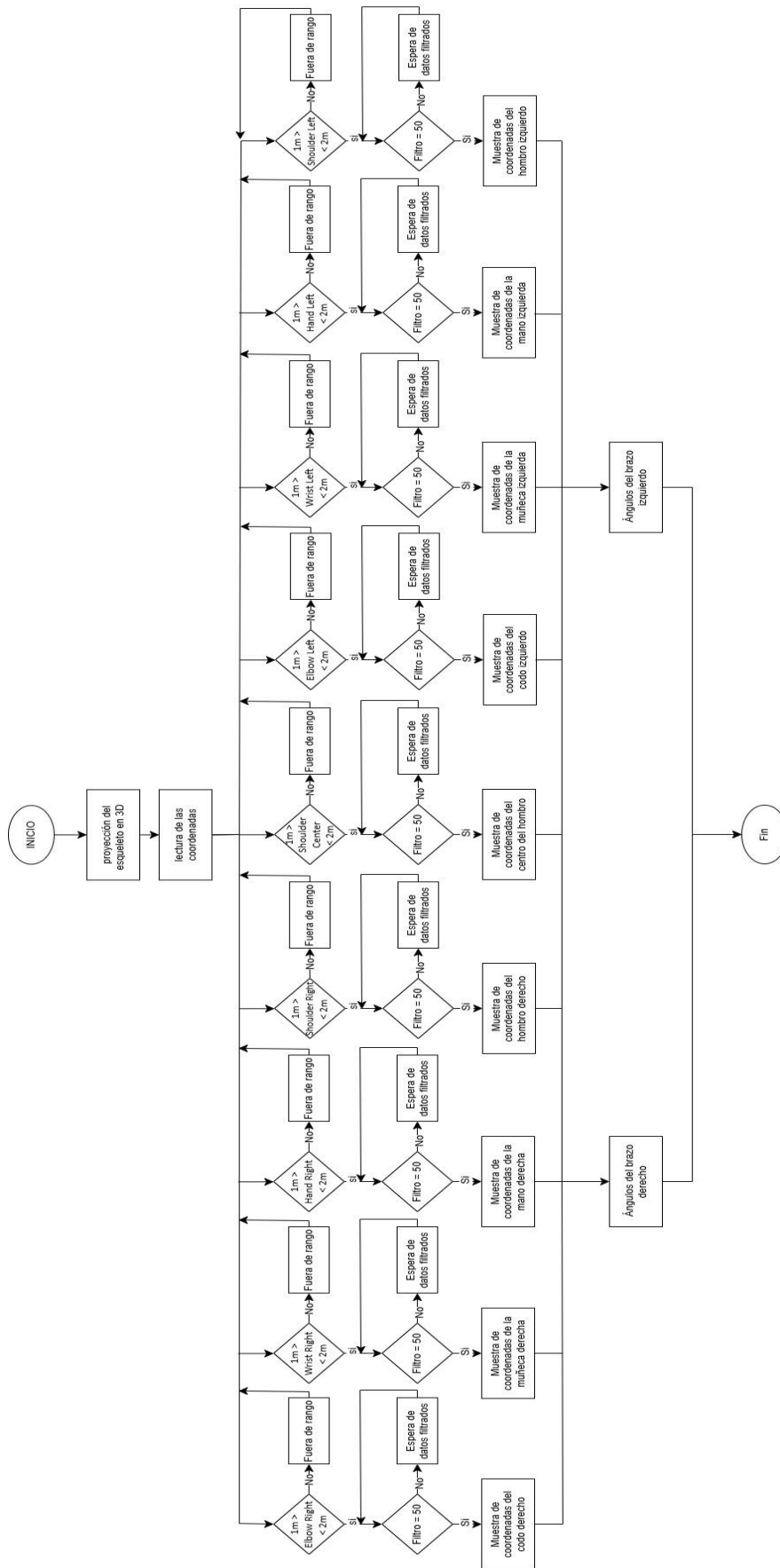


Figura 61. Diagrama de flujo para obtener las coordenadas y ángulo [Fuente: Autor].

3.2.2.2. Configuración del brazo robótico en Arduino

El presente apartado, detalla la configuración del sistema de imitación para el brazo robótico utilizando la plataforma Arduino. El código desarrollado permite gestionar de manera precisa los movimientos del brazo robótico, este sistema emplea un enfoque basado en comunicación serie para recibir comandos y mover los servos a posiciones específicas.

La implementación utiliza la librería “LobotServoController.h”, diseñada por el fabricante del brazo robótico específicamente para el manejo de controladores de servomotores, y la librería “SoftwareSerial.h”, que habilita una comunicación serie secundaria en pines configurables (en este proyecto se usaron los pines 10 y 11).

El programa recibe comandos estructurados desde el puerto serie en un formato predefinido (ID: Ángulo), lo que permite especificar tanto el motor que se desea mover como su ángulo objetivo, incorporando un mecanismo para optimizar los movimientos del brazo, evitando instrucciones redundantes si un servo ya se encuentra en la posición solicitada.

En particular, como se muestra en la figura 62, el sistema utiliza la función `map()` para convertir los valores de ángulos recibidos (en grados) a posiciones requeridas por los servomotores, el cual permite ajustar el movimiento de los componentes clave del brazo robótico, tales como la muñeca, el codo, los hombros en sus diferentes planos (XY, XZ) y el actuador final (gripper). Por ejemplo, si el ID es 1, se asigna un rango de valores posición adecuado para mover la muñeca del brazo; si el ID es 2, se realiza el mapeo correspondiente para mover el codo. Posteriormente, la función `moveServo()` es llamada para enviar la posición calculada al motor, junto con un parámetro que define la velocidad del movimiento (en este caso, 250ms), esto permitirá una gestión modular y precisa de los diferentes grados de libertad del brazo robótico.


```

if (ID == 1)
{
  servoPosition = map(Angulo, 0, 180, 185, 925);
  myse.moveServo(3, servoPosition, 250); // Mover el servo 1 MUÑECA
}
else if (ID == 2)
{
  servoPosition = map(Angulo, 90, 180, 590, 925);
  myse.moveServo(4, servoPosition, 250); // Mover el servo 2 CODD
}
else if (ID == 3)
{

```

Figura 62. *Uso de la función map () en el código en Arduino IDE [Fuente: Autor].*

3.2.2.3. Configuración del brazo robótico en LabVIEW

El control del brazo robótico X-Arm 1S mediante LabVIEW se fundamenta en la integración de hardware y software para capturar datos de movimiento humano, procesarlos y traducirlos en comandos que permitan la manipulación precisa de las articulaciones robóticas.

A continuación, se describe en detalle el contenido y la función de cada extracto programado en este sistema de imitación desarrollado en el programa LabVIEW:

En la figura 63, se observa el uso del recurso VISA para establecer la comunicación entre el sistema y el sensor Kinect, se configuran también parámetros esenciales como la resolución del video (640x840 píxeles) y la velocidad de transmisión (9600 baudios), dicha configuración incluye la inicialización del sensor Kinect, la captura de datos del esqueleto y video, y la preparación del sistema para procesar la información en tiempo real. Esta etapa es fundamental para asegurar que los datos obtenidos sean precisos y útiles para los algoritmos posteriores de control del brazo robótico.

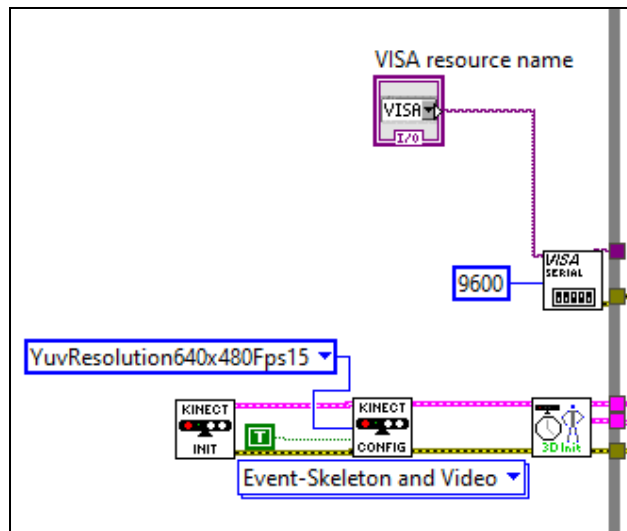


Figura 63. Recurso VISA y Kinect [Fuente: Autor].

En la figura 64, presenta los diferentes modos de control implementados para el brazo robótico. Cada articulación del robot: muñeca – codo - hombro en plano XY (arriba y abajo) - hombro en plano XZ (adelante y atrás) - gripper o garra, está configurada para recibir entradas específicas a través de bloques de control. Estas entradas son traducidas en ángulos que determinan los movimientos del robot. Este esquema permite tanto el control manual mediante sliders como el control automático basado en coordenadas generadas por el sensor Kinect mediante el movimiento de imitación creado por la persona.

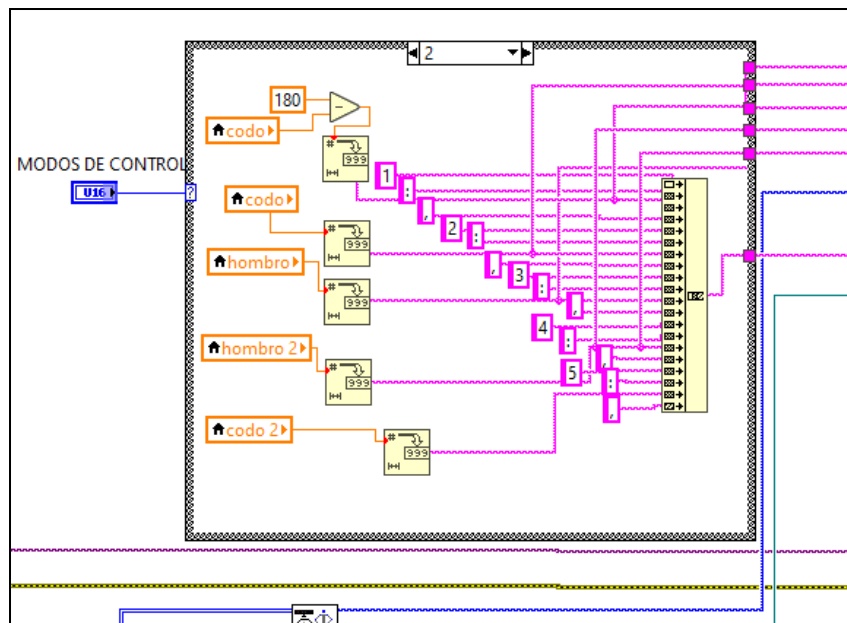


Figura 64. Modos de control configurado en LabVIEW [Fuente: Autor].

En la figura 65, se detalla la configuración para el control de los movimientos en las articulaciones individuales del brazo robótico. Se observa cómo cada articulación está asociada a un bloque específico que traduce los datos obtenidos en movimientos controlados.

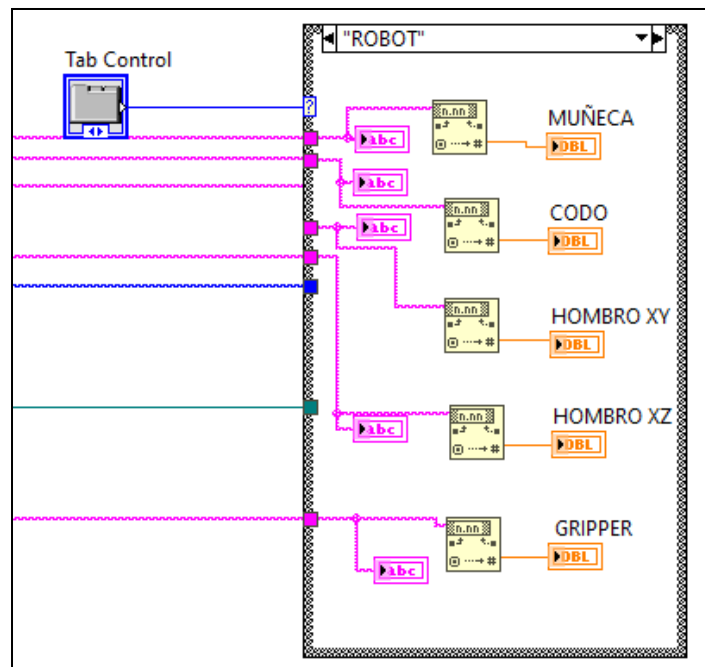


Figura 65. Control de movimientos en muñeca, codo, hombros y gripper en brazo robótico [Fuente: Autor].

En la figura 66, se muestra el diseño lógico implementado para interpretar las coordenadas obtenidas por el sensor Kinect. Estas coordenadas son convertidas en ángulos específicos para las articulaciones del brazo derecho. Cada articulación cuenta con su propia estructura de procesamiento, la cual se encarga de recibir, interpretar y transmitir los datos necesarios para mover las articulaciones en X, Y, Z (0,1,2) de manera coordinada, mediante captura de 30 fotogramas y tomando la media de estos usando una fórmula.

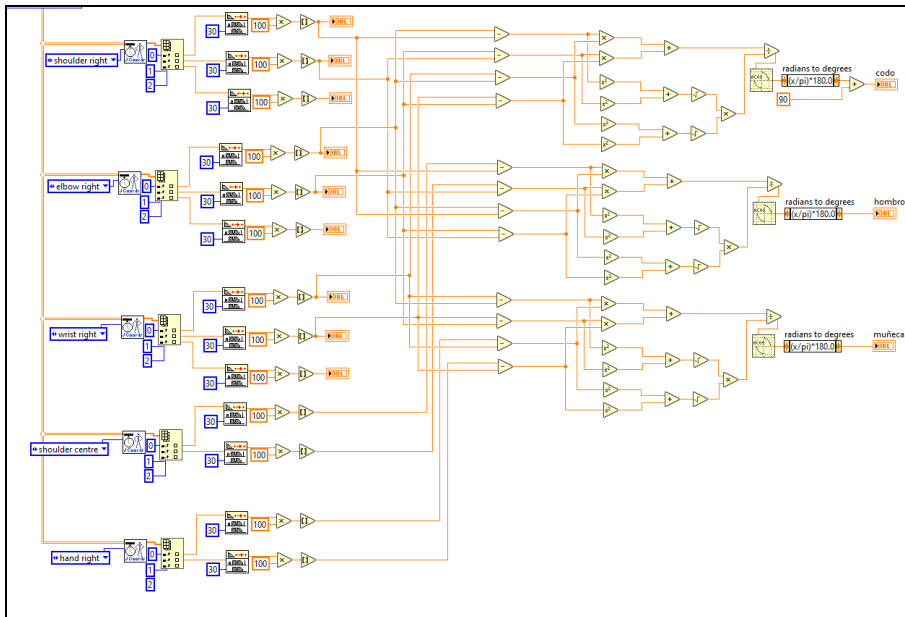


Figura 66. Lectura de ángulos del brazo derecho [Fuente: Autor].

En la figura 67, ilustra un diseño similar al de la figura 68, pero aplicado al brazo izquierdo. Este esquema procesa las coordenadas obtenidas por el Kinect y las traduce en ángulos específicos para las articulaciones correspondientes. La lógica aplicada es modular y paralela, lo que permite el control simultáneo de ambos brazos de la persona traducido en diferentes tipos de movimientos realizado en el brazo robótico siendo estos: muñeca – codo - hombro en plano XY (arriba y abajo) - hombro en plano XZ (adelante y atrás) - gripper o garra (abrir o cerrar).

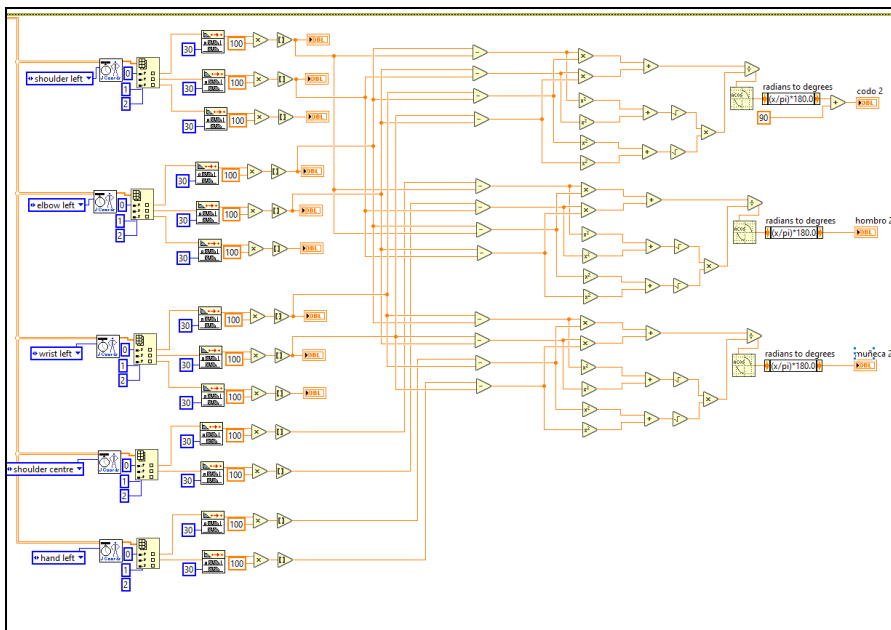


Figura 67. Lectura de ángulos del brazo izquierdo [Fuente: Autor].

3.2.3. Diseño final del Sistema:

Para el diseño final implementado del proceso (véase figura 68) para el movimiento por imitación del brazo robótico se usó de tipo comercial el brazo robótico X-Arm 1S, el cual tiene 5 grados de libertad y un gripper, con una programación base del robot mismo para el movimiento de sus servos inteligentes integrados, estos son accionados mediante comunicación UART mediante un controlador Arduino UNO y los programas Arduino IDE y LabVIEW, esta programación realizada; en un ordenador, para la captura de los movimientos y la adquisición de los ángulos mediante una cámara Kinect 360°.



Figura 68. Diagrama del proceso [Fuente: Autor].

Dentro de la interfaz HMI ya finalizado del sistema de imitación desarrollado en LabVIEW, se dividirá en 3 ventanas, de las cuales cumplirán las siguientes funciones:

- Ventana 1 (véase figura 69), o también llamada “Cámara”, cumple la función de mostrar dos pantallas al usuario, la primera permite la captación de imagen de la cámara Kinect, donde se refleja la persona en cuestión y la segunda pantalla muestra un esqueleto humano en 2D, el cual imita el movimiento de las articulaciones de la persona. También cuenta con dos botones en la parte inferior izquierda, los cuales permiten la selección del puerto COM del Arduino UNO conectado al ordenador y un botón “stop” que permite la detención del sistema.



Figura 69. Ventana principal de interfaz del sistema de imitación en LabVIEW [Fuente: Autor].

- Ventana 2 (véase figura 70), o también llamada “coordenadas y ángulos”, esta ventana cumple la función de mostrar los ángulos de los brazos izquierdo y derecho (incluyendo hombros, codos y muñecas), en las diferentes coordenadas X, Y, Z, valores que son procesados mediante el movimiento realizado por la persona captada por la cámara Kinect.

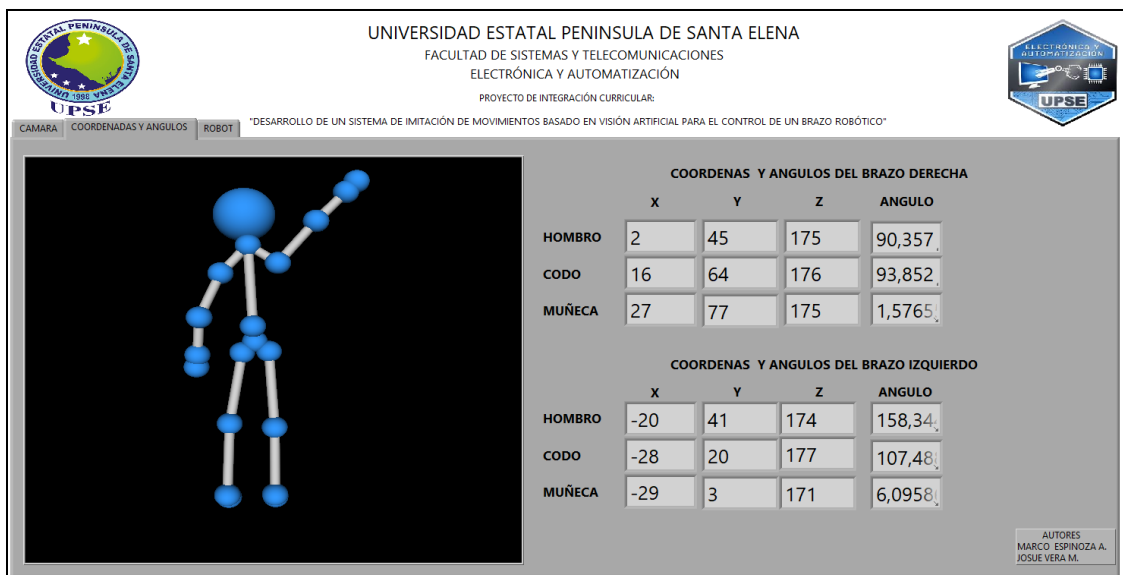


Figura 70. Ventana de visualización de coordenadas en interfaz del sistema de imitación en LabVIEW [Fuente: Autor].

- Ventana 3 (véase figura 71), o también llamada “robot”, es una ventana que permite al usuario cambiar el tipo de modo de control y su vez el control manual

del brazo robótico mediante sliders, mostrando también sus ángulos realizados mediante números y gráficamente.

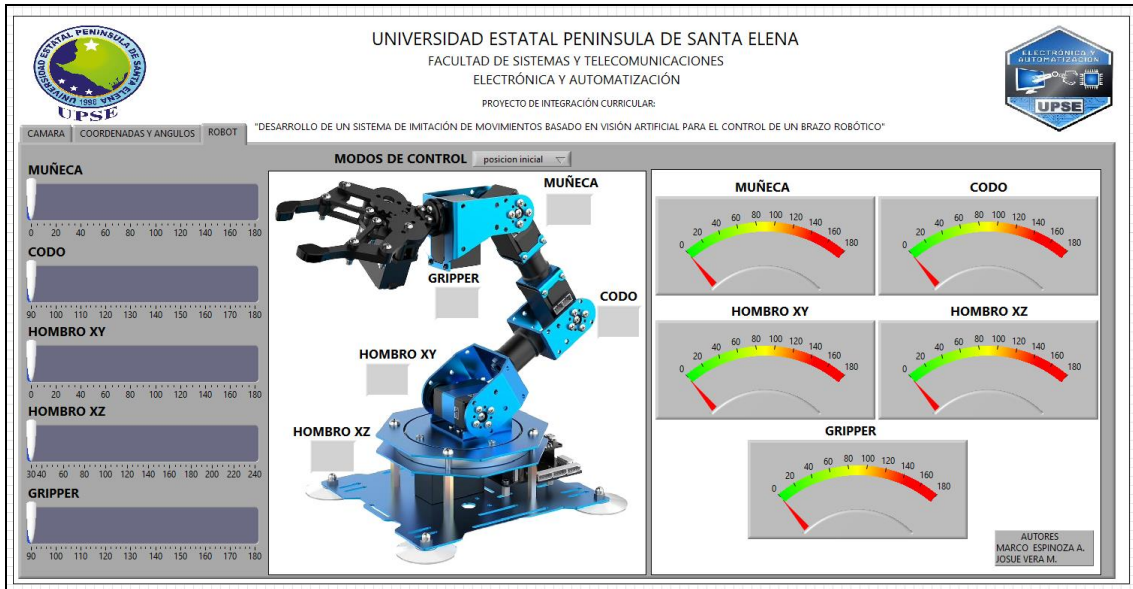


Figura 71. Ventana de monitoreo de interfaz del sistema de imitación en LabVIEW [Fuente: Autor].

3.3, PRUEBAS Y RESULTADOS

En este capítulo, se abordan los resultados obtenidos en las pruebas realizadas donde se muestra el cumplimiento de los puntos que fueron planteados en la propuesta a realizar. Se realizaron pruebas como lo es la calibración del sensor de la cámara, los ángulos obtenidos en LabVIEW, precisión de los servomotores, con el fin de mostrar las varias pruebas que se tuvo que llegar a realizar para conseguir que el sistema de imitación se logre con éxito, tanto en parte de hardware como en el software, las limitaciones que se llegaron a tener, así como las perturbaciones o contratiempos.

3.3.1. Pruebas en calibración de cámara Kinect (sensor)

Se divide en dos puntos importantes, la luminosidad del entorno en que se trabaja y la distancia entre el sensor y la persona, siendo estos los dos puntos importantes para que la cámara Kinect pueda captar y reflejar los puntos del cuerpo con precisión donde la correcta luminosidad evita que se creen sombras o reflejos que interfieran en la captura

de los puntos, mientras que la distancia permite que los puntos captados no se distorsionen mientras se realizan movimientos y se registran a la vez.

- **Luminosidad**

Para poder identificar cual es la luminosidad exacta con la que la cámara puede trabajar se realizó, mediante pruebas, ya que la cámara trabaja con un sensor infrarrojo donde se determinó que es importante que el lugar de trabajo cuente con una iluminación moderada. Si se trabaja con poca luz (véase figura 72), no se tiene inconveniente para detectar el cuerpo humano, pero si muestra desventajas en la resolución de video. Por otro lado, si se trabaja con demasiada luz (véase figura 73), el sensor infrarrojo puede generar datos incorrectos el cual afecta la precisión de las mediciones.



Figura 72. Prueba luminosidad - Baja.



Figura 73. Prueba luminosidad - Alta.

- **Distancia**

La distancia depende de la estatura de la persona, mediante pruebas se determinó que la altura promedio de la persona debe estar entre 1.40 a 1.50 m de esta manera se muestra la siguiente tabla:

Distancia (cm)	Visualización
0-84	No se visualiza la persona
85-149	Se muestran distorsiones
150-200	Rango considerado de visualización
201-270	Se muestran distorsiones

Tabla 8. Pruebas de distancia y visualización con la cámara Kinect.

En la tabla 8, se puede concluir que el rango que debe estar ubicado el usuario es de 150 a 200 cm (véase figura 74), dado la captación de imagen que presenta el sensor de la cámara Kinect, se interpreta de la siguiente manera: si la persona está cerca lo interpreta como si estuviera lejos (véase figura 75) y si está lejos la interpretación es de mayor tamaño (véase figura 76).

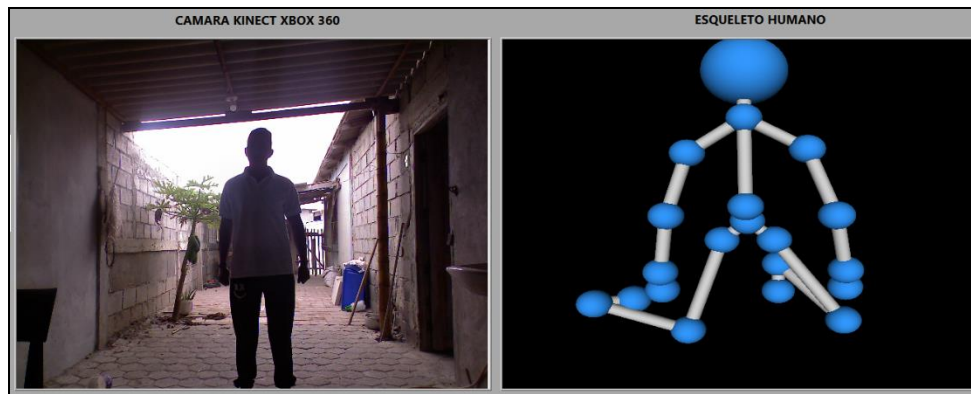


Figura 74. Usuario a distancia menor de 2 metros.

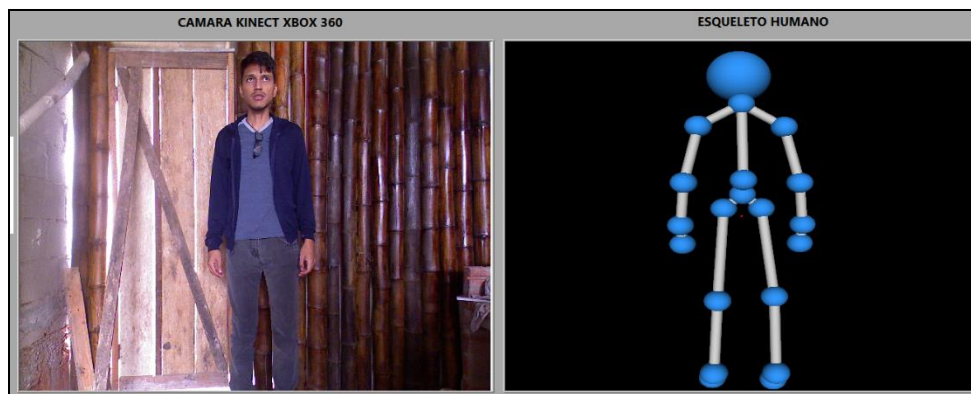


Figura 75. Usuario a distancia mayor igual a 2 metros.



Figura 76. Usuario a distancia mayor 3 metros.

3.3.2. Pruebas de grados formados por las articulaciones de los brazos

Mediante las pruebas realizadas se obtuvieron las coordenadas que conforman los brazos derecho e izquierdo, con las cuales, se pueden encontrar a la vez, los ángulos de los hombros, codos y muñecas.

A continuación, se comparan los ángulos que forman los brazos del usuario con los ángulos que muestra la figura en 3D, de esta manera se consideró trabajar con ángulos que van de 0 a 180° en cada articulación. Se tomaron en cuenta los siguientes ángulos: 90°, 125°, 150° y 180 ya que el cuerpo humano no permite realizar ciertas acciones, por lo tanto, la proyección en 3D dará datos erróneos.

Hombros

En la tabla 9, se pueden observar los resultados obtenidos al colocar las articulaciones que conectan los hombros con el codo en una posición de 90°, donde se muestran que los ángulos formados no son exactos, el brazo derecho en una posición de 90.357° (véase figura 77), y el brazo izquierdo en una posición de 90.447° (véase figura 78).

Pruebas	Derecho		Izquierdo	
	90°			
	Obtenido	Error	Obtenido	Error
1	90.357	0.357	91.019	1.019
2	91.909	1.909	90.447	0.447
3	90.552	0.552	90.689	0.689
4	90.657	0.657	90.447	0.447

Tabla 9. Pruebas realizadas en los hombros a 90°.

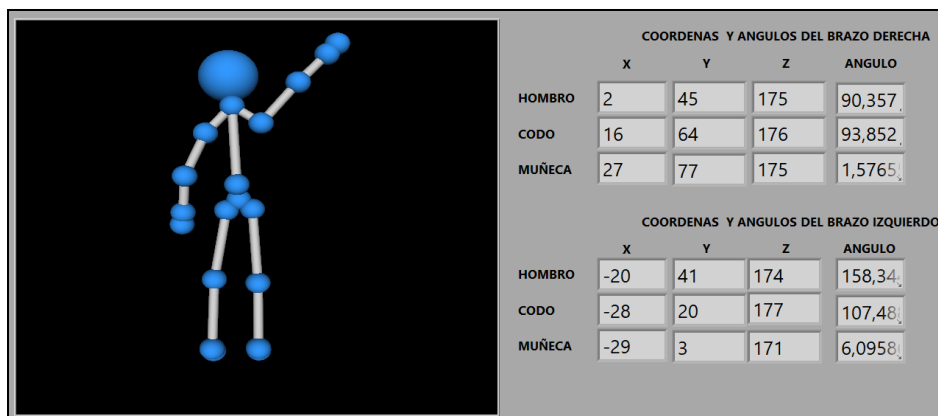


Figura 77. Pruebas con hombro derecho a 90° [Fuente: Autor].

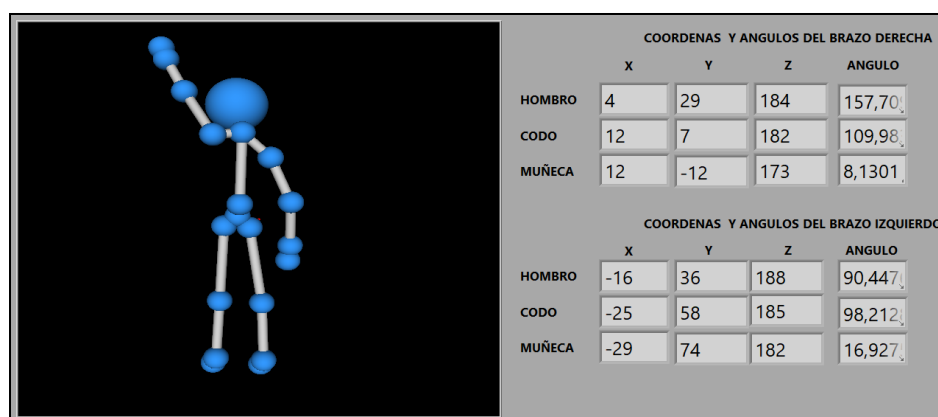


Figura 78. Pruebas con hombro izquierdo a 90° [Fuente: Autor].

En la tabla 10, se observan los resultados al colocar la articulación en una posición de 125°, se puede divisar que todas las pruebas tienen un error mínimo, el brazo derecho a 125.280° (véase figura 79) y el brazo izquierdo a 125.270° (véase figura 80).

Pruebas	Derecho		Izquierdo	
	125°			
	Obtenido	Error	Obtenido	Error
1	125.280	0.280	124.400	-0.600
2	125.300	0.300	125.270	0.270
3	125.301	0.301	125.830	0.830
4	125.290	0.290	125.550	0.550

Tabla 10. Pruebas realizadas en los hombros a 125°.

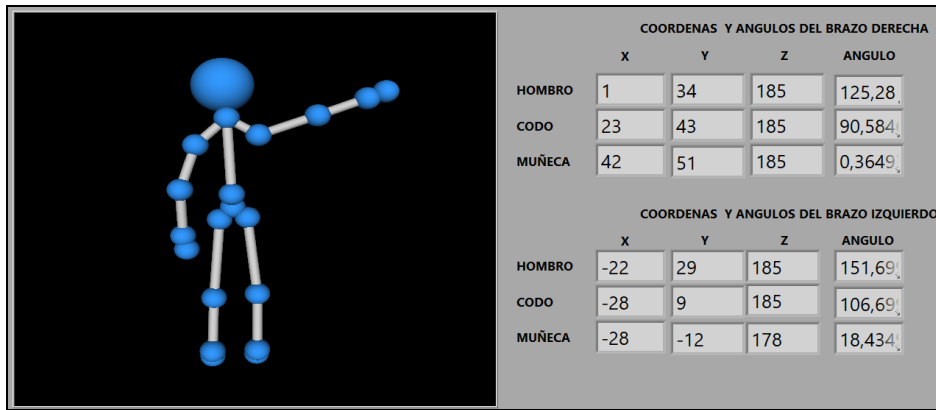


Figura 79. Pruebas con hombro derecho a 125° [Fuente: Autor].

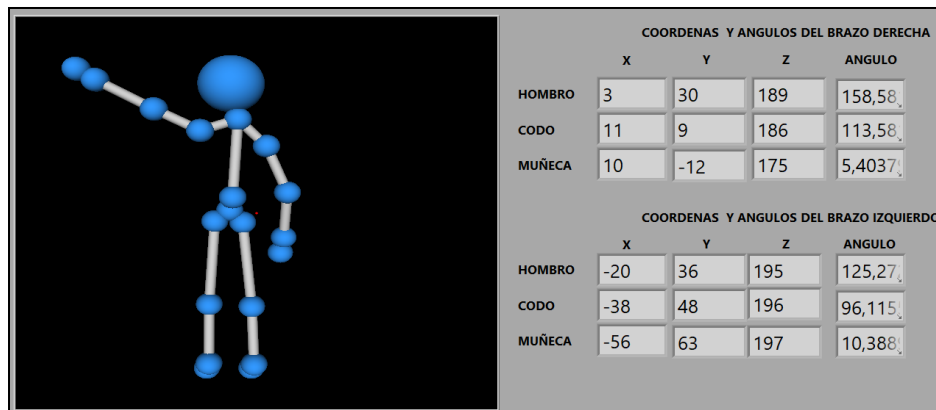


Figura 80. Pruebas con hombro izquierdo a 125° [Fuente: Autor].

En la tabla 11, se observa que los resultados tienen un error de casi 1° para ambos brazos si se toman los resultados a partir de los 150°, caso contrario se aproximará al grado deseado si los resultados se encuentran entre -0.5 y 0.4 grados del ángulo evaluado, en la figura 81 se observa como la articulación (hombro) se encuentra a 149.740°; por lo tanto, se puede decir que este valor se aproxima a 150° y en la figura 82 la articulación se encuentra en una posición de 150.990°, por lo tanto no se puede aproximar

Pruebas	Derecho		Izquierdo	
	150°			
	Obtenido	Error	Obtenido	Error
1	150.830	0.830	150.990	0.990
2	149.740	-0.260	149.210	-0.790
3	150.840	0.840	150.991	0.991
4	150.835	0.835	150.990	0.990

Tabla 11. Pruebas realizadas en los hombros a 150°.

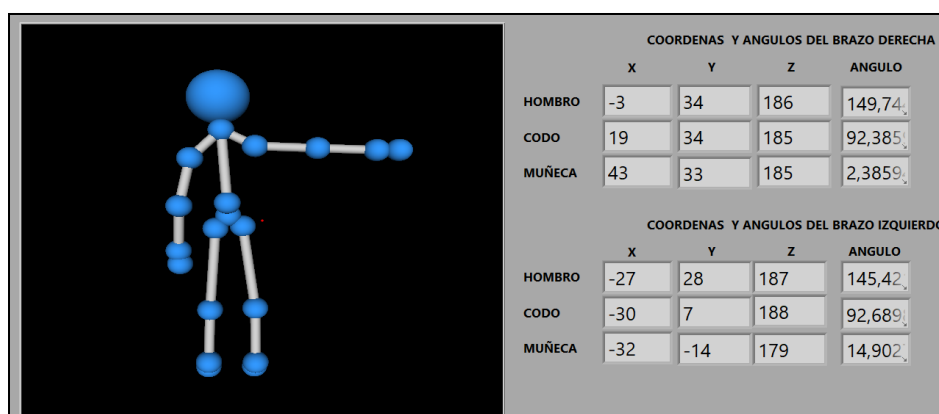


Figura 81. Pruebas con hombro derecho a 150° [Fuente: Autor].

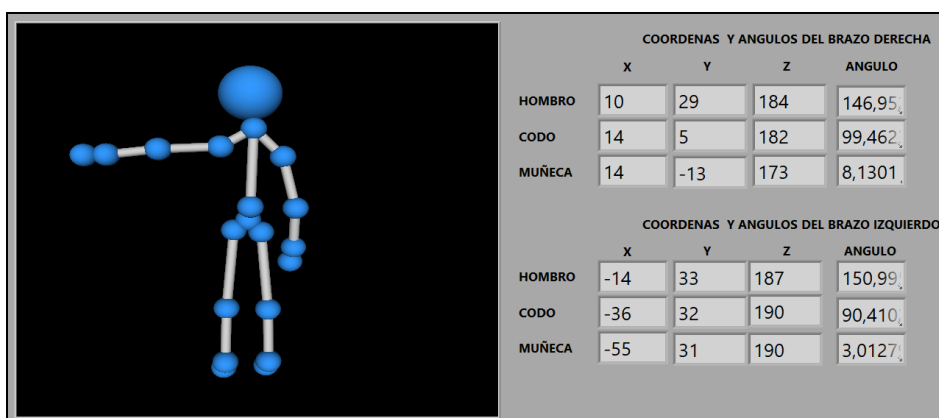


Figura 82. Pruebas con hombro izquierdo a 150° [Fuente: Autor].

En la tabla 12, se observa que los resultados solo se aproximan al ángulo deseado para ambos brazos de los cuales se puede ver que los ángulos no pasan de 180°, los hombros aproximan al valor deseado (véase las figuras 83 y 84).

Pruebas	Derecho		Izquierdo	
	180°			
	Obtenido	Error	Obtenido	Error
1	179.330	-0.670	179.560	-0.440
2	179.230	-0.770	179.150	-0.850
3	179.480	-0.520	179.540	-0.460
4	179.480	-0.520	179.460	-0.540

Tabla 12. Pruebas realizadas en los hombros a 180°.

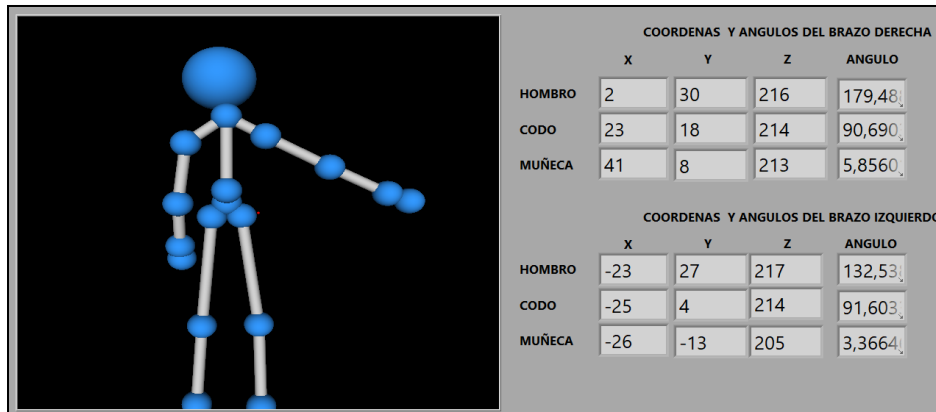


Figura 83. Pruebas con hombro derecho a 180° [Fuente: Autor].

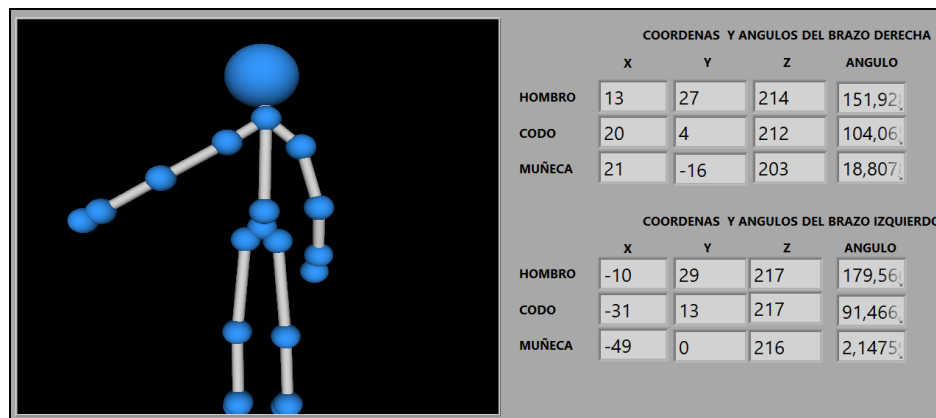


Figura 84. Pruebas con hombro izquierdo a 180° [Fuente: Autor].

○ **Resultados**

En las pruebas realizadas en los hombros se puede observar que la articulación hombro central-hombro - derecho/izquierdo, se encuentra de manera inclinada, la articulación hombro derecho/izquierdo – codo derecho/ izquierdo son los únicos que se encuentran en movimiento, se logró formar ángulos que van 60 a 180 grados aproximando, si se intenta formar ángulos superiores al ya mencionado, éste los tomará como si fueran ángulos que

se encuentren en el rango dado que la figura 3D, estas articulaciones tienen restricciones en esos 3 puntos tomados para calcular los grados para los hombros.

Codos

En la tabla 13, se encuentran los resultados obtenidos por las articulaciones que fueron usadas para encontrar los ángulos en los codos, para esta prueba se consideró evaluar las posiciones de las articulaciones a 90° en los codos; entre las cuales, solo se obtuvieron errores de 0.666 y 0.909%, para ambos brazos el 90% de las pruebas realizadas fueron ángulos exactos, 6% fueron errores que se aproximan y el 4% fueron errores de ángulos que se alejan, en las figuras 85 y 86 se muestran cuál es la posición a 90°, las cuales, son representadas si los brazos se encuentran extendidos.

Pruebas	Derecho		Izquierdo	
	90°			
	Obtenido	Error	Obtenido	Error
1	90.666	0.666	90.871	0.871
2	90	0	90.909	0.909
3	90	0	90	0
4	90.670	0.670	90	0

Tabla 13. Pruebas realizadas en los codos a 90°.

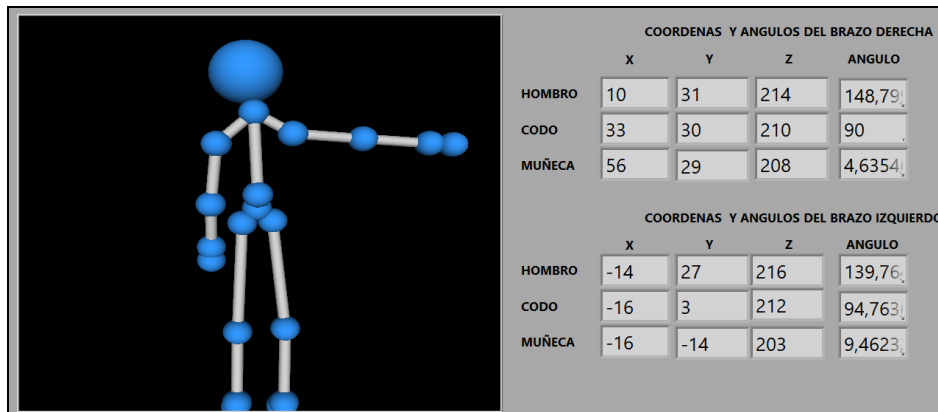


Figura 85. Pruebas con codo derecho a 90° [Fuente: Autor].

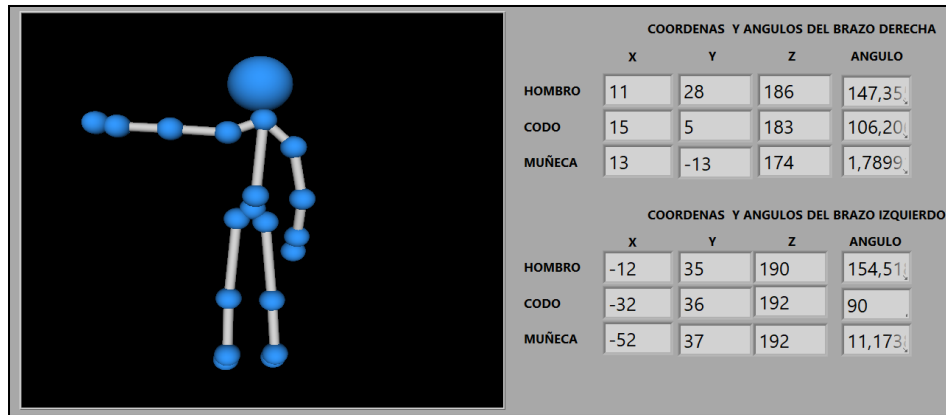


Figura 86. Pruebas con codo izquierdo a 90° [Fuente: Autor].

En la tabla 14, los resultados no fueron favorables para la posición de 125° entre las articulaciones que conforman el codo, en las cuales se alejan mucho del ángulo deseado teniendo un error aproximado de más menos 1°, las articulaciones del brazo derecho forman un ángulo de 124.150 (véase figura 87) el cual está lejos del ángulo deseado, mientras que las articulaciones del brazo izquierdo (véase figura 88) forman un ángulo de 126.220°, el cual se aleja del ángulo que corresponde formarse

Pruebas	Derecho		Izquierdo	
	125°			
	Obtenido	Error	Obtenido	Error
1	124.150	-0.850	126.790	1.790
2	126.950	1.950	126.220	1.220
3	124.001	-0.999	126.770	1.770
4	126.960	1.960	126.240	1.240

Tabla 14. Pruebas realizadas en los codos a 125°.

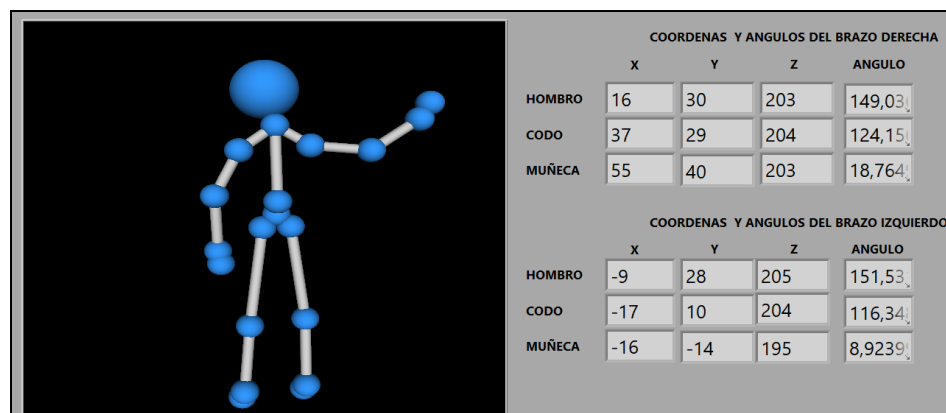


Figura 87. Pruebas con codo derecho a 125° [Fuente: Autor].

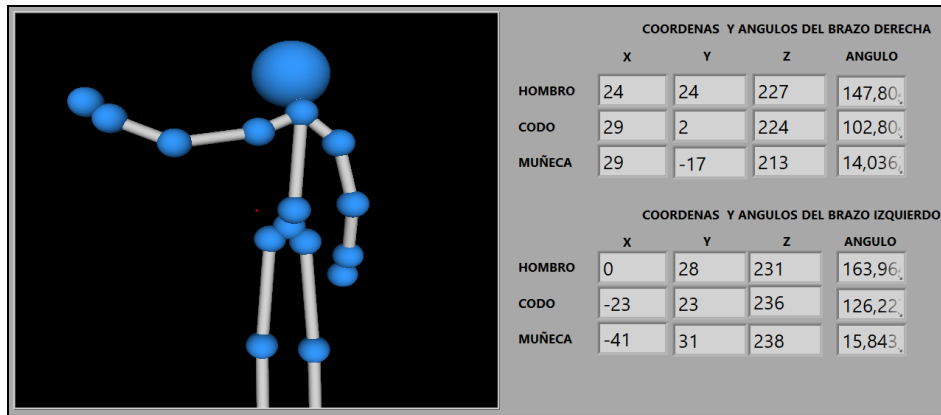


Figura 88. Pruebas con codo izquierdo a 125° [Fuente: Autor].

En la tabla 15, se observa los ángulos obtenidos por el brazo derechos presentan un error mínimo, lo que significa que el 99% de las pruebas realizadas fueron exitosas, caso contrario con el brazo opuesto, las pruebas realizadas en el codo dieron un error que supera un grado del valor deseado, las articulaciones solo muestran un error de 0.020% del valor deseado (véase figura 90) mientras que las articulaciones se alejan del ángulo a evaluar (véase figura 91).

Pruebas	Derecho		Izquierdo	
	150°			
	Obtenido	Error	Obtenido	Error
1	150.020	0.020	151.420	1.420
2	150.020	0.020	151.660	1.660
3	150.020	0.020	151.430	1.430
4	150.020	0.020	151.552	1.552

Tabla 15. Pruebas realizadas en los codos a 150°.

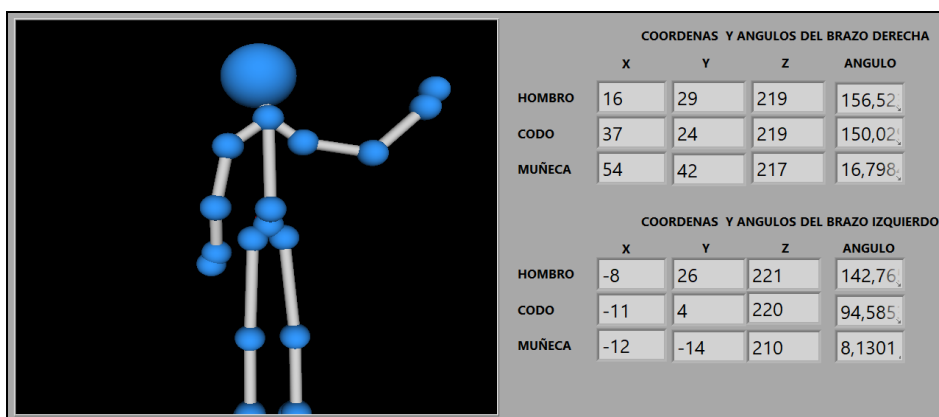


Figura 89. Pruebas con codo derecho a 150° [Fuente: Autor].

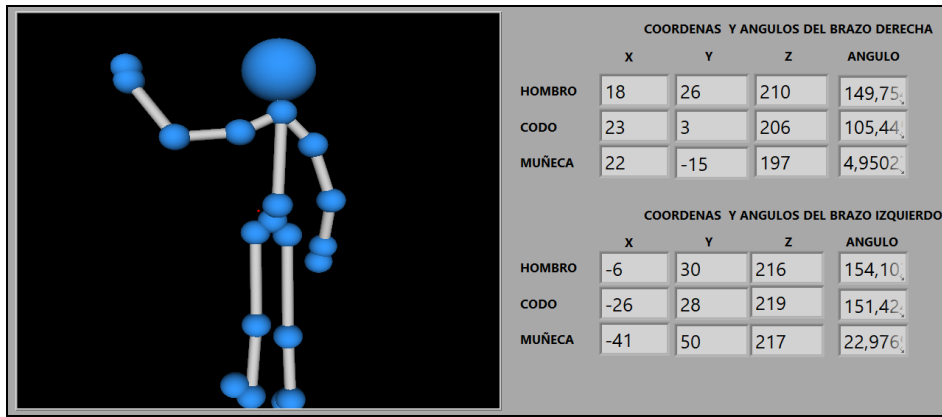


Figura 90. Pruebas con codo izquierdo a 150° [Fuente: Autor].

En la tabla 16, se observa a las articulaciones, estas poseen mayor precisión al formar ángulos perpendiculares. Si el error máximo que se obtuvo fue de 0.450%, la extremidad izquierda y de 0.230% en la extremidad derecha, siendo la extremidad que llego a formar el ángulo deseado con exactitud mientras que en el otro brazo solo se pudo aproximar sin tener pruebas de exactitud en la figura 91 se puede demostrar que el codo derecho tiene mayor precisión que el brazo izquierdo como se muestra en la figura 92.

Pruebas	Derecho		Izquierdo	
	180°			
	Obtenido	Error	Obtenido	Error
1	180.230	0.230	180.450	0.450
2	180	0	180.150	0.150
3	180	0	180.160	0.160
4	180	0	189.340	0.340

Tabla 16. Pruebas realizadas en los codos a 180°.

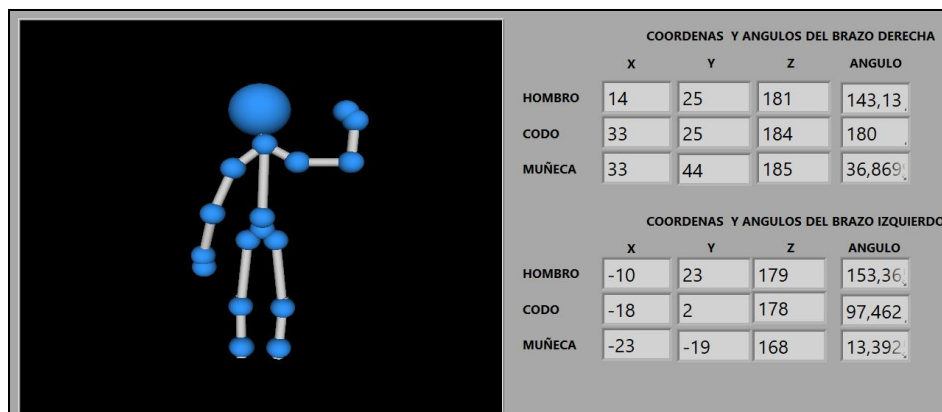


Figura 91. Pruebas con codo derecho a 180° [Fuente: Autor].

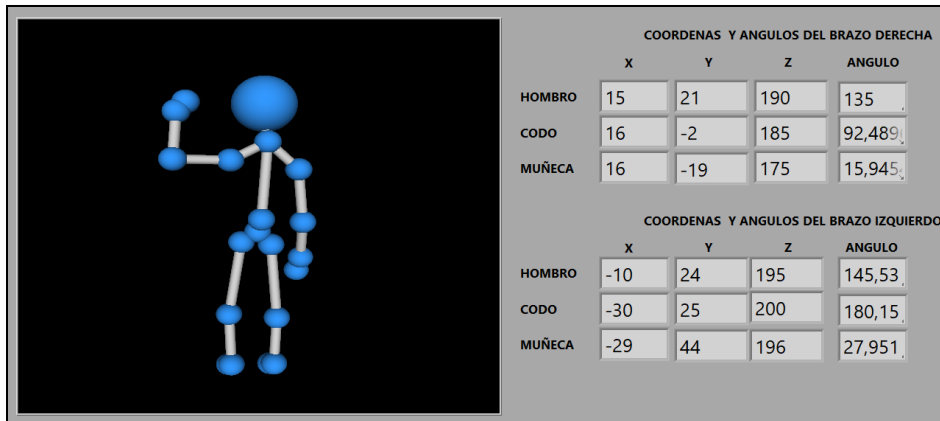


Figura 92. Pruebas con codo izquierdo a 180° [Fuente: Autor].

○ **Resultados**

Durante las pruebas que se realizaron en los codos se pudo demostrar que los ángulos son exactos mientras las articulaciones se encuentren de manera perpendicular o formen rectas entre sí, caso contrario si se desea cambiar de posición solo se lograra aproximar a los ángulos deseado. Del punto de vista en que fueron tomadas las posiciones solo se pueden llegar tener ángulos vallan de 90 a 180 dado que la articulación del brazo humano posee bloque óseo y ligamentoso que no permiten que el codo se mueva hacia atrás.

Muñeca

En la tabla 17, se muestran los resultados obtenidos por las articulaciones que conforma la muñeca, en la cual no se obtuvo resultados favorables, dado que en la figura 3D los puntos que unen a las muñecas con las manos se encuentran casi unidas cualquier ángulo que se quiera formar no tendrá exactitud, debido a que las articulaciones presenta demasiado ruido hasta para formar ángulos que no pasen de 20°, se encuentran unido los puntos que unen a la muñeca con la mano (véase figura 93), por ende a esta obstrucción que se tiene no es posible seguir con las pruebas ya que se tendrán los mismo errores.

Pruebas	Derecho		Izquierdo	
	90°			
	Obtenido	Error	Obtenido	Error
1	18.434	-71.566	6.340	-83.660
2	9.395	-80.605	2.752	-87.248
3	15.962	-74.038	0.784	-89.216
4	15.982	-74.018	7.745	-82.255

Tabla 17. Pruebas realizadas en la muñeca a 90°.

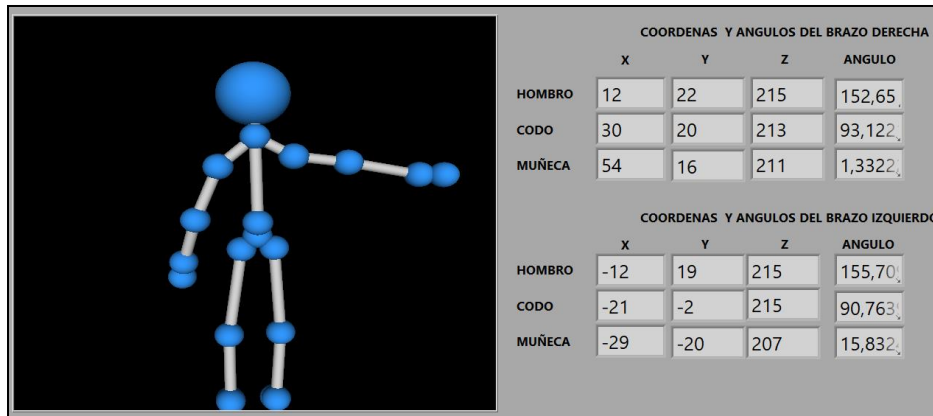


Figura 93. Pruebas con muñeca a 90° [Fuente: Autor].

○ **Resultados**

En las pruebas que se realizaron se llegó a la conclusión que las muñecas de los brazos no pueden ser consideradas para trabajar con el brazo robótico debido que sus ángulos oscilan con el mínimo movimiento realizado, dado que en la figura 3D los puntos que conforma la articulación muñeca – mano es muy corto.

CAPITULO IV

4.1. CONCLUSIONES

- Se consideró en el transcurso de la investigación que el programa LabVIEW fue mejor opción gracias a la librería SDK de Kinect, la cual no se encontraba en una versión adecuada en otro tipo de lenguaje de programación como Python o MATLAB.
- Mediante las pruebas realizadas se comprobó que el brazo robótico es capaz de imitar en un 90% los movimientos ejecutados por el usuario, el cual corrobora la eficiencia del sistema implementado para interacción entre el operador y el manipulador.
- Debido a la baja precisión conseguida en los resultados relacionados con los movimientos de la muñeca del usuario se optó por sustituir los resultados que se obtienen del codo para manipular la articulación que representa a la muñeca del robot. Esta decisión se fundamenta en las similitudes entre los movimientos que realizan estas dos extremidades, lo que permite que la imitación sea lo más real a las acciones que realiza el operador.
- Dado que el robot posee más articulaciones y un gripper, se dispuso el uso del brazo izquierdo del operador para el controlar estas funciones adicionales. Esto permite que el robot ejecute acciones como agarrar objetos con el gripper y mover el brazo hacia delante y atrás ampliando así, su capacidad de trabajo realizando acciones más complejas.
- Se encontró la peculiaridad, que la baja iluminación favorece al sensor infrarrojo de la cámara Kinect en la detección de la persona, puesto que permite un mejor escaneo a diferencia de espacio muy iluminado, la persona, aunque se refleje en la pantalla, su resolución de exactitud en el sistema de imitación será ineficiente por este espacio muy iluminado.
- Se llegó a la conclusión que se necesitó de una plataforma o cuerpo para la ubicación de los componentes físicos del sistema, esto siendo de gran ayuda, permitiendo un mejor control al momento de darse la imitación entre el brazo robótico de forma horizontal, referente al brazo del usuario, el cual se encuentra en esta misma posición para más comodidad.

4.2. RECOMENDACIONES

- Si se opta por usar una cámara Kinect de Xbox 360°, sería recomendable considerar sus limitaciones al momento de capturar los movimientos de la persona, las articulaciones no pueden llegar a toparse ya que éstas pueden interferir entre sí y el movimiento dado por el usuario al brazo robótico será inexacto, para evitar estos fallos sería factible mejor el uso de una Kinect Xbox One la cual dará una mejor interfaz en el campo de 3D.
- Es recomendable primero encender el brazo robótico y posterior la placa Arduino UNO conectado en el ordenador portátil, caso contrario el brazo robótico primero recibirá los 5V de la placa Arduino y al encenderlo, el brazo robótico emitirá una alarma dando lugar una falla de conexión, una forma adicional de evitar este contratiempo es el no conectar el VCC del brazo robótico con el VCC de la placa Arduino.
- Se recomienda colocar en el rango de seguridad permisible el brazo robótico; mínimo de un radio de 1 metro, puesto que los movimientos de este pueden dañar tanto al mecanismo como a la persona o personas en su alrededor, también la cámara Kinect por precaución no debe colocarse en este radio mínimo del brazo robótico, puesto que queda expuesta a que haya roces o golpes de este a la cámara.
- Se recomienda desarrollar un algoritmo basado en redes neuronales que permita mejorar el control de las articulaciones del brazo robótico. Esto permite lograr una mayor precisión y estabilidad en los movimientos realizados a través del sistema de imitación, optimizando la capacidad para replicar los gestos del usuario de manera eficaz.
- Es recomendable tener repuestos del brazo robótico (cables o servos), puesto que estos después de muchas pruebas realizadas tienden a dañarse muy rápido. Por eso, es factible al momento de buscar un brazo robótico de tipo comercial, pueda ser uno que sea posible encontrar repuestos.
- Se recomienda implementar un sistema de comunicación inalámbrico para el control del robot, permitiendo que el sistema de imitación no se limite a una ubicación fija. Esto permite que el robot pueda operar en áreas más amplias y diversas, facilitando su implementación en escenario donde el operador y el robot no estén en la misma ubicación física. Este enfoque mejora la flexibilidad y aplicabilidad

del sistema haciéndolo adecuado para tareas en entornos industriales, educativos o de asistencia remota.

BIBLIOGRAFÍA

- [1] Aníbal Ollero Baturone, *Robótica, manipuladores y robots móviles*. Barcelona, España: MARCOMBO, S.A., 2001.
- [2] Erica Palmerini, "Robótica y derecho: sugerencias, confluencias, evoluciones en el marco de una investigación europea*," *Revista de Derecho Privado*, p. 53 a 97, Enero - Junio 2017.
- [3] Marvin Molina Cárdenas and Patty Pedroza Barrios, "Diseño y Construcción del Prototipo de un Brazo Robótico con Tres Grados de Libertad, como Objeto de Estudio," Barranquilla - Colombia, 2015.
- [4] Pedro Antonio López Ramírez and Hugo Andrade Sosa, "Aprendizaje de y con robótica, algunas experiencias," *Revista Educación*, pp. 43 - 63, junio 2013.
- [5] Intel. (2020, marzo) La fábrica automatizada de Audi cada vez está más cerca de la industria 4.0. [Online]. <https://www.intel.es/content/www/es/es/robotics/robotic-arm.html>
- [6] Eddie Angel Sobrado Malpartida, "Sistema de visión artificial para el reconocimiento y manipulación de objetos utilizando un brazo robot," Lima - Perú, 2003.
- [7] Erik Daniel Garcés Hernández, "Implementación de un prototipo de brazo robótico que realiza control de calidad con visión artificial, para el laboratorio de automatización de la escuela de ingeniería industrial de la epoch," Riobamba - Ecuador, 2017.
- [8] Sandra Nope Rodríguez, Humberto Loaiza-Correa, and Eduardo Caicedo-Bravo, "Imitación de gestos por brazos robóticos: Una propuesta para evaluar su calidad," *DYNA*, pp. 413 - 420, Junio 2010.

- [9] Bernard Marr, "7 asombrosos ejemplos de visión artificial y por computadora en la práctica.," *Forbes*, 2019.
- [10] Andrea Liliana Condo Asnalema, "Desarrollo de un sistema de visión artificial a través de una tarjeta de desarrollo para controlar el brazo robótico Mitsubishi," Quito, Aug. 2017. [Online].
<https://dspace.ups.edu.ec/bitstream/123456789/14537/1/UPS%20-%20ST003185.pdf>
- [11] Luisa Fernanda Gallo Sánchez, Mónica Alejandra Guerrero Ramírez, Juan Diego Vásquez Salcedo, and Miguel Ángel Alonso Castro, "Diseño de un Prototipo Electromecánico para la Emulación de los Movimientos de un Brazo Humano," *INGE CUC*, vol. 12, no. 2, pp. 17 - 25, Aug. 2016.
- [12] Carlos F. Rodríguez, Hugo Quintero, and Helena Aschner, "Movimiento del brazo de los tres planos de dimensiones.," Bogotá - Colombia, Noviembre 2005.
- [13] Jared Avelino Rodriguez. (2017) muciza. [Online].
<https://historico.muciza.com.mx/muciza-2017/project/brazo-robotico-hiadraulico/>
- [14] Germán Darío Buitrago Salazar, "Sistema teledirigido de un brazo robótico de 4 grados de libertad aplicando visión de máquina," *Revista*, vol. 12, no. 24, p. 10, 2015.
- [15] Leopoldo Acosta Sánchez, "Matemáticas y robótica," San Cristobal , 2005.
]
- [16] Aleks Emilov Goranov, "Control de un brazo robot con articulaciones elastica ," Valencia , 2017.
- [17] Roberto de la Torre Zevallos, "Diseño del circuito de control de un actuador para un brazo robotico de apoyo en artroscopia de rodilla para diagnostico de lesiones," 2009.

- [18 Giovanni Sánchez, "Dispositivo Sensórico para el Manejo Inalámbrico del Brazo] Robótico Mitsubishi RV-M1," 2010.
- [19 Chistopher paul Carrasco Villadoma, "Diseño dde un contolador lineal para un] brazo robotico prototipo de 4 grados de libertad orientado a la automatizacion del proceso de empacado en la produccio de pisco," Lima, 2021.
- [20 Diego Almao. (2020, Octubre) cinconoticias. [Online].] <https://www.cinconoticias.com/brazos-mecanicos/>
- [21 Melisa Díaz. SlidePlayer. [Online]. <https://slideplayer.es/slide/13994001/>]
- [22 Antonio Barrientos, *Fundamentos de Robótica*. Madrid: McGraw-Hill] Interamericana de España, 2005.
- [23 Daily Milanés Herмосilla and Alejandro Castilla Pérez , "Generación de] trayectorias para el brazo robótico (ArmX).," *Ingeniería Electrónica, Automática y Comunicaciones*, vol. 37(3), pp. 58-71, Septiembre 2016.
- [24 Jesús Medina Cervantes, Nereyda Castro Gutierrez, Edgar Mejía, and Rubén] Villafuerte, "Aplicación móvil para el control de un brazo r," Universidad Veracruzana, Veracruz, 2016.
- [25 Luis Felipe Giraldo, Edilson Delgado, and Germán Castellanos, "Cinemática] Inversa de un Brazo Robot Utilizando Algoritmos Genéticos," *Revista Avances en Sistemas e Informática*, pp. 29-34, 2006.
- [26 Kryscia Ramírez Benavides. (2021) kramirez. [Online].] <https://www.kramirez.net/Robotica/Material/Presentaciones/CinematicaRobot.pdf>
- [27 César Rolando Batz Saquimux, "Diseño y construcción de un brazo robótico,"] Guatemala, 2005.

- [28 José Luis Ramírez Arias and Astrid Rubiano Fonseca, "Modelamiento Matemático de la cinemática directa e inversa de un robot Manipulador de tres grados de libertad," *Ingeniería solidaria*, vol. 8, pp. 46-52, Septiembre 2012.
- [29 H.S. Lee and S.L. Chang , "Development of a cad/cae/camsystem for a robot] manipulator," *Journal of Materials Processing Technology*, pp. 100-104, 2003.
- [30 Kryscia Ramírez Benavides. [Online].] <https://www.kramirez.net/Robotica/Material/Presentaciones/CinemáticaDirectaRobot.pdf>
- [31 A., Álvarez, M., Hernández, J. D., Del Cerro, J., & Rossi, C. Barrientos, "Modelado de Cadenas Cinemáticas mediante Matrices de Desplazamiento. Una alternativa al método de Denavit-Hartenberg.," *Revista Iberoamericana de Automática e Informática industrial*, vol. 9, pp. 371-382, 2012.
- [32 Edilson Delgado, Germán Castellanos, and Luis Giraldo, "Cinemática inversa de un] brazo robot utilizando algoritmos genéticos.," *Revista Avances en Sistemas e Informática*, vol. 3, no. 1, pp. 29-34, Junio 2006.
- [33 Leonardo Ezequiel Gutiérrez Pérez, "Cinemática inversa utilizando cinemática] diferencial.," *REVISTA CIENTÍFICA*, p. 47, 2009.
- [34 Erik Daniel Garcés Hernández, "Implementación de un prototipo de brazo robótico] que realiza control de calidad con visión artificial, para el laboratorio de automatización de la escuela de ingeniería industrial de la espoeh," Riobamba, 2017.
- [35 Emima Ioana Jiva, "Desarrollo de la teleoperación de robots industriales y] colaborativos mediante técnicas avanzadas de visión artificial," Valencia, 2019.
- [36 P. Pérez and M. Valente. (2018) FaMAF. [Online].] <https://www.famaf.unc.edu.ar/~pperez1/manuales/cim/cap2.html>

- [37 Nicolás Aguirre Dobernack, "Implementación de un sistema de detección de señales] de tráfico mediante visión artificial basado en FPGA.," Universidad de Sevilla, Sevilla, España, 2013.
- [38 Darwin Santiago Aldás Salazar, Santiago Marcos Collantes Vaca, and John Paúl] Reyes Vásquez, "Procesamiento de imágenes con visión artificial para el reconocimiento de placas vehiculares.," *SATHIRI*, vol. 11, pp. 133-142, 2016.
- [39 Luis Eduardo Escobar Carrión, "Desarrollo de algoritmo para visión artificial] aplicada a robots cooperativos en proceso de clasificación y distribución de sustancias químicas," QUITO, 2020.
- [40 Oscar Rubio. (2020, Septiembre) vistronica. [Online].] <https://www.vistronica.com/blog/post/procesamiento-de-imagenes.html>
- [41 Juan A. Ramírez Q. and Mario I. Chacón M., "Redes neuronales artificiales para el] procesamiento de imágenes, una revisión de la última década," *Revista de Ingeniería Eléctrica, Electrónica y Computación*, vol. 9, pp. 7-16, Julio 2011.
- [42 Edwin Monsalve Lesmes and Leonardo Lanido Agudelo, "Programa de] procesamiento de imágenes adquiridas por medio el Sensor Kinect para determinar la posibilidad una víctima en determinada zona.," UNIVERSIDAD DISTRITAL "FRANCISCO JOSÉ DE CALDAS" , Bogotá, MONOGRAFÍA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CONTROL 2015.
- [43 Ana Pascual Tomás, "Desarrollo de una aplicación basada en cámaras 3D para la] generación de movimientos en un robot humanoide," Valencia, 2016.
- [44 Juan Sebastián Hernández Santan, Yosef Harvey Cortes Millán, Carlos Mario] González Giraldo, and Erika Sofía Upegui Cardona, "Reutilización de fotos adquiridas con cámaras 360° de Google Street View para la reconstrucción 3D en un enfoque de promoción turística: estudio de caso Bogotá, Colombia," *Revista Educacion en Ingenieria*, pp. 48 - 55, Julio 2021.

- [45 Eric Magro Viforcós, "Aplicación de las cámaras 3D al reconocimiento de] actividades," Madrid, 2012.
- [46 Freestylephoto. Freestylephoto. [Online]. <https://www.freestylephoto.com/195120->] [Holga-120PC-3D-Stereo-Pinhole-Camera](https://www.freestylephoto.com/195120-Holga-120PC-3D-Stereo-Pinhole-Camera)
- [47 Ivet Challenger Pérez, Yanet Díaz Ricardo, and Robert Antonio Becerra García, "El] lenguaje de programación Python," *Ciencias Holguín*, pp. 1-13, junio 2014.
- [48 Sagar Imambi and Kolla Bhanu Prakash, *Programming with TensorFlow: Solution] for Edge Computing Applications*. Cham: Springer International Publishing, 2021.
- [49 Alejandro Santorum Varela, "Desarrollo de la librería scikit-rmt en Python para la] simulación y el análisis de matrices aleatorias," Madrid, 2021.
- [50 Carlos Antona Cortés, "Herramientas modernas en redes neuronales: la librería] Keras ," Madrid, 2017.
- [51 Vicente Benjumea and Manuel Roldán, "Fundamentos de Programación con el] Lenguaje de Programación C++," Málaga, 2022.
- [52 Álvaro Loras Navas, "Percepción de la actividad de un conductor mediante cámaras] 3D," Madrid, 2013.
- [53 Juan Zamorano Ruiz, "Comparación y análisis de métodos de clasificación con las] bibliotecas scikit-learn y TensorFlow en Python.," Málaga, 2019.
- [54 Anthony Stalin Llerena Buenaño and Marc Jens Salazar Villamar, "Automatización] de un sistema identificador y posicionador de objetos a través de un brazo robótico mediante visión artificial con lenguaje python," Guayaquil, 2022.
- [55 Pablo Garrido Abenza , "Comenzando a programar con JAVA.," 2015.
]

- [56 Yenisleidy Fernández Romero and Karen García Pombo, "Interfaz USB," *Revista Telematica*, Enero - Abril 2012.
- [57 Rodrigo Castro, "Avanzando en la seguridad de las redes WIFI.," *Boletín de RedIRIS*, pp. 23 - 33, Septiembre 2005.
- [58 Diana Franco and Francisco Castillo, "Comunicaciones Inalámbricas: Bluetooth.," *Prisma Tecnológico*, vol. 1, no. 1, pp. 19 - 21, Septiembre 2009.
- [59 Noel Machado Toranzo, "Seguidor Solar, optimizando el aprovechamiento de la energía solar," *Ingeniería Energética*, pp. 190-199, Agosto 2015. [Online].
http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1815-59012015000200008&lng=es&nrm=iso
- [60 Antonio Caicedo Pedrera, *Arduino para Principiantes: 2ª Edición.*: IT Campus Academy, 2017.
- [61 Yúbal Fernández. (2022, Septiembre) Xataka. [Online].
<https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>
- [62 Albert Álvarez Carulla, "Comunicación de un módulo ESP32 con Ubidots mediante MQTT.," 2021.
- [63 NOVAtronic. NOVAtronic. [Online].
<https://novatronicec.com/index.php/product/esp32-placa-de-desarrollo-v1/>
- [64 Eugenio López Aldea, *Raspberry PI fundamentos y aplicaciones.*: Ra-Ma Editorial., 2017.
- [65 Yúbal Fernández. (2018, Agosto) Xataka. [Online].
<https://www.xataka.com/basics/arduino-raspberry-pi-que-cuales-sus-diferencias>

- [66 Eddie Angel Sobrado Malpartida, "Sistema de visión artificial para el] reconocimiento y manipulación de objetos utilizando un brazo robot.," Perú, 2011.
- [67 Sandra Nope Rodríguez , Humberto Loaiza-Correa, and Eduardo Caicedo-Bravo,] "Imitación de gestos por brazos robóticos: Una propuesta para evaluar su calidad.," 2010.
- [68 Jesús Sergio Artal and Luis Montañés, "Desarrollo de un brazo robotico e] implementacion de una estrategia de control por reconocimiento de gestos mediante Leap Motion.," Zaragoza, 2016.
- [69 Jhon Alexander Niño Peña, Pablo Enrique Parra Gaon, and Sebastian Felipe] Rodriguez, "Brazo robotico manipulador de residuos envasados," Colombia, 2017-02-21.
- [70 electrocompuquito. electrocompuquito. [Online].] <https://www.electrocompuquito.com/product/arduino-uno-r3-atmega-328p/>
- [71 Cesar Cinjordiz. (2015, Agosto) infootec.net. [Online].] <https://www.infootec.net/arduino/>
- [72 Hiwonder. [Online].] https://drive.google.com/drive/folders/1j_YVIQjYEU06SoYOjVJ7aGmyR_S7v8Jf
- [73 Leland Holmquest , "Kinect - Starting to Develop with Kinect," *MSDN Magazine*,] Junio 2012.
- [74 José Guerra Carmenate. programarfacil. [Online].] <https://programarfacil.com/blog/arduino-blog/arduino-ide/>
- [75 Andrea Liliana Condo Asnalema, "Desarrollo de un sistema de visión artificial a] través de una tarjeta de desarrollo para controlar el brazo robótico Mitsubishi.,"

Quito,

Aug.

2017.

[Online].

<https://dspace.ups.edu.ec/bitstream/123456789/14537/1/UPS%20-%20ST003185.pdf>

ANEXOS

ANEXO A - Código de simulación Arduino

```
#include "LobotServoController.h" // Librería del controlador de servos
#include <SoftwareSerial.h>       // Librería para SoftwareSerial

// Definir pines para SoftwareSerial (ajusta estos pines según tu
// configuración)
SoftwareSerial mySerial(10, 11); // RX en pin 10, TX en pin 11

// Instanciar el controlador del servo y pasar el puerto SoftwareSerial
LobotServoController myse(mySerial);

char Dato;           // Recibe cada uno de los caracteres
String arreglo = ""; // Almacena temporalmente los datos recibidos
int ID = 0;          // Variable para el ID del servo
int Angulo = 0;      // Variable para el ángulo del servo

void setup() {
    Serial.begin(9600); // Puerto serie para recepción de datos
    mySerial.begin(9600); // Inicializar SoftwareSerial en los pines 10 y
    11 para el controlador de servos
}

void loop() {
    if (Serial.available() > 0) // Verifica la disponibilidad del puerto
    {
        Dato = Serial.read(); // Lee un carácter del puerto serie

        if (Dato == ':') // Cuando llega el delimitador ':', identificamos
        que es el ID
        {
            ID = arreglo.toInt(); // Convertimos el ID a entero
            arreglo = ""; // Reiniciamos el arreglo para leer el ángulo
        }
        else if (Dato == ',') // Cuando llega la coma, significa que termina
        el ángulo
        {
            Angulo = arreglo.toInt(); // Convertimos el ángulo a entero
            arreglo = ""; // Reiniciamos el arreglo para el próximo conjunto de
            datos

            // Asegurarse de que el ángulo esté en el rango correcto
            Angulo = constrain(Angulo, 0, 180); // Limitar el ángulo entre 0 y
            180 grados
        }
    }
}
```

```

    // Convertir el ángulo en el rango de 500 a 2500 (si el controlador
lo requiere)
    int servoPosition = map(Angulo, 0, 180, 500, 2500);

    // Movemos el servomotor correspondiente según el ID
    if (ID == 1)
    {
        myse.moveServo(3, servoPosition, 20); // Mover el servo 1 en 20
ms al ángulo mapeado
    }
    else if (ID == 2)
    {
        myse.moveServo(4, servoPosition, 20); // Mover el servo 2 en 500
ms al ángulo mapeado
    }
    }
    else // Si no es delimitador, seguimos construyendo la cadena
    {
        arreglo += Dato;
    }
}
}
}

```

Tesis_Vera y Espinoza

3%
 Textos sospechosos



< 1% Similitudes
 0% similitudes entre comillas
 < 1% entre las fuentes mencionadas

2% Idiomas no reconocidos

7% Textos potencialmente generados por la IA (ignorado)

Nombre del documento: Tesis_Vera y Espinoza.docx
ID del documento: 109f16716eb55ecd8d325fc887d3042ec4441221
Tamaño del documento original: 17,07 MB
Autores: []

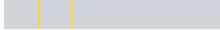
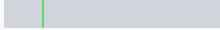
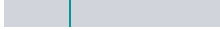
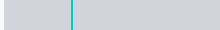
Depositante: Junior Rafael Figueroa Olmedo
Fecha de depósito: 29/11/2024
Tipo de carga: interface
fecha de fin de análisis: 29/11/2024

Número de palabras: 20.824
Número de caracteres: 135.706

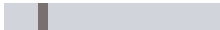
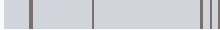

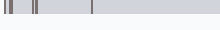
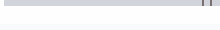
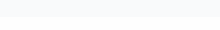
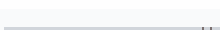
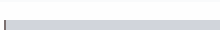
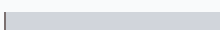
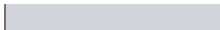
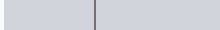

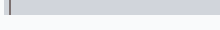

Ubicación de las similitudes en el documento:


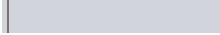

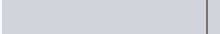

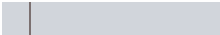

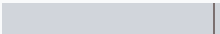

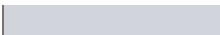




Fuentes principales detectadas




N°	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	Documento de otro usuario #56c609 El documento proviene de otro grupo 14 fuentes similares	< 1%		Palabras idénticas: < 1% (56 palabras)
2	blogs.uninter.edu.mx https://blogs.uninter.edu.mx/ESCAT/wp-content/uploads/2023/07/PROYECTO-FINAL-1.pdf	< 1%		Palabras idénticas: < 1% (35 palabras)
3	www.kramirez.net https://www.kramirez.net/Robotica/Material/Presentaciones/CinematicaDirectaRobot.pdf 7 fuentes similares	< 1%		Palabras idénticas: < 1% (31 palabras)
4	www.doi.org https://www.doi.org/10.1109/CIIMA50553.2020.9290293 12 fuentes similares	< 1%		Palabras idénticas: < 1% (32 palabras)

Fuentes ignoradas Estas fuentes han sido retiradas del cálculo del porcentaje de similitud por el propietario del documento.

N°	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	Documento de otro usuario #af4b02 El documento proviene de otro grupo	4%		Palabras idénticas: 4% (862 palabras)
2	Documento de otro usuario #292fe6 El documento proviene de otro grupo	3%		Palabras idénticas: 3% (564 palabras)
3	Documento de otro usuario #53cfe6 El documento proviene de otro grupo	2%		Palabras idénticas: 2% (324 palabras)
4	Documento de otro usuario #c7e4be El documento proviene de otro grupo	2%		Palabras idénticas: 2% (318 palabras)
5	Documento de otro usuario #45d4d5 El documento proviene de otro grupo	< 1%		Palabras idénticas: < 1% (56 palabras)
6	dspace.ups.edu.ec http://dspace.ups.edu.ec/bitstream/123456789/22832/1/UPS-GT003848.pdf	< 1%		Palabras idénticas: < 1% (59 palabras)
7	repositorio.upse.edu.ec https://repositorio.upse.edu.ec/bitstream/46000/9175/1/UPSE-TEA-2023-0003.pdf	< 1%		Palabras idénticas: < 1% (53 palabras)
8	dspace.esPOCH.edu.ec DSpace ESPOCH.: Implementación de un prototipo de brazo... http://dspace.esPOCH.edu.ec/handle/123456789/6850	< 1%		Palabras idénticas: < 1% (54 palabras)
9	Documento de otro usuario #568f62 El documento proviene de otro grupo	< 1%		Palabras idénticas: < 1% (38 palabras)
10	Documento de otro usuario #763655 El documento proviene de otro grupo	< 1%		Palabras idénticas: < 1% (38 palabras)
11	Documento de otro usuario #851b99 El documento proviene de otro grupo	< 1%		Palabras idénticas: < 1% (38 palabras)
12	dialnet.unirioja.es Desarrollo de un Brazo Robótico e implementación de una estr... https://dialnet.unirioja.es/servlet/articulo?codigo=8665046	< 1%		Palabras idénticas: < 1% (36 palabras)
13	Documento de otro usuario #fe3e68 El documento proviene de otro grupo	< 1%		Palabras idénticas: < 1% (28 palabras)
14	TESINA JAIRALA STEEVEN.pdf TESINA JAIRALA STEEVEN #7fc15b El documento proviene de mi grupo	< 1%		Palabras idénticas: < 1% (27 palabras)

Nº	Descripciones	Similitudes	Ubicaciones	Datos adicionales
15	 repositorio.upse.edu.ec https://repositorio.upse.edu.ec/bitstream/46000/12012/1/UPSE-TEA-2024-0018.pdf	< 1%		🔗 Palabras idénticas: < 1% (27 palabras)
16	 riunet.upv.es Modelado de Caden as Cinemáticas mediante Matrices de Desplaza... https://riunet.upv.es/handle/10251/144406	< 1%		🔗 Palabras idénticas: < 1% (20 palabras)
17	 www.doi.org https://www.doi.org/10.1109/CWCAS.2014.6994614	< 1%		🔗 Palabras idénticas: < 1% (18 palabras)
18	 dspace.ups.edu.ec Repositorio Institucional de la Universidad Politécnica Salesiana... https://dspace.ups.edu.ec/handle/123456789/22832	< 1%		🔗 Palabras idénticas: < 1% (19 palabras)
19	 Documento de otro usuario #ce7b46 👤 El documento proviene de otro grupo	< 1%		🔗 Palabras idénticas: < 1% (16 palabras)
20	 www.doi.org https://www.doi.org/10.1109/ANDESCON.2018.8564569	< 1%		🔗 Palabras idénticas: < 1% (14 palabras)

Fuentes mencionadas (sin similitudes detectadas) Estas fuentes han sido citadas en el documento sin encontrar similitudes.

-  <https://www.intel.es/content/www/es/es/robotics/robotic-arm.html>
-  <https://historico.muciza.com.mx/muciza-2017/project/brazo-robotico-hidraulico/>
-  <https://www.cinconoticias.com/brazos-mecanicos/>
-  <https://slideplayer.es/slide/13994001/>
-  <https://www.kramirez.net/Robotica/Material/Presentaciones/CinematicaRobot.pdf>